Idoc スクリプト・リファレンス・ガイド

10*g* リリース 3（10.1.3.3.0）

部品番号 : B51312-01

2008 年 12 月

ORACLE®

Idoc スクリプト・リファレンス・ガイド , 10*g* リリース 3（10.1.3.3.0）

部品番号 : B51312-01

原本名： Idoc Script Reference Guide, 10*g* Release 3 (10.1.3.3.0)

原本協力者 : Will Harri, Jean Wilson, Sebastian Celis, Eva Cordes, Alec Kloss, Rick Petty, Sam White

Copyright © 2007, Oracle. All rights reserved.

制限付権利の説明

このプログラム（ソフトウェアおよびドキュメントを含む）には、オラクル社およびその関連会社に所有権のある情報が含まれています。このプログラムの使用または開示は、オラクル社およびその関連会社との契約に記された制約条件に従うものとします。著作権、特許権およびその他の知的財産権と工業所有権に関する法律により保護されています。

独立して作成された他のソフトウェアとの互換性を得るために必要な場合、もしくは法律によって規定される場合を除き、このプログラムのリバース・エンジニアリング、逆アセンブル、逆コンパイル等は禁止されています。

このドキュメントの情報は、予告なしに変更される場合があります。オラクル社およびその関連会社は、このドキュメントに誤りが無いことの保証は致し兼ねます。これらのプログラムのライセンス契約で許諾されている場合を除き、プログラムを形式、手段（電子的または機械的）、目的に関係なく、複製または転用することはできません。

このプログラムが米国政府機関、もしくは米国政府機関に代わってこのプログラムをライセンスまたは使用する者に提供される場合は、次の注意が適用されます。

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations.As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987).Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

このプログラムは、核、航空、大量輸送、医療あるいはその他の本質的に危険を伴うアプリケーションで使用されることを意図しておりません。このプログラムをかかる目的で使用する際、上述のアプリケーションを安全に使用するために、適切な安全装置、バックアップ、冗長性（redundancy）、その他の対策を講じることは使用者の責任となります。万一かかるプログラムの使用に起因して損害が発生いたしましても、オラクル社およびその関連会社は一切責任を負いかねます。

Oracle、JD Edwards、PeopleSoft、Siebel は米国 Oracle Corporation およびその子会社、関連会社の登録商標です。その他の名称は、他社の商標の可能性があります。

このプログラムは、第三者の Web サイトへリンクし、第三者のコンテンツ、製品、サービスへアクセスすることがあります。オラクル社およびその関連会社は第三者の Web サイトで提供されるコンテンツについては、一切の責任を負いかねます。当該コンテンツの利用は、お客様の責任になります。第三者の製品またはサービスを購入する場合は、第三者と直接の取引となります。オラクル社およびその関連会社は、第三者の製品およびサービスの品質、契約の履行（製品またはサービスの提供、保証義務を含む）に関しては責任を負いかねます。また、第三者との取引により損失や損害が発生いたしましても、オラクル社およびその関連会社は一切の責任を負いかねます。

目次

第 1 章 : はじめに

構成 1-2
このガイドについて 1-2
対象読者 1-3
表記規則 1-3

第 2 章 : 新機能

使用に関する最新情報 2-1
10*g*R3 で新規 / 修正された Idoc スクリプト 2-2
7.5.1 で新規 / 修正された Idoc スクリプト 2-7

第 3 章 : Idoc スクリプトの使用

Idoc のネーミング規則 3-2
Idoc スクリプトの構文 3-2
Idoc スクリプトのタグ 3-2
Idoc スクリプトのコメント 3-3
Idoc スクリプトの使用方法 3-3
インクルード 3-4
インクルードの例 3-5
super タグ 3-6
super タグの例 3-7
変数 3-7
変数の作成 3-8
変数の参照 3-8
値の割当て 3-8
カンマをセパレータとして使用 3-8
条件での変数の参照 3-9
変数の置換順序 3-9
標準変数 3-9
関数 3-10

パーソナライズ関数 3-10
条件 3-11
条件の例 3-12
ループ 3-13
ResultSet のループ 3-13
ResultSet のループの例 3-14
while ループ 3-14
while ループの例 3-15
ループの終了 3-15
管理インタフェース 3-16
ワークフロー管理 3-16
Web レイアウト・エディタ 3-17
バッチ・ローダー 3-17
アーカイバ 3-17
システム・プロパティ 3-17
検索式 3-18
電子メール 3-18
特殊キーワード 3-18
キーワードと関数の対比 3-19
exec キーワード 3-20
eval 関数 3-20
include キーワード 3-21
inc 関数 3-21
演算子 3-22
比較演算子 3-22
特殊な文字列演算子 3-23
ワイルドカード 3-24
数値演算子 3-24
ブール演算子 3-25
メタデータ・フィールド 3-26
メタデータ・フィールドのネーミング 3-26
標準メタデータ・フィールド 3-27
共通メタデータ・フィールド 3-27
その他のフィールド 3-28
オプション・リスト 3-33
内部オプション・リスト 3-33
オプション・リストのスクリプト 3-34
オプション・リストの作成 3-35
動的サーバー・ページでのメタデータの参照 3-36
Verity Integration での日付 / 時間 3-37
MergeInclude を使用したレスポンスのフォーマット 3-38
有効範囲付のローカル変数 3-40

第 4 章 : タイプ別 Idoc スクリプト

条件付き動的変数 4-2
構成変数 4-3
Collaboration Manager の構成変数 4-4
Content Server の構成変数 4-5
Inbound Refinery の構成変数 4-16
Link Manager の構成変数 4-19
Pop-Up Calendar の構成変数 4-19
Record Manager および Retention Manager の構成変数 4-20
動的変数 4-20
グローバル関数 4-21
ページ変数 4-25
ページ表示変数 4-25
フィールド表示変数 4-26
フィールド情報変数 4-26
共通フィールド表示変数 4-27
その他のフィールド表示変数 4-29
読取り専用変数 4-31
テンプレート読取り専用変数 4-31
テンプレート読取り専用変数の例 4-32
ユーザー読取り専用変数 4-33
コンテンツ読取り専用変数 4-33
その他の読取り専用変数 4-33
設定可能変数 4-34
値変数 4-34
Web サーバー変数 4-36
ワークフロー・スクリプト 4-37
ワークフローの概要 4-37
ワークフロー関数 4-38
ワークフロー変数 4-39

第 5 章 : 使用方法別 Idoc スクリプト

アーカイバ 5-3
バッチ・ローダー 5-3
クライアント 5-4
クラスタリング 5-5
Collaboration Manager 5-5
コンテンツ・アイテム 5-6

コンテンツ・プロファイル 5-7
コンテンツ・サーバー 5-8
変換 5-11
Inbound Refinery 5-11
PDF Converter 5-12
XML Converter 5-16
Dynamic Converter 5-17
Compression 5-18
データベース 5-18
日付と時間 5-19
デバッグ 5-20
ディレクトリおよびパス 5-22
Enterprise Search 5-24
フィールド表示 5-24
Folders および WebDAV 5-26
HTML プレビュー 5-26
Idoc スクリプト 5-27
索引付け 5-28
JSP 5-29
ローカライゼーション 5-29
ページ表示 5-30
パーソナライズ 5-31
Pop-Up Calendar 5-31
Record Manager および Retention Manager 5-32
ResultSet 5-32
スキーマ 5-33
検索 5-34
セキュリティ 5-36
内部セキュリティ 5-36
外部セキュリティ 5-37
文字列 5-39
テンプレート 5-40
ユーザー 5-40
Web サーバー 5-42
ワークフロー 5-44

第 6 章： Idoc スクリプト索引

abortToErrorPage 6-2
AcadUseLISPInterface 6-3
AccountMapPrefix 6-4
AcctPermDelim 6-5
AcctPrefix 6-7
actionItemInterval 6-9
addEmptyOption 6-10
AdditionalIndexBuildParams 6-11
AdditionalSubscriptionCollatedFieldList 6-14
AdjustPrinterMargins 6-15
AdminAtLeastOneGroup 6-16
AdminDataDir 6-17
AdsSimpleAuth 6-18
AdsUserName 6-19
AdsUserPassword 6-20
AfterLogin 6-21
AllowAllRedirectUrl 6-22
AllowAlternateMetaFile 6-23
AllowCheckin 6-24
AllowCheckout 6-25
AllowConcurrentUpdate 6-26
AllowForceDelete 6-27
AllowIntranetUsers 6-28
AllowMatchesInDateCheck 6-29
AllowMixedCaseACL 6-30
AllowMrSidRetryCompressionLevel 6-31
AllowPassthru 6-32
AllowPrimaryMetaFile 6-33
AllowRetentionPeriodWithoutCutoff 6-34
AllowReview 6-35
AllowSamePrimaryAlternateExtensions 6-36
AllowSecondaryConversion 6-37

AllowSkippedHyperlinktoCauseIncomplete 6-38
AppletChunkSize 6-39
AppletChunkThreshold 6-40
ARCHIVE_COLLECTION 6-41
ArchiverDoLocks 6-42
ArchiverTouchMonitorInterval 6-43
ArchiverTraceLockMaintenance 6-44
ArchiverTraceLocks 6-45
AttributeMap 6-46
AuthorAddress 6-47
AuthorDelete 6-48
AutoCad2000PlotterFilePath 6-49
AutoMaxMergeTrace 6-50
AutoNumberPrefix 6-51
AutoNumberWidth 6-52
BatchLoaderPath 6-53
BatchLoaderUserName 6-54
BinDir 6-55
break 6-56
BrowserVersionNumber 6-57
c 6-58
CachedResultRowCount 6-59
cacheInclude 6-60
captionEntryWidth 6-62
captionFieldWidth 6-63
CGI_DEBUG 6-64
CGI_RECEIVE_DUMP 6-65
CGI_SEND_DUMP 6-66
CgiFileName 6-67
CharMap 6-68
CheckMkvdkDocCount 6-69
ChunkedRequestTrace 6-70
CLASSPATH 6-71

CleanUp 6-72
clearSchemaData 6-73
ClientControlled 6-74
ClusterBinDirRule 6-75
ClusterNodeAddress 6-76
ClusterNodeName 6-77
cmDiscussionDocType 6-78
cmMemoFieldSize 6-79
cmZonedSecurityFields 6-80
CollectionDeleteEnabled 6-81
CollectionHiddenMeta 6-82
CollectionInhibitUpdateMeta 6-83
CollectionLocation 6-84
CollectionMeta 6-85
CollectionMoveEnabled 6-86
CollectionReadOnlyMarkedFolders 6-87
CollectionReadOnlyMeta 6-88
CollectionReleasedOnly 6-89
CollectionSecurityReadOnly 6-90
CollectionTrashDeleteDate 6-91
CollectionTrashDeleteLocation 6-92
CollectionTrashDeleteOldName 6-93
CollectionTrashDeleter 6-94
CollectionWebDAVServer 6-95
ColumnMapFile 6-96
CompressionMinHeight 6-97
CompressionMinWidth 6-98
computeDocUrl 6-99
computeRenditionUrl 6-100
ComputerName 6-101
ConfigDir 6-102
ConnectionName 6-103
CONTENT_LENGTH 6-104

ConversionCacheExpirationPeriodInDays 6-105
ConvertXmlAsWebviewable 6-106
CookieAuthTimeoutInDays 6-107
CookieProxyPassword 6-108
coreContentOnly 6-109
createAlternateMetaFile 6-110
CreatePDFThumbnails 6-111
createPrimaryMetaFile 6-112
CreateXMLRendition 6-113
CURRENT_DATE 6-114
CURRENT_ROW 6-115
CustomConversionWaitTime 6-116
CustomConverterPath 6-117
DatabasePreserveCase 6-118
DatabaseWildcards 6-119
DataDir 6-120
dateCurrent 6-121
DatedCacheIntervalDays 6-122
DateOutputFormat 6-123
DCMaxFileSize 6-124
dcShowExportLink 6-125
DCTimeOut 6-126
DCViewFormat 6-127
DebugMode 6-128
DebugStdConversion 6-129
Default Accounts 6-130
DefaultAuth 6-131
defaultFieldInclude 6-132
DefaultFilterInputFormat 6-133
DefaultFilterOutputFormat 6-134
DefaultGraphicTimeout 6-135
DefaultHtmlConversion 6-136
DefaultHtmlRefreshTimeoutInSeconds 6-137

DefaultMasterDomain 6-138
DefaultNativeTimeout 6-139
DefaultNetworkAccounts 6-140
defaultOptionListScript 6-142
DefaultPasswordEncoding 6-143
DefaultPostscriptTimeout 6-144
deleteActionItemInterval 6-145
deleteWatchdogInterval 6-146
DelimitedUserRoles 6-147
DirectoryLockingLogPath 6-148
DisableAmberLayouts 6-149
DisableClassicLayoutPublishing 6-150
DisableCollegiateSkinPublishing 6-151
DisableContentProfiles 6-152
DisableDBLongIntegerField 6-153
DisableForcedConversions 6-154
DisableGzipCompression 6-155
DisableHttpUploadChunking 6-156
DisableKeepingForcedConversions 6-157
DisableMeetingCenter 6-158
DisableOptimizedLatestReleaseQuerying 6-159
DisableSharedCacheChecking 6-160
DisableStellent05SkinPublishing 6-161
DisableStellentSkinPublishing 6-162
DisableTopMenusPublishing 6-163
DisableTraysPublishing 6-164
DisableWindowsPublishing 6-165
DisplayPSOrigin 6-166
DistillerNormJobSetting 6-167
DistillerOptJobSetting 6-168
DistillerPath 6-169
DoAutoMaxMerge 6-170
DocConverterEngineDir 6-171

docLoadResourceIncludes 6-172
docRootFilename 6-175
DocTypeSelected 6-176
DocUrl 6-177
docUrlAllowDisclosure 6-178
DoDocNameOrder 6-179
DomainControllerName 6-180
DownloadApplet 6-181
DownloadSuggestedName 6-182
dpGet 6-183
dpPromote 6-184
dpPromoteRs 6-185
dpSet 6-186
DtmPort 6-187
dWfName 6-188
dWfStepName 6-189
DynamicConverterHtmlFlavor 6-190
EmptyAccountCheckinAllowed 6-191
EnableCopyConvertCopy 6-192
EnableDocumentHighlight 6-193
EnableEnvironmentPackager 6-194
EnableErrorFile 6-195
EnableExpirationNotifier 6-196
EnableOptimizedLatestReleaseQuerying 6-198
EnableRebuildForcesReconversion 6-199
EnableSchemaPublishing 6-200
encodeHtml 6-201
EnterpriseSearchAsDefault 6-203
entryCount 6-204
eval 6-205
ExcludeConversionFilter 6-207
ExclusiveCheckout 6-208
ExclusiveIncludeConversionFilter 6-209

exec 6-210
executeService 6-211
ExpireRevisionOnly 6-213
ExternalUserAccounts 6-214
ExternalUserRoles 6-215
fieldCaption 6-216
fieldCaptionInclude 6-217
fieldCaptionStyle 6-218
fieldDefault 6-219
fieldEditWidth 6-220
fieldEntryInclude 6-221
fieldExtraScriptInclude 6-222
fieldInclude 6-223
fieldIsOptionList 6-224
fieldMaxLength 6-225
fieldName 6-226
fieldOptionListType 6-227
fieldType 6-228
fieldValue 6-229
fieldValueStyle 6-230
fieldWidth 6-231
FileEncoding 6-232
fileUrl 6-233
FILTER_DEBUG 6-234
FIRSTREV 6-235
ForcedConversionRules 6-236
ForceDistinctRevLabels 6-237
ForceDocTypeChoice 6-238
forceExpire 6-239
ForceJdbcSync 6-241
ForceSecurityGroupChoice 6-242
formatDate 6-243
formatDateDatabase 6-245

formatDateDisplay 6-246
formatDateOnly 6-247
formatDateOnlyDisplay 6-249
formatDateOnlyFull 6-250
formatDateWithPattern 6-251
formatTimeOnly 6-252
formatTimeOnlyDisplay 6-253
FormInheritMetaDataFieldList 6-254
FrameMakerCyclesAllowed 6-255
FrameMakerexePath 6-256
FrameMakerTimeout 6-257
GATEWAY_INTERFACE 6-258
generateUniqueId 6-259
getCookie 6-260
GetCopyAccess 6-261
getDebugTrace 6-262
getErrorTrace 6-263
getFieldConfigValue 6-264
getFieldViewDisplayValue 6-265
getFieldViewValue 6-266
getFreeMemory 6-267
getHelpPage 6-268
getOptionListSize 6-269
getParentValue 6-270
getRequiredMsg 6-271
getTextFile 6-272
getTotalMemory 6-273
getUserValue 6-274
getValue 6-275
getValueForSpecifiedUser 6-278
getViewValue 6-279
getViewValueResultSet 6-280
hasAppRights 6-281

HasExternalUsers 6-282
HasGlobalUsers 6-283
HasLocalCopy 6-284
hasOptionList 6-285
HasOriginal 6-286
HasPredefinedAccounts 6-287
HasSiteStudio 6-288
HasUrl 6-289
HeavyClient 6-290
HelpDir 6-291
HideVitalReview 6-292
HighlightBegin 6-293
HighlightEnd 6-294
HtmlDataInputFilterLevel 6-295
HTMLEditorPath 6-296
htmlRefreshTimeout 6-297
htmlRefreshUrl 6-298
HTTP_ACCEPT 6-299
HTTP_ACCEPT_ENCODING 6-300
HTTP_ACCEPT_LANGUAGE 6-301
HTTP_COOKIE 6-302
HTTP_HOST 6-303
HTTP_INTERNETUSER 6-304
HTTP_REFERER 6-305
HTTP_USER_AGENT 6-306
HttpAbsoluteCgiPath 6-307
HttpAbsoluteCgiRoot 6-308
HttpAbsoluteWebRoot 6-309
HttpAdminCgiPath 6-310
HttpBrowserFullCgiPath 6-311
HttpCgiPath 6-312
HttpCommonRoot 6-313
HttpEnterpriseCgiPath 6-314

HttpHelpRoot 6-315
HttpIgnoreWebServerInternalPortNumber 6-316
HttpImagesRoot 6-317
HttpLayoutRoot 6-318
HttpRelativeAdminRoot 6-319
HttpRelativeCgiRoot 6-320
HttpRelativeWebRoot 6-321
HttpServerAddress 6-322
HttpSharedRoot 6-323
HttpSystemHelpRoot 6-324
HttpWebRoot 6-325
IDC_Name 6-326
IdcAdminServerHostName 6-327
IdcAdminServerHostName 6-328
IdcAdminServerPort 6-329
IdcAnalyzeLogDir 6-330
IdcAuthExtraRequestParams 6-331
IdcCommandFile 6-332
IdcCommandLog 6-333
IdcCommandServerHost 6-334
IdcCommandUser 6-335
IdcHttpHeaderVariables 6-336
IdcLocale 6-338
IdcRegistrationPath 6-339
IdcServerBindAddress 6-340
IdcServerSocketQueueDepth 6-341
ImageAlchemyExePath 6-342
inc 6-343
incDynamicConversionByRule 6-345
incGlobal 6-346
include 6-347
IncludeXmlFlexionDocType 6-349
incTemplate 6-350

IndexableRenditions 6-351
IndexerAutoWorkInterval 6-352
IndexerLargeFileSize 6-353
IndexerPath 6-354
IndexerRebuildStagingManualCopy 6-355
IndexerRebuildStagingPath 6-356
indexerSetCollectionValue 6-357
IndexerStyleDir 6-358
IndexerTouchMonitorInterval 6-359
IndexerTraceCycleListChanges 6-360
IndexerTraceLockFileChanges 6-361
IndexerTraceStartStop 6-362
IndexVaultFile 6-363
InitialColID 6-365
InstanceDescription 6-366
InstanceMenuLabel 6-367
IntradocDir 6-368
IntradocRealm 6-369
IntradocServerHostName 6-370
IntradocServerPort 6-371
isActiveTrace 6-372
IsAllowIndexRenditions 6-373
IsAllowIndexVault 6-374
IsAutoArchiver 6-375
IsAutoNumber 6-376
IsAutoQueue 6-377
IsAutoSearch 6-378
isCheckin 6-379
IsCheckinPreAuthed 6-380
isComponentEnabled 6-381
IsContentRefineryPresent 6-382
IsContributor 6-383
IsCriteriaSubscription 6-384

IsCurrentNav 6-385
isDocPage 6-386
IsDpSubmitErrorFatal 6-387
IsDynamic 6-388
IsDynamicConverterEnabled 6-389
isEditMode 6-390
IsEditRev 6-391
isExcluded 6-392
IsExternalUser 6-393
IsFailedConversion 6-394
IsFailedIndex 6-395
isFalse 6-396
isFieldExcluded 6-397
isFieldHidden 6-398
isFieldInfoOnly 6-399
isFieldMemo 6-400
IsFilePresent 6-401
IsFormsPresent 6-402
isFormSubmit 6-403
rule:IsFragmentOnly 6-404
IsFullTextIndexed 6-405
isHidden 6-406
isInfo 6-407
isInfoOnly 6-408
IsIntranetAuthOnly 6-409
IsJava 6-410
IsJdbc 6-411
IsJdbcLockTrace 6-412
IsJdbcQueryTrace 6-413
IsJspServerEnabled 6-414
isLayoutEnabled 6-415
IsLocalSearchCollectionID 6-416
IsLoggedIn 6-417

IsMac 6-418
IsMaxRows 6-419
isMultiOption 6-420
IsMultiPage 6-421
isNew 6-422
IsNotLatestRev 6-423
IsNotSyncRev 6-424
IsNTCompressionPresent 6-425
IsOverrideFormat 6-426
isPhysicallySplitDir 6-427
IsPrimaryUserBase 6-428
IsPromptingForLogin 6-429
IsProxiedServer 6-430
isQuery 6-431
isRelocated 6-432
isRepromptLogin 6-433
IsRequestError 6-434
isRequired 6-435
IsSavedQuery 6-436
IsSavedWfCompanionFile 6-437
IsSoap 6-438
isStrictList 6-439
IsSubAdmin 6-440
IsSun 6-441
IsSysManager 6-442
IsThumbnailPresent 6-443
isTrue 6-444
isUpdate 6-446
isUploadFieldScript 6-447
IsUploadSockets 6-448
IsUserEmailPresent 6-449
isUserOverrideSet 6-450
isValidateFile 6-451

isVerboseTrace 6-452
IsWebServerPagesOnly 6-453
IsWindows 6-454
IsWorkflow 6-455
IsXml 6-456
indexerSetCollectionValue 6-457
isZoneField 6-458
JdbcConnectionString 6-459
JdbcDriver 6-460
JdbcPassword 6-461
JdbcPasswordEncoding 6-462
JdbcUser 6-463
js 6-464
jsFilename 6-465
JspAdminQuery 6-466
JspDefaultIndexPage 6-467
JspEnabledGroups 6-468
JvmCommandLine 6-469
lastEntryTs 6-470
LatestReleasedTimestampTimeout 6-471
lc 6-472
lcCaption 6-474
LdapAdminDN 6-475
LdapAdminPassword 6-476
LdapPort 6-477
LdapServer 6-478
LdapSuffix 6-479
LkDisableOnRebuild 6-480
LkDisableReferenceInfo 6-481
LkDisallowConversionFormats 6-482
LkIsSecureSearch 6-484
LkReExtractOnRebuild 6-485
LkRefreshBatchSize 6-486

LkRefreshErrorPercent 6-488
LkRefreshErrorsAllowed 6-489
LkRefreshErrorThreshold 6-490
LMDefaultLayout 6-491
LMDefaultSkin 6-492
lmGetLayout 6-493
lmGetSkin 6-494
loadCollectionInfo 6-495
loadDocMetaDefinition 6-496
loadDocumentProfile 6-497
lcCaption 6-498
loadEnterpriseSearchCollections 6-499
loadEnterpriseSearchResults 6-500
loadSchemaData 6-501
loadSearchOperatorTables 6-502
loadUserMetaDefinition 6-503
LocaleArchiverDateFormat 6-504
LocalGroupServer 6-505
localPageType 6-506
LogDir 6-507
LogMessagePrefix 6-508
LogName 6-509
LogWithConnectionName 6-510
MacSupportsSignedApplets 6-511
MailServer 6-512
MajorRevSeq 6-513
MaxArchiveErrorsAllowed 6-514
MaxCollectionSize 6-515
MaxConversionCacheSizeInMegs 6-516
MaxDocIndexErrors 6-517
MaxErrorsAllowed 6-518
MaxHitCount 6-519
MaxIntegerAllowed 6-520

maxLength 6-521
MaxMergeBaseCount 6-522
MaxNumberConvertedHistoryShow 6-523
MaxNumRecursiveStepDefinitions 6-524
MaxQueryRows 6-525
MaxRequestThreadCount 6-526
MaxResults 6-527
MaxSavedSearchResults 6-528
MaxSearchConnections 6-529
MaxSizeResourceFileCacheMillions 6-530
MaxStandardDatabaseResults 6-531
MemoFieldSize 6-532
MinIntegerAllowed 6-533
MinMemoFieldSize 6-534
MinorRevSeq 6-535
MrSidLowestCompressionLevel 6-536
MSIE 6-537
MSPubexePath 6-538
MultiUpload 6-539
NetworkAdminGroup 6-540
NeverTimeoutForcedConversionsTemplateChange 6-541
NoAutomation 6-542
noInfoColumn 6-543
NoMatches 6-544
noMCPrefill 6-545
noRevisionsColumn 6-546
NoSearchConnectionCloseOnDocChange 6-547
NotationForRequiredFields 6-548
NotificationIntervalInDays 6-549
NotificationMaximum 6-550
NotificationQuery 6-551
NotifyExtras 6-553
NotifyTime 6-554

NtlmSecurityEnabled 6-555
NumAdditionalRenditions 6-556
NumConnections 6-557
OdbcDateFormat 6-558
OneMatch 6-559
OpenOfficeListenerPort 6-560
OpenOfficeListenerURL 6-561
OpenOfficePath 6-562
OptimizePDF 6-563
optionListKey 6-564
optionListName 6-565
optionListResultSet 6-567
optionListScript 6-568
optionListValueInclude 6-569
optionsAllowPreselect 6-571
optList 6-572
PageMakerExePath 6-574
PageParent 6-575
parseClbraProject 6-576
parseDataEntryDate 6-577
parseDate 6-578
parseDateWithPattern 6-580
PATH_INFO 6-581
PATH_TRANSLATED 6-582
PdfVersion 6-583
PLUGIN_DEBUG 6-584
pneNavigation 6-585
PostConversionTimeOut 6-586
PostprocessPDFPath 6-587
PostscriptPrinterName 6-588
PowerPointPrintOutput 6-589
PreconversionPath 6-590
PreconversionWaitTime 6-591

PreviewOutputExtension 6-592
PreviewPath 6-593
PrimaryWorkQueueTimeout 6-594
PrinterPortPath 6-595
Priority 6-596
prjUseCustomRouting 6-597
prjWizardTokens 6-598
prjWizardWorkflowField 6-599
ProcessExcelUrlHyperlinks 6-600
ProcessInDesignBookMarks 6-601
ProcessInDesignHyperlinks 6-602
ProcessInDesignInternalPDFThumbnail 6-603
ProcessInDesignLinkUpdate 6-604
ProcessInDesignPDFCompatibility50 6-605
ProcessPowerpointHyperlinks 6-606
ProcessVisioInternalHyperlinks 6-607
ProcessVisioUrlHyperlinks 6-608
ProcessWordBookMarks 6-609
ProcessWordComments 6-610
ProcessWordEmailLinks 6-611
ProcessWordFootEndNote 6-612
ProcessWordHeadings 6-613
ProcessWordHyperlinks 6-614
ProcessWordToc 6-615
ProcessWordUrlHyperlinks 6-616
ProfileCacheTimeoutInMins 6-617
ProviderClass 6-618
ProviderConfig 6-619
ProviderConnection 6-620
ProviderDir 6-621
ProviderType 6-622
proxiedBrowserFullCgiWebUrl 6-623
proxiedCgiWebUrl 6-624

pucDefaultTime 6-625
pucEnableOnSearch 6-626
pucHideTimeInDateFields 6-627
pucUseMonthLocale 6-628
pucUseWeekdayLocale 6-629
pucXOffset 6-630
pucYOffset 6-631
pucZIndex 6-632
QUERY_STRING 6-633
QueueDir 6-634
RecordsManagementDenyAuthorFreePassOnRMSecurity 6-635
RecordsManagementNumberOverwriteOnDelete 6-636
RedirectHostsFilter 6-637
regexMatches 6-638
regexReplaceAll 6-639
regexReplaceFirst 6-640
REMOTE_ADDR 6-641
REMOTE_HOST 6-642
REQUEST_METHOD 6-643
RequestAuditAdditionalVerboseFieldsList 6-644
RequestAuditIntervalSeconds1 6-645
RequestAuditIntervalSeconds2 6-646
RequestAuditListDepth1 6-647
RequestAuditListDepth2 6-648
RequestAuditSubClassificationField 6-649
requiredMsg 6-650
ResultsTitle 6-651
RmaAddDocWhereClauseForScreening 6-652
RMAHideExternalFieldsFromCheckInUpdate 6-653
RMAHideExternalFieldsFromSearchInfo 6-654
RmaNotifyDispReviewerAndCatAuthor 6-655
RmaNotifyReviewerAndAlternateReviewer 6-656
RolePrefix 6-657

rptDisplayMapValue 6-659
rsAddFields 6-660
rsAddRowCountColumn 6-661
rsAppend 6-662
rsAppendNewRow 6-663
rsAppendRowvalues 6-664
rsCopyFiltered 6-666
rsCreateResultSet 6-667
rsDeleteRow 6-668
rsDocInfoRowAllowDisclosure 6-669
rsExists 6-670
rsFieldByIndex 6-671
rsFieldExists 6-672
rsFindRowPrimary 6-673
rsFirst 6-674
rsInsertNewRow 6-675
rsIsRowPresent 6-676
rsMakeFromList 6-677
rsMakeFromString 6-680
rsMerge 6-682
rsMergeDelete 6-683
rsMergeReplaceOnly 6-684
rsNext 6-685
rsNumFields 6-686
rsNumRows 6-687
rsRemove 6-688
rsRename 6-689
rsRenameField 6-690
rsSetRow 6-691
rsSort 6-692
rsSortTree 6-693
SafeDir 6-695
SchemaAllowRecursiveFileMove 6-696

SchemaBuildDigestFile 6-697
SchemaPublishCheckInterval 6-698
SchemaPublishInterval 6-699
SchemaPublishMaximumInterval 6-700
SchemaPublishMinimumInterval 6-701
SchemaPublishRenameRetryInitialInterval 6-702
SchemaPublishRenameRetryIterations 6-703
SchemaPublishRetryInterval 6-704
SchemaPublishRetryIntervalMax 6-705
SchemaVerifyPublishing 6-706
SchemaWin32RenameDelayInterval 6-707
SCRIPT_NAME 6-708
ScriptDebugTrace 6-709
ScriptErrorTrace 6-710
SearchCacheCleanerInterval 6-711
SearchCacheHardLimit 6-712
SearchCacheMaxItemAge 6-713
SearchCacheSoftLimit 6-714
SearchCacheTrace 6-715
SearchConnectionWaitTimeout 6-716
SearchDebugLevel 6-717
SearchDir 6-718
SearchIndexerEngineName 6-719
SearchQueryFormat 6-720
SearchQueryTrace 6-721
SecondaryConversionStepName 6-722
SelectAccessMemberAllowManualEntry 6-723
SelectAccessMemberResultLimit 6-724
SelfRegisteredAccounts 6-725
SelfRegisteredRoles 6-726
SERVER_NAME 6-727
SERVER_PORT 6-728
SERVER_PROTOCOL 6-729

SERVER_SOFTWARE 6-730
setContentType 6-731
setCookie 6-732
setExpires 6-733
setHttpHeader 6-734
setMaxAge 6-735
SetPdfVersion 6-736
setResourceInclude 6-737
setValue 6-738
SharedDir 6-739
SharedWeblayoutDir 6-740
ShowCreateExternalMenu 6-741
showDocActions 6-742
ShowHyperlinkBox 6-743
ShowOnlyKnownAccounts 6-744
ShowRevisionsColumn 6-745
showSearchResultsCheckoutColumn 6-746
ShowWebDavUrlInBrowserRequest 6-747
SingleGroup 6-748
SkipMozillaMailToWorkaround 6-749
SmtpPort 6-750
SocketHostAddressSecurityFilter 6-751
SocketHostNameSecurityFilter 6-752
SourceID 6-753
SourcePath 6-754
SpecialAuthGroups 6-755
StatusCode 6-756
StatusMessage 6-757
StdPageWidth 6-758
stdSecurityCheck 6-759
strCenterPad 6-760
rptDisplayMapValue 6-761
strCommaAppendNoDuplicates 6-762

strConfine 6-763
StrConfineOverflowChars 6-765
strEquals 6-766
strEqualsIgnoreCase 6-767
strGenerateRandom 6-768
strIndexOf 6-769
strLeftFill 6-770
strLeftPad 6-771
strLength 6-772
strLower 6-773
strRemoveWs 6-774
strReplace 6-775
strReplaceIgnoreCase 6-776
strRightFill 6-777
strRightPad 6-778
strSubstring 6-779
strTrimWs 6-781
strUpper 6-782
StyleForRequiredFields 6-783
SubjectsDir 6-784
SubscriptionCollatedFieldList 6-785
SubscriptionMethod 6-786
SysAdminAddress 6-787
SystemDateFormat 6-788
SystemLocale 6-789
SystemReportsDir 6-791
SystemResourcesDir 6-792
SystemTemplatesDir 6-793
SystemTimeZone 6-794
TempDir 6-795
TemplateClass 6-796
TemplateFilePath 6-798
TemplateName 6-799

TemplateType 6-800
TextIndexerFilterValidateFormats 6-802
ThumbnailDPI 6-803
ThumbnailHeight 6-804
ThumbnailJPEGQuality 6-805
ThumbnailSizeLimit 6-806
ThumbnailSizingMethod 6-807
ThumbnailTemplate 6-808
ThumbnailWidth 6-809
TimeoutChangedTemplateForcedConversionsInMins 6-810
TimeoutMismatchedForcedConversionsInMins 6-811
TimeoutPerOneMegInSec 6-812
toInteger 6-813
trace 6-814
TraceConversionCacheCleanup 6-816
TraceIsVerbose 6-817
TraceResourceConflict 6-818
TraceResourceLoad 6-819
TraceResourceOverride 6-820
TraceSectionsList 6-821
UieHideSearchCheckboxes 6-824
UniversalSearchAllowFullText 6-825
UploadApplet 6-826
url 6-827
urlEscape7Bit 6-828
UseAccounts 6-829
UseAdobeDistiller 6-830
UseAdsi 6-831
UseAlchemy111 6-832
UseAutoCad2000 6-833
UseAutocadModelSpace 6-834
UseBellevueLook 6-835
UseCascadingStyleSheets 6-836

UseCustomModaling 6-837
UseDatabaseShortIndexName 6-838
UseDCLApplet 6-839
UseFourDigitYear 6-840
UseFullGroupName 6-841
UseGroupFilter 6-842
UseHtmlOrTextHighlightInfo 6-843
UseLightweightToPostscriptStep 6-844
UseLocalGroups 6-845
UseMicrosoftVM 6-846
UseNetscape 6-847
UseNtlm 6-848
UserAccounts 6-849
UserAddress 6-850
UserAppRights 6-851
UserCache 6-852
UserCacheTimeout 6-853
UserDefaultAccount 6-854
UseRedirectedOutput 6-855
UseRevisionCreateDate 6-856
UserFilteredSubjects 6-857
UserFullName 6-858
userHasAccessToAccount 6-859
userHasGroupPrivilege 6-860
userHasRole 6-861
UserIsAdmin 6-862
UserLanguageID 6-863
UserLocaleId 6-864
UserName 6-865
UserPublishCacheDir 6-866
UserRoles 6-867
UserSelectAppletColumnOption 6-868
UserSelectAppletDatabaseCaseSensitive 6-869

UserSelectAppletDatabaseIso8859DictionaryOrder 6-870
UserSelectAppletExternalAliasDefaultPermissions 6-871
UserSelectAppletExternalUserDefaultPermissions 6-872
UserSelectAppletForceEntriesToLowerCase 6-873
UserSelectAppletInternalAliasDefaultPermissions 6-874
UserSelectAppletInternalUserDefaultPermissions 6-875
UseSearchCache 6-876
UseSearchML 6-877
UseSecureLdap 6-878
UseSelfRegistration 6-879
UseShortNamesAlways 6-880
UseSSL 6-881
UseUserSelectApplet 6-882
UseVdk4CustomStyle 6-883
UseXmlEmbeddings 6-884
UseXmlExtensibleDTD 6-885
UseXmlNoDTD 6-886
UseXmlNoReference 6-887
UseXmlPStyleNames 6-888
UseXmlSchema 6-889
UseXmlSchemaDTD 6-890
UseXmlUrl 6-891
UseXpedioLook 6-892
utGetValue 6-893
utLoad 6-894
utLoadDocumentProfiles 6-895
utLoadResultSet 6-896
valueStyle 6-897
VaultDir 6-898
VDKSUMMARY 6-899
VerboseMode 6-900
VerityAppName 6-901
VerityAppSignature 6-902

VerityEncoding 6-903
VerityInstallDir 6-904
VerityLocale 6-905
WarnAboutTimeZone 6-906
watchdogInterval 6-907
WebBrowserPath 6-908
WeblayoutDir 6-909
WebProxyAdminServer 6-910
WebServerAuthOnly 6-911
wfAction 6-912
wfAddActionHistoryEvent 6-913
wfAdditionalExitCondition 6-914
wfAddUser 6-915
wfComputeStepUserList 6-916
wfCurrentGet 6-917
wfCurrentSet 6-918
wfCurrentStep 6-919
wfDisplayCondition 6-920
wfExit 6-921
wfGet 6-922
wfGetStepTypeLabel 6-923
wfIsFinishedDocConversion 6-925
wfIsNotifyingUsers 6-926
wfIsReleasable 6-927
wfJumpEntryNotifyOff 6-928
wfJumpMessage 6-929
wfJumpName 6-930
wfJumpReturnStep 6-931
wfJumpTargetStep 6-932
wfLoadDesign 6-933
wfMailSubject 6-934
wfMessage 6-935
wfNotify 6-936

wfParentList 6-937
wfReleaseDocument 6-938
wfSet 6-939
wfSetIsNotifyingUsers 6-940
WfStart 6-941
wfUpdateMetaData 6-942
WorkflowDir 6-943
WorkflowIntervalHours 6-944
xml 6-945
XMLEncodingMode 6-946
XMLTimeout 6-947
ZonedSecurityFields 6-948

付録 A: サード・パーティ・ライセンス

Apache Software License A-2
W3C® Software Notice and License A-2
Zlib License A-4
General BSD License A-5
General MIT License A-5
Unicode License A-6
その他の帰属 A-7

索引

1

はじめに

Idoc スクリプトは、固有のサーバーサイド・スクリプト言語です。このガイドでは、Idoc スクリプトの使用方法および構文について説明し、Idoc スクリプトの事前定義済の関数、変数および構成設定の詳細および例を示します。

Idoc スクリプトに関するさらなる詳細および使用方法は、次のマニュアルを参照してください。

- 『Services Reference Guide』
- 『Working with Content Server Components』
- 『The Definitive Guide to Content Server Development』Brian Huff 著 © 2006, Apress, Berkeley, CA

この項の内容は次のとおりです。

- 構成（1-2 ページ）
- このガイドについて（1-2 ページ）
- 対象読者（1-3 ページ）
- 表記規則（1-3 ページ）

構成

このガイドは、次の章で構成されています。

- 第 2 章「新機能」では、Idoc スクリプトの新規および非推奨の変数および関数の一覧を示します。
- 第 3 章「Idoc スクリプトの使用」では、Content Server での Idoc スクリプトの使用方法を説明します。
- 第 4 章「タイプ別 Idoc スクリプト」では、異なるタイプの Idoc スクリプト変数および関数について説明し、変数および関数をタイプ別に分類します。
- 第 5 章「使用方法別 Idoc スクリプト」では、Idoc スクリプトの変数および関数を一般的な使用方法に応じて分類します。
- 第 6 章「Idoc スクリプト索引」では、Idoc スクリプトの各関数および変数について詳しく説明します。

このガイドの末尾に索引があります。

このガイドについて

このガイドでは、Content Server のサービスに関する詳細情報を提供します。これらのサービスは、コンテンツ・サーバーによって実行される関数またはプロシージャです。コンテンツ・サーバーのサービスをコール（サービスをリクエスト）することが、コンテンツ・サーバーと通信またはデータベースにアクセスする唯一の方法です。このガイドでは、サービスの使用方法および構文について説明し、標準（事前定義済）の Content Server サービスの詳しい説明および例を示します。

注意：このガイドの情報は、Content Server 10*g*R3 に基づいて記載されています。この情報は、製品技術の進化、ハードウェアおよびオペレーティング・システムの作成および変更に伴い変更される可能性があります。ブラウザ、データベース、Web サーバーおよびオペレーティング・システムの技術的な性質により、オラクル社では、サード・パーティ製品のすべてのバージョンおよび機能との互換性を保証できません。

注意：このリファレンス・ガイドは、Software Developer's Kit（SDK）の一部です。Content Server のカスタマイズおよび統合の詳細は、SDK を参照してください。

対象読者

このガイドは、サービスを使用してソフトウェアをカスタマイズする必要がある開発者および Content Server の機能にアクセスする必要があるアプリケーション開発者を対象としています。

表記規則

このガイドでは、次の表記規則が使用されています。

- *<Install_Dir>/* という表記は、コンテンツ・サーバー・インスタンスがインストールされているシステム上の場所を参照するために使用されます。
- スラッシュ（/）は、パス名のディレクトリ・レベルの区切りとして使用されます。スラッシュは、ディレクトリ名の後に常に表示されます。
- 注意、技術ヒント、重要な通知および警告には、次の表記規則が適用されます。

記号	説明
	これは注意です。特別な注意事項を示すために使用されます。
	これは技術ヒントです。タスクをより簡単に実行するための情報を示します。
	これは重要な通知です。必要な手順または情報を示すために使用されます。
	これは警告です。データの損失または重大なシステム問題の原因となる可能性がある情報を示すために使用されます。

2

新機能

この項では、バージョン 10*g*R3 の新情報について説明します。

- 使用に関する最新情報（2-1 ページ）
- 10gR3 で新規 / 修正された Idoc スクリプト（2-2 ページ）
- 7.5.1 で新規 / 修正された Idoc スクリプト（2-7 ページ）

使用に関する最新情報

8.0 では、Idoc スクリプトのネーミング規則は、3-2 ページの「Idoc のネーミング規則」に記載されています。

また、有効範囲付ローカル変数に関する情報も記載されています。詳細は、3-40 ページの「有効範囲付のローカル変数」を参照してください。

10*g*R3 で新規 / 修正された Idoc スクリプト

バージョン 10*g*R3 で修正または新規にドキュメントされた Idoc スクリプトの変数および関数は次のとおりです。

- AllowAllRedirectUrl（6-22 ページ）
- AllowConcurrentUpdate（6-26 ページ）
- AllowForceDelete（6-27 ページ）
- AllowRetentionPeriodWithoutCutoff（6-34 ページ）
- AutoNumberWidth（6-52 ページ）
- clearSchemaData（6-73 ページ）
- computeDocUrl（6-99 ページ）
- CookieAuthTimeoutInDays（6-107 ページ）
- CookieProxyPassword（6-108 ページ）
- coreContentOnly（6-109 ページ）
- dcShowExportLink（6-125 ページ）
- DebugMode（6-128 ページ）
- DebugStdConversion（6-129 ページ）
- DefaultHtmlRefreshTimeoutInSeconds（6-137 ページ）
- DefaultMasterDomain（6-138 ページ）
- DisableAmberLayouts（6-149 ページ）
- DisableClassicLayoutPublishing（6-150 ページ）
- DisableCollegiateSkinPublishing（6-151 ページ）
- DisableContentProfiles（6-152 ページ）
- DisableDBLongIntegerField（6-153 ページ）
- DisableStellent05SkinPublishing（6-161 ページ）
- DisableStellentSkinPublishing（6-162 ページ）
- DisableTopMenusPublishing（6-163 ページ）
- DisableTraysPublishing（6-164 ページ）
- DisableWindowsPublishing（6-165 ページ）
- dpGet（6-183 ページ）

- dpPromote（6-184 ページ）
- dpPromoteRs（6-185 ページ）
- dpSet（6-186 ページ）
- encodeHtml（6-201 ページ）
- fieldEditWidth（6-220 ページ）
- fieldMaxLength（6-225 ページ）
- formatDateOnly（6-247 ページ）
- generateUniqueId（6-259 ページ）
- getFieldConfigValue（6-264 ページ）
- getFieldViewDisplayValue（6-265 ページ）
- getFieldViewValue（6-266 ページ）
- getFreeMemory（6-267 ページ）
- getHelpPage（6-268 ページ）
- getOptionListSize（6-269 ページ）
- getParentValue（6-270 ページ）
- getRequiredMsg（6-271 ページ）
- getTotalMemory（6-273 ページ）
- getViewValue（6-279 ページ）
- getViewValueResultSet（6-280 ページ）
- HasSiteStudio（6-288 ページ）
- HideVitalReview（6-292 ページ）
- HtmlDataInputFilterLevel（6-295 ページ）
- htmlRefreshTimeout（6-297 ページ）
- htmlRefreshUrl（6-298 ページ）
- HttpLayoutRoot（6-318 ページ）
- HttpSystemHelpRoot（6-324 ページ）
- IdcAdminServerHostName（6-327 ページ）
- IdcServerBindAddress（6-340 ページ）
- IndexableRenditions（6-351 ページ）
- IndexerPath（6-354 ページ）（場所の変更）

- indexerSetCollectionValue（6-357 ページ）
- IndexerStyleDir（6-358 ページ）
- IsAllowIndexRenditions（6-373 ページ）
- IsAllowIndexVault（6-374 ページ）
- IsDpSubmitErrorFatal（6-387 ページ）
- isLayoutEnabled（6-415 ページ）
- IsSavedWfCompanionFile（6-437 ページ）
- isValidateFile（6-451 ページ）
- IsWebServerPagesOnly（6-453 ページ）
- IsWorkflow（6-455 ページ）
- IsXml（6-456 ページ）
- isZoneField（6-458 ページ）
- lcCaption（6-474 ページ）
- LkDisableOnRebuild（6-480 ページ）
- LkDisableReferenceInfo（6-481 ページ）
- LkDisallowConversionFormats（6-482 ページ）
- LkIsSecureSearch（6-484 ページ）
- LkReExtractOnRebuild（6-485 ページ）
- LkRefreshBatchSize（6-486 ページ）
- LkRefreshErrorsAllowed（6-489 ページ）
- LkRefreshErrorPercent（6-488 ページ）
- LkRefreshErrorThreshold（6-490 ページ）
- LMDefaultLayout（6-491 ページ）
- LMDefaultSkin（6-492 ページ）
- loadDocumentProfile（6-497 ページ）
- loadEnterpriseSearchCollections（6-499 ページ）
- loadEnterpriseSearchResults（6-500 ページ）
- loadUserMetaDefinition（6-503 ページ）
- loadSchemaData（6-501 ページ）
- localPageType（6-506 ページ）

- MaxSavedSearchResults（6-528 ページ）
- NotificationIntervalInDays（6-549 ページ）
- NotificationMaximum（6-550 ページ）
- NotifyExtras（6-553 ページ）
- NotifyTime（6-554 ページ）
- parseClbraProject（6-576 ページ）
- pucDefaultTime（6-625 ページ）
- pucEnableOnSearch（6-626 ページ）
- pucHideTimeInDateFields（6-627 ページ）
- pucUseMonthLocale（6-628 ページ）
- pucUseWeekdayLocale（6-629 ページ）
- pucXOffset（6-630 ページ）
- pucYOffset（6-631 ページ）
- pucZIndex（6-632 ページ）
- RecordsManagementNumberOverwriteOnDelete（6-636 ページ）
- RecordsManagementDenyAuthorFreePassOnRMSecurity（6-635 ページ）
- RedirectHostsFilter（6-637 ページ）
- RmaAddDocWhereClauseForScreening（6-652 ページ）
- RMAHideExternalFieldsFromCheckInUpdate（6-653 ページ）
- RMAHideExternalFieldsFromSearchInfo（6-654 ページ）
- RmaNotifyDispReviewerAndCatAuthor（6-655 ページ）
- RmaNotifyReviewerAndAlternateReviewer（6-656 ページ）
- rptDisplayMapValue（6-659 ページ）
- rsExists（6-670 ページ）
- rsFieldByIndex（6-671 ページ）
- rsFieldExists（6-672 ページ）
- rsIsRowPresent（6-676 ページ）
- rsNumFields（6-686 ページ）
- rsNumRows（6-687 ページ）
- rsRemove（6-688 ページ）

- SchemaAllowRecursiveFileMove（6-696 ページ）
- SearchCacheHardLimit（6-712 ページ）
- SearchCacheSoftLimit（6-714 ページ）
- SearchIndexerEngineName（6-719 ページ）
- ShowCreateExternalMenu（6-741 ページ）
- strCommaAppendNoDuplicates（6-762 ページ）
- strGenerateRandom（6-768 ページ）
- strReplaceIgnoreCase（6-776 ページ）
- UniversalSearchAllowFullText（6-825 ページ）
- urlEscape7Bit（6-828 ページ）
- UseCustomModaling（6-837 ページ）
- UseRedirectedOutput（6-855 ページ）
- userHasAccessToAccount（6-859 ページ）
- UserLanguageID（6-863 ページ）
- UserLocaleId（6-864 ページ）
- userHasGroupPrivilege（6-860 ページ）
- utLoadDocumentProfiles（6-895 ページ）
- wfAddActionHistoryEvent（6-913 ページ）
- wfComputeStepUserList（6-916 ページ）
- wfIsFinishedDocConversion（6-925 ページ）
- wfIsNotifyingUsers（6-926 ページ）
- wfSetIsNotifyingUsers（6-940 ページ）

7.5.1 で新規 / 修正された Idoc スクリプト

バージョン 7.5.1 で修正または新規にドキュメントされた Idoc スクリプトの変数および関数は次のとおりです。

- ClusterBinDirRule（6-75 ページ）
- ClusterNodeAddress（6-76 ページ）
- ClusterNodeName（6-77 ページ）
- IsSoap（6-438 ページ）
- MaxStandardDatabaseResults（6-531 ページ）
- SelectAccessMemberAllowManualEntry（6-723 ページ）
- SelectAccessMemberResultLimit（6-724 ページ）
- setResourceInclude（6-737 ページ）
- SkipMozillaMailToWorkaround（6-749 ページ）

次の変数はドキュメントに記載されていませんが、使用可能です。これらは、スキーマ・キャッシュへのアクセスに使用され、スキーマの操作でユーザーが設定する変数ではありません。

```
getViewValue
getViewDisplayValue
getFieldViewValue
generateUniqueID
getViewValuesResultSet
getOtherTableColumn
getMyTableColumn
getParentValue
getFieldViewDisplayValue
```

次の変数は、バージョン 7.5.1 で削除されました。

```
CompactIntervalDays
```

3

Idoc スクリプトの使用

Idoc スクリプトは、Content Server 用のサーバーサイドのカスタム・スクリプト言語です。Idoc スクリプトでは、変数の参照、条件に基づいて HTML ページのコンテンツをインクルード、および問合せから返された結果に対するループ処理を実行できます。Idoc スクリプトはクライアント側ではなく、サーバー側で評価されるため、ページ要素の処理は、ブラウザがリクエストした後、リクエストしたページがクライアントに返される前に実行されます。

Idoc スクリプトは主に、HTML テンプレートおよび構成設定の表示で使用されます。

この章の内容は次のとおりです。

- Idoc のネーミング規則（3-2 ページ）
- Idoc スクリプトの構文（3-2 ページ）
- Idoc スクリプトの使用方法（3-3 ページ）
- 特殊キーワード（3-18 ページ）
- 演算子（3-22 ページ）
- メタデータ・フィールド（3-26 ページ）
- Verity Integration での日付 / 時間（3-37 ページ）
- MergeInclude を使用したレスポンスのフォーマット（3-38 ページ）
- 有効範囲付のローカル変数（3-40 ページ）

IDOC のネーミング規則

Idoc 変数（構成変数または環境変数とも呼ばれる）は、Idoc スクリプトおよび構成ファイルで使用できます。

通常、変数が構成の一部である場合、その変数は大文字で始まります。一般に、config.cfg ファイルまたは intradoc.cfg ファイルに指定される変数は、先頭の 1 文字が大文字です。例については、6-203 ページの「EnterpriseSearchAsDefault」を参照してください。サービス・リクエストの多くのパラメータも、大文字で始まります。

ページで定義される変数（導出され、std_page.htm などのファイルで使用される変数）は、小文字で始まります。例については、6-211 ページの「executeService」を参照してください。変数は環境変数またはサービス変数から計算され、表示のために使用されます。

変数をオブジェクトの定義に使用する場合、その変数は小文字で始まり、定義するオブジェクトのタイプを示します。ワークフロー固有の変数の例は、6-939 ページの「wfSet」を参照してください。また、すべての関数は小文字で始まり、多くの関数の先頭には、関数のタイプを表す接頭辞が付けられています。たとえば、ほとんどの文字列操作関数は `str` で始まり、また ResultSet 関数は `rs` で始まります。

カスタム・メタデータ・フィールド以外のすべてのデータベース列名は、小文字の `d` で始まります。コンテンツ・サーバーによって作成されるカスタム・メタデータのすべてのデータベース列名は、小文字の `x` で始まります。

IDOC スクリプトの構文

Idoc スクリプトは、次の基本的な構文規則に従います。

- Idoc スクリプトのタグ（3-2 ページ）
- Idoc スクリプトのコメント（3-3 ページ）

Idoc スクリプトのタグ

Idoc スクリプトのすべてのコマンドは、デリミタ `<$` で始まり、デリミタ `$>` で終わります。次に例を示します。

```
<$dDocTitle$>
<$if UseGuiWinLook and isTrue(UseGuiWinLook)$>
```

技術ヒント：HCSP または HCSF ページで Idoc スクリプトを使用する場合、Idoc スクリプトのタグには構文 `<!--$script-->` を使用する必要があります。詳細は、『Dynamic Server Pages Guide』を参照してください。

Idoc スクリプトのコメント

Idoc スクリプト・コードでは、標準 HTML コメントまたは Idoc スクリプトのコメントを使用できます。

```
<!-- HTML Comment -->
<$c = "Idoc Script Comment"$>
```

IDOC スクリプトの使用方法

Idoc スクリプトの基本的な使用方法は次の 6 通りです。

- インクルード（3-4 ページ）を使用して、Idoc スクリプトおよび HTML のコード・ピースを再利用できます。
- 変数（3-7 ページ）を使用して、変数値を定義および置換できます。
- 関数（3-10 ページ）を使用して、文字列比較ルーチン、文字列操作ルーチン、日付フォーマット、ResultSet 操作などのアクションを実行できます。
- 条件（3-11 ページ）を使用して、*if* および *else* 句を評価して、アセンブルされたページにコードを含めるか、除外することができます。
- ループ（3-13 ページ）を使用して、問合せから返された ResultSet の各行に対してコードを繰り返し実行できます。
- 管理インタフェース（3-16 ページ）を使用して、Idoc スクリプトをコンテンツ・サーバーのアプレットおよびカスタマイズで使用できます。

インクルード

インクルードは、コンテンツ・サーバーの Web ページの作成のために使用するコード・ピースを定義します。インクルードは、リソース・ファイルに一度定義され、必要に応じてそれを複数のテンプレート・ファイルで参照します。コンテンツ・サーバーでは、インクルードを頻繁に使用します。インクルードは、コンテンツ・サーバー Web サイトのほとんどすべてのページで使用されます。

注意：インクルードにより、コンポーネント・アーキテクチャおよび動的サーバー・ページを使用してコンテンツ・サーバー・インスタンスを簡単にカスタマイズできます。インクルードおよびカスタマイズの詳細は、『Working with Content Server Components』を参照してください。

- インクルードは、次の書式を使用して HTM リソース・ファイルで定義します。

```
<@dynamichtml name@>
    code
<@end@>
```

- 次の Idoc スクリプト書式を使用して、HTM テンプレート・ファイルからインクルードをコールします。

```
<$include name$>
```

- インクルードには、JavaScript、Java アプレット、カスケード・スタイル・シートおよびコメントを含む Idoc スクリプトおよび有効な HTML コードを含めることができます。
- インクルードは、コール元と同じファイルに定義するか、別個のファイルに定義できます。
- 標準のインクルードは、<*install_dir*>/shared/config/resources/std_page.htm ファイルに定義されます。

詳細は、次の項を参照してください。

- インクルードの例（3-5 ページ）
- super タグ（3-6 ページ）
- super タグの例（3-7 ページ）
- 特殊キーワード（3-18 ページ）
- キーワードと関数の対比（3-19 ページ）

インクルードの例

最も一般的なインクルードの 1 つは、本体の定義要素 *<@dynamichtml body_def>* です。このインクルードは、ページの背景色、ハイパーリンクの色および背景イメージを設定します。次のコードは、*<install_dir>*/shared/config/resources/std_page.htm ファイルにあります。

図 3-1　　インクルードの定義例

```
<@dynamichtml body_def@>
<!--Background image defined as part of body tag--->
<body
    <$if background_image$>
        background="<$HttpImagesRoot$><$background_image$>"
    <$elseif colorBackground$>
        bgcolor="<$colorBackground$>"
    <$endif$>
    <$if xpedioLook$>
        link="#663399" vlink="#CC9900"
    <$else$>
        link="#000000" vlink="#CE9A63" alink="#9C3000"
    <$endif$>
    marginwidth="0" marginheight="0" topmargin="0" leftmargin="0"
>
<@end@>
```

ほとんどの標準テンプレート・リソース・ファイル（たとえば、*<install_dir>*/shared/config/templates/pne_home_page.htm）には、ページの先頭近くに次の Idoc スクリプト・コードが含まれます。

```
<$include body_def$>
```

このコードを含むテンプレート・ページを解決するとき、コンテンツ・サーバーは `<@dynamichtml body_def@>` 定義を探し、プレースホルダ・コードを定義されたコードに置き換えます。

super タグ

super タグを使用して、既存のインクルードに対する例外を定義します。super タグを指定すると、既存のインクルードで開始した後、指定したコードによって既存インクルードへの追加または修正が行われます。

- super タグの構文は次のとおりです。

```
<@dynamichtml my_resource@>
    <$include super.my_resource$>
    exception code
<@end@>
```

- super タグを使用して、標準インクルードまたはカスタム・インクルードを参照できます。super タグは、最後にロードされたインクルードを取り込みます。
- 複数の super タグを指定して、最後のバージョンより前にロードされたインクルードをコールできます。たとえば、2 つの異なるコンポーネントで、標準インクルードの *body_def* に対する例外を設定する場合は、最後にロードされるインクルードで次の構文を使用します。

```
<$include super.super.body_def$>
```

警告 : 1 つのインクルードに複数の super タグを使用する場合は、リソースのロード元およびロードされる順序を把握しておく必要があります。

- super タグは、大きなインクルードに対して小さなカスタマイズを行う場合、またはソフトウェアのバージョンごとに変更される可能性がある標準コードをカスタマイズする場合に特に役立ちます。Content Server の新バージョンにアップグレードするとき、super タグにより、インスタンスのカスタマイズに必要な特定のコードのみが修正され、コンポーネントで最新バージョンのインクルードを確実に使用できます。

関連項目

– super タグの例（3-7 ページ）

super タグの例

この例では、あるコンポーネントで *my_resource* インクルードが次のように定義されています。

```
<@dynamichtml my_resource@>
    <$a = 1, b = 2$>
<@end@>
```

後からロードされる別のコンポーネントでは、super タグを使用して *my_resource* インクルードを拡張しています。次の拡張によって、"a" には値 1 が割り当てられ、"b" に値 3 が割り当てられます。

```
<@dynamichtml my_resource@>
    <$include super.my_resource$>
    <!--Change "b" but not "a" -->
    <$b = 3$>
<@end@>
```

関連項目

– super タグ（3-6 ページ）

変数

変数により、変数の値を定義および置換できます。

次の項では、Idoc スクリプトの変数の操作方法を説明します。

- 変数の作成（3-8 ページ）
- 変数の参照（3-8 ページ）
- 値の割当て（3-8 ページ）
- カンマをセパレータとして使用（3-8 ページ）
- 条件での変数の参照（3-9 ページ）
- 変数の置換順序（3-9 ページ）
- 標準変数（3-9 ページ）

変数の作成

Idoc スクリプトの変数は、次のいずれかの方法で作成されます。

- コンテンツ・サーバーに事前定義済の変数が多数あります。
- 独自のカスタム変数を定義できます。
- 一部の変数値は、問合せおよびサービスを使用して生成する必要があります。（一部の変数の情報は、コンテンツ・サーバー・データベースから自動的に取得できないため、問合せおよびサービスを定義して要求する必要があります。）

Idoc スクリプト変数のタイプの詳細は、第 4 章「タイプ別 Idoc スクリプト」を参照してください。

変数の参照

テンプレートおよびその他のリソース・ファイルの変数を参照するには、次の Idoc スクリプト・タグを使用します。

```
<$variable_name$>
```

値の割当て

- 変数に値を割り当てるには、次の構造を使用します。

  ```
  <$variable=value$>
  ```

  たとえば、`<$i=0$>` は、変数 *i* に値 0 を割り当てます。

- 次のような書式で名前と値のペアを使用して、環境リソース（CFG）ファイルに変数値を定義することもできます。

  ```
  variable=value
  ```

  たとえば、標準の構成変数は、*<install_dir>*/config/config.cfg ファイルに定義されます。

カンマをセパレータとして使用

Idoc スクリプトでは、1 つのスクリプト・ブロック内でのカンマ区切りの複数の句をサポートします。

たとえば、`<$a=1$>` と `<$b=2$>` のような 2 つの個別の文ではなく、`<$a=1,b=2$>` を使用できます。

条件での変数の参照

次の構造を使用して、変数の存在を評価できます。

```
<$if variable_name$>
```

変数が定義されている場合、この条件は TRUE になります。変数が定義されていないか、空（NULL）文字列として定義されている場合は、FALSE になります。

このタイプの参照の一般的な使用方法の例は、3-12 ページの「条件の例」を参照してください。

関連項目

– 条件（3-11 ページ）

変数の置換順序

サービス・リクエストを満たすために変数値が必要な場合は、DataBinder にキャッシュされたデータが、次のデフォルトの順序で評価されます。

1. LocalData
2. アクティブな ResultSet
3. 非アクティブな ResultSet
4. Environment

注意： DataBinder の詳細は、『Component Architecture Guide』を参照してください。

標準変数

特殊な評価ロジック（条件付き動的変数（4-2 ページ）など）を持たない標準変数は、#active（3-18 ページ）キーワード接頭辞の使用と等価です。

たとえば、`<$variable$>` タグは、`<$#active.variable$>` と等価です。ただし、変数が見つからない場合、*#active* が明示的に指定されていないと、コンテンツ・サーバーのデバッグ出力にエラー報告が表示されます。

関数

Idoc スクリプトには、多数の組込みグローバル関数があります。関数は、文字列比較ルーチン、文字列操作ルーチン、日付フォーマット、ResultSet 操作などのアクションを実行します。一部の関数は、結果も返します（計算や比較の結果など）。

関数に情報を渡すには、関数名の後に情報をカッコで囲んで指定します。関数に渡す情報のことをパラメータと呼びます。パラメータを取らない関数、1 つのパラメータを取る関数、および複数のパラメータを取る関数があります。また、関数の使用方法によってパラメータの数が異なる関数もあります。

Idoc スクリプト関数の完全なリストは、4-21 ページの「グローバル関数」を参照してください。

関連項目

– グローバル関数（4-21 ページ）
– キーワードと関数の対比（3-19 ページ）
– パーソナライズ関数（3-10 ページ）

パーソナライズ関数

パーソナライズ関数は、コンテンツ・サーバーのパーソナライズ・ファイル（「ユーザー・トピック」ファイルとも呼ばれる）に定義されたユーザー・プロパティを参照します。各ユーザーのユーザー・プロファイルの設定、左側のナビゲーション・バーの個人用リンク、およびキュー内のワークフローの情報は、ユーザー・トピック・ファイルにすべて定義されます。ユーザー・トピック・ファイルは、<*install_dir*>/data/users/profiles/*us*/*username*/ ディレクトリにある HDA ファイルです。

次のグローバル関数は、ユーザー・トピック・ファイルを参照します。

❖ utGetValue（6-893 ページ）
❖ utLoad（6-894 ページ）
❖ utLoadResultSet（6-896 ページ）

たとえば、ユーザーの左側のナビゲーション・バーにある「Portal Design」リンクは、*pne_nav_userprofile_links* インクルード（*<install_dir>*/shared/config/resources/std_page.htm リソース・ファイル）の次のコードから生成されます。このリンクは、*<install_dir>*/data/users/profiles/*us*/*username*/pne_portal.hda ファイルの *portalDesignLink* プロパティが TRUE の場合に表示されます。

```
<$if utGetValue("pne_portal", "portalDesignLink") == 1$>
    <$hasUserProfileLinks=1$>
    <tr>
    <td colspan=2 nowrap align="left">
        <a class=pneLink href="<$HttpCgiPath$>?IdcService=GET_PORTAL_PAGE
        &Action=GetTemplatePage&Page=PNE_PORTAL_DESIGN_PAGE">
        <$lc("wwPortalDesign")$></a>
    <td>
    </tr>
<$endif$>
```

条件

条件では、if および else 句を使用して、アセンブルされたページにコードを含めるか、除外することができます。

❖ 次の Idoc スクリプトのキーワードを使用して、条件を評価します。

- `<$if condition$>`
- `<$else$>`
- `<$elseif condition$>`
- `<$endif$>`

❖ 条件句では、次の一般的な構造を使用します。

```
<$if conditionA$>
    <!--Code if conditionA is true-->
<$elseif conditionB$>
    <!--Code if conditionB is true-->
<$else$>
    <!--Code if neither conditionA nor conditionB are true-->
<$endif$>
```

❖ 条件式には任意の Idoc スクリプト関数または変数を使用できます。

詳細は、3-9 ページの「条件での変数の参照」を参照してください。

❖ ブール演算子（3-25 ページ）を使用して、条件句を結合できます。たとえば、次のように and 演算子を使用できます。

```
<$if UseBellevueLook and isTrue(UseBellevueLook)$>
```

- 条件式が、HTML ページに含めることが可能な ResultSet の名前であり、その ResultSet に 1 つ以上の行が存在する場合、条件句は true を返します。これにより、ResultSet に行が存在する場合にのみ、テンプレート・ページに ResultSet の情報が表示されます。
- 特殊な計算をトリガーしない条件句は、#active（3-18 ページ）接頭辞を使用して評価されます。値が NULL でなく、空でない文字列か 0 以外の整数のいずれかである場合、結果は true です。

条件付のコードの例は、3-12 ページの「条件の例」を参照してください。

条件の例

この例では、変数 *xDepartment* の値に応じて、表のセル `<td>` が定義されます。

```
<$if xDepartment$>
    <td><$xDepartment$></td>
<$else$>
    <td>Department is not defined.</td>
<$endif$>

<$xDepartment=""$>
```

- *xDepartment* の値が定義されている場合、表のセルに *xDepartment* の値が含まれます。
- *xDepartment* の値が定義されていないか、空（NULL）文字列である場合、表のセルの内容としてメッセージが書き込まれます。
- コードの最終行は、*xDepartment* 変数をクリアして空の文字列にリセットします。

ループ

ループ構造では、同じコードの可変回の実行が可能になります。Idoc スクリプトでは、ループを実行する方法は 2 通りあります。

- ResultSet のループ（3-13 ページ）
- while ループ（3-14 ページ）

ループ構造の終了に関する詳細は、3-15 ページの「ループの終了」を参照してください。

ResultSet のループ

ResultSet のループでは、問合せから返された ResultSet の各行に対して一連のコードを繰り返し実行できます。次の構文を使用して、ループ処理を行う ResultSet の名前を変数で指定します。

```
<$loop ResultSet_name$>
    code
<$endloop$>
```

- ResultSet 内の各行で 1 回ずつ、`<$loop$>` タグと `<$endloop$>` タグの間のコードが繰り返されます。
- ResultSet のループ処理の中で、getValue（6-275 ページ）関数を使用して ResultSet から値を取得できます。値の置換は、ループ内で現在アクセスされている行に依存します。
- ResultSet のループ処理中は、ResultSet がアクティブになり、変数および条件文を評価する際に他の ResultSet に優先します。
- `<$loop$>` タグは、ResultSet を指す変数に対するループ処理には使用できません。この場合は、rsFirst（6-674 ページ）および rsNext（6-685 ページ）関数を使用して、ResultSet に対するループ処理を手動で行う必要があります。

  たとえば、ResultSet に対するループ処理に次のコードは使用できません。

```
<$name="SearchResults"$>
<$loop name$>
    <!--output code-->
<$endloop$>
```

  かわりに、次のコードを使用します。

```
<$name="SearchResults"$>
<$rsFirst(name)$>
<$loopwhile getValue(name, "#isRowPresent")$>
    <!--output code-->
    <$rsNext(name)$>
<$endloop$>
```

ResultSet のループの例

この例では、*SearchResults* という ResultSet に対するループ処理によって検索結果表が作成されます。SearchResults は、GET_SEARCH_RESULTS サービスによって生成されます。

図 3-2　　ResultSet ループの例

```
<$QueryText="dDocType <matches> 'ADACCT'"$>
<$executeService("GET_SEARCH_RESULTS")$>
<table>
    <tr>
        <td>Title</td><td>Author</td>
    </tr>
<$loop SearchResults$>
    <tr>
        <td><a href="<$SearchResults.URL$>">
        <$SearchResults.dDocTitle$></a></td>
        <td><$SearchResults.dDocAuthor$></td>
    </tr>
<$endloop$>
</table>
```

関連項目

- rsFirst（6-674 ページ）
- rsNext（6-685 ページ）
- ループの終了（3-15 ページ）

while ループ

while ループにより、条件付のループを作成できます。次に while ループの構文を示します。

```
<$loopwhile condition$>
    code
<$endloop$>
```

- 条件式の結果が true の場合、`<$loopwhile$>` タグと `<$endloop$>` タグの間のコードが実行されます。
- ループ内のすべてのコードを実行した後、ループの先頭に制御が戻り、条件式が再び評価されます。
  - 結果が true の場合は、コードを再び実行します。
  - 結果が false の場合は、ループを終了します。

while ループの例

この例では、*abc* という変数が、ループを通過するたびに 2 ずつ増加します。6 回目になると（*abc* は 10 に等しい）、条件式は true でないため、ループを終了します。

図 3-3　　while ループの例

```
<$abc=0$>
<$loopwhile abc<10$>
    <$abc=(abc+2)$>
<$endloop$>
```

関連項目

– ループの終了（3-15 ページ）

ループの終了

Idoc スクリプトには、ResultSet ループまたは while ループを終了する 2 つのタグがあります。

- `<$endloop$>` は、次回の処理のためにループの先頭に制御を戻します。すべてのループは、`<$endloop$>` タグでクローズする必要があります。
- `<$break$>` により、最も内側のループが終了します。制御は、ループの end に続く最初の文から再開されます。

管理インタフェース

Idoc スクリプトは、管理インタフェースの次のような領域で使用できます。

- ワークフロー管理（3-16 ページ）
- Web レイアウト・エディタ（3-17 ページ）
- バッチ・ローダー（3-17 ページ）
- アーカイバ（3-17 ページ）
- システム・プロパティ（3-17 ページ）
- 検索式（3-18 ページ）
- 電子メール（3-18 ページ）

ワークフロー管理

ワークフロー管理ツールでは、Idoc スクリプトを使用して次のものを定義できます。

- ステップ・イベント
- ジャンプ・メッセージ
- 追加の終了条件
- トークン
- ジャンプのカスタム効果

たとえば、次のステップ・エントリ・スクリプトは、*Secure* セキュリティ・グループのドキュメントをワークフローの次のステップに送ります。

```
<$if dSecurityGroup like "Secure"$>
    <$wfSet("wfJumpName", "New")$>
    <$wfSet("wfJumpTargetStep", wfCurrentStep(1))$>
    <$wfSet("wfJumpEntryNotifyOff", "0")$>
<$endif$>
```

詳細は、4-37 ページの「ワークフロー・スクリプト」を参照してください。

Web レイアウト・エディタ

Web レイアウト・エディタでは、ページのタイトル、ページの説明、URL の説明、問合せ結果ページおよびコンテンツの問合せで Idoc スクリプトを使用できます。

次に例を示します。

- 問合せ結果のページ定義で Idoc スクリプトを使用して、検索結果表の各行の内容を指定できます。
- 検索結果に 7 日以内のすべてのコンテンツ・アイテムが返されるように設定するには、次のような検索問合せを定義できます。
  `dInDate > '<$dateCurrent(-7)$>'`
- 現行ユーザーに基づいた結果を返すレポートを定義するには、レポートの問合せ式の一部を `User Name is <$UserName$>` として定義します。

詳細は、『システム設定およびプロセスの管理』を参照してください。

バッチ・ローダー

バッチ・ローダーでは、マッピング・ファイルで Idoc スクリプトを使用して、BatchBuilder ユーティリティでのファイル・レコードのメタデータの決定方法を指定できます。詳細は、『システム設定およびプロセスの管理』を参照してください。

アーカイバ

アーカイバでは、次のような領域で Idoc スクリプトを使用できます。

- **問合せ値のエクスポート。**たとえば、1 年を超える古いコンテンツをアーカイブするには、`<$dateCurrent(-365)$>` をリリース日付の値として使用できます。
- **値マップの出力値。**たとえば、インポートしたすべてのリビジョンの失効日を 1 週間後に設定するには、`<dateCurrent(7)$>` を出力値として使用できます。

詳細は、『System Migration Guide』を参照してください。

システム・プロパティ

Admin Server で、または Inbound Refinery の「Local Configuration」画面または「Shared Configuration」画面で、コンテンツ・サーバーの System Properties ユーティリティの値を設定すると、実際には、Idoc スクリプトの構成値が設定されます。詳細は、各アプリケーションのマニュアルを参照してください。

検索式

問合せ式で Idoc スクリプトを使用できます。たとえば、次の式を URL で使用して、アドオン・アプリケーションの Verity Integration を使用して検索を実行できます。

```
QueryText=dDocAuthor <matches> '<$UserName$>'
QueryText=dInDate > <$dateCurrent(-7)$>
```

電子メール

Idoc スクリプトを使用して、コンテンツ・サーバーで送信される電子メール通知の件名行および本文をカスタマイズできます。次に例を示します。

```
Hello, <$UserName$>.Content item <$dDocName$> requires your review.
```

特殊キーワード

Idoc スクリプトで、次のキーワードには特殊な意味があります。

キーワード	例	説明
#active	<$#active.*variable*$>	DataBinder から、指定された変数の値を取得します。デフォルトの検索順序は次のとおりです。 1. ローカル・データ 2. アクティブな ResultSet 3. 他のすべての ResultSet 4. Environment 変数が見つからなかった場合、コンテンツ・サーバーのデバッグ出力にエラー・レポートは送信されません。
#local	<$#local.*variable*$>	ローカル・データから、指定された変数の値を取得します。変数が見つからなかった場合、コンテンツ・サーバーのデバッグ出力にエラー・レポートは送信されません。
#env	<$#env.*variable*$>	環境設定から、指定された変数の値を取得します。変数が見つからなかった場合、コンテンツ・サーバーのデバッグ出力にエラー・レポートは送信されません。

キーワード	例	説明
exec	<$exec *expression*$>	式を実行し、出力を抑制します（ページに式を表示しません）。 **注意：** Idoc Script の旧バージョンでは、*exec* キーワードは、出力ファイルでの変数値の表示を抑制するために必要でした。現行バージョンでは、*exec* キーワードは、出力ファイルでの式の表示を抑制するためにのみ必要です。
include	<$include *ResourceName*$>	指定されたリソースからコードを含めます。詳細は、3-4 ページの「インクルード」を参照してください。
super	<$include super.<*include*>$>	既存バージョンのインクルード・コードで開始します。詳細は、3-6 ページの「super タグ」を参照してください。

関連項目

– 変数の置換順序（3-9 ページ）
– キーワードと関数の対比（3-19 ページ）
– getValue（6-275 ページ）

キーワードと関数の対比

コンテンツ・サーバーのページでは、特殊キーワード（3-18 ページ）*include* および *exec*、関数（3-10 ページ）*inc* および *eval* を頻繁に使用します。この項では、これらのコマンドの違いを説明し、使用方法の例を示します。

キーワード *include* および *exec* は、定義されたパラメータで動作するスタンドアロン・コマンドですが、変数をパラメータとして取ることができません。関数 *inc* および *eval* は、同様の目的を果しますが、変数をパラメータとして取ることが可能であるため、変数の値に応じて Idoc スクリプト・コードを動的に作成できます。

次の項では、これらのキーワードおよび関数について詳しく説明します。

❖ exec キーワード（3-20 ページ）
❖ eval 関数（3-20 ページ）
❖ include キーワード（3-21 ページ）
❖ inc 関数（3-21 ページ）

exec キーワード

exec キーワードは、Idoc スクリプト式を実行し、出力を抑制します（ページに式を表示しません）。これは主に、ページに書込みをせずに変数を設定するために使用されます。

Idoc Script の旧バージョンでは、*exec* キーワードは、出力ファイルでの変数値の表示を抑制するために必要でした。現行バージョンでは、*exec* キーワードは、出力で式の表示を抑制するためにのみ必要です。

たとえば、次の最初の 1 行は、最後の 2 行と等価です。

```
<$varA="stringA", varB ="stringB"$>

<$exec varA="stringA"$>
<$exec varB="stringB"$>
```

詳細は、6-210 ページの「exec」を参照してください。

eval 関数

eval 関数は、実際の Idoc スクリプトのように式を評価します。

次の例では、*one* という変数に文字列 Company Name を割り当て、*two* という変数に、変数 *one* を含む文字列を割り当てます。

```
<$one="Company Name"$>
<$two="Welcome to <$one$>"$>
<$one$><br>
<$two$><br>
<$eval(two)$>
```

ページの出力で、変数 *one* は文字列 `Company Name`、変数 *two* は文字列 `Welcome to <$one$>`、および関数 *eval(two)* は文字列 `Welcome to Company Name` を表示します。

評価の対象となる文字列は、Idoc スクリプトのデリミタ <$ $> で囲む必要があります。そうでない場合は、Idoc スクリプトとして評価されません。

また、この方法で非常に多くのコンテンツが動的に生成されると、ページの表示に時間がかかる可能性があります。ページで *eval* 関数を頻繁に使用する場合は、コードを include に含め、inc 関数（3-21 ページ）を *eval* 関数と併用すると効果的な場合があります。

詳細は、6-205 ページの「eval」を参照してください。

include キーワード

include キーワードは、現行のページにコード・チャンクを組み込む標準の方法です。*include* はキーワードであるため、変数をパラメータとして取ることができません。パラメータは、既存 include の名前である必要があります。

詳細は、3-4 ページの「インクルード」および 6-347 ページの「include」を参照してください。

inc 関数

inc 関数は、変数をパラメータとして取ることが可能である以外は、*include* キーワードと同様の処理を行います。この関数は、現行の変数値に応じて使用する include を動的に変更する場合に非常に役立ちます。

たとえば、すべてのカスタム・メタデータ・フィールドではなく、一部のフィールドのみで Idoc スクリプトを実行する場合があります。その場合は、次の Idoc スクリプトを実行して、フィールド名に基づいてインクルードを動的に作成できます（*specific_include_xComments* など）。

```
<$loop DocMetaDefinition$>
    <$myInclude = "specific_include_" & dName$>
    <$exec inc(myInclude)$>
<$endloop$>
```

exec キーワード（3-20 ページ）の使用により、*inc* 関数によって指定されたインクルードの出力が抑制される点に注意してください。*inc* の前に *exec* を使用しないと、指定されたインクルード内の HTML がページに表示されます。

specific_include_xComments が存在しない場合でも、出力が表示されないため、このコードはエラーをスローしません。

詳細は、6-343 ページの「inc」を参照してください。

演算子

Idoc スクリプトは多数の演算子をサポートします。この項の内容は次のとおりです。

- 比較演算子（3-22 ページ）
- 特殊な文字列演算子（3-23 ページ）
- 数値演算子（3-24 ページ）
- ブール演算子（3-25 ページ）

比較演算子

次の比較演算子を使用して 2 つのオペランドの値を比較し、比較の結果に基づいて値 *true* または *false* を返します。これらの演算子は、Idoc スクリプトで整数およびブール値の比較に使用できます。

技術ヒント：HCSP または HCSF ページで Idoc スクリプトを使用する場合は、特殊な比較演算子を使用する必要があります。詳細は、『Dynamic Server Pages Guide』を参照してください。

演算子	説明	例
==	等しい	`<$if 2 == 3$>` の結果は false
!=	等しくない	`<$if 2 != 3$>` の結果は true
<	より小さい	`<$if 2 < 2$>` の結果は false
<=	以下	`<$if 2 <= 2$>` の結果は true
>	より大きい	`<$if 3 > 2$>` の結果は true
>=	以上	`<$if 3 >= 2$>` の結果は true

重要：これらは数値演算子であり、特殊なケースとして、日付などの数値的に有効な意味を持つ場合にのみ文字列に有効です（標準の比較演算子で使用する場合、日付はミリ秒に変換される）。

- 文字列連結、文字列論理和、および簡単な文字列比較には、特殊な文字列演算子（3-23 ページ）を使用します。
- 拡張文字列操作を実行するには、strEquals（6-766 ページ）、strReplace（6-775 ページ）またはその他の文字列関連のグローバル関数を使用します。

特殊な文字列演算子

文字列の連結および比較には、次の特殊な文字列演算子を使用します。

演算子	説明	例

&	**文字列結合演算子**は、文字列の連結を実行します。この演算子を使用してスクリプトを作成して、Idoc スクリプトのリソース・インクルードを行います。	`<$"<$include " & VariableInclude & "$>"$>`
		結果 :
		`<$include VariableName$>`

like	**文字列比較演算子**は、2 つの文字列を比較します。
	❖ 最初の文字列を 2 番目の文字列のパターンと比較します。(2 番目の文字列では、アスタリスクおよび疑問符文字をワイルドカード（3-24 ページ）として使用できます。)
	❖ この演算子では、大 / 小文字は区別されません。
		結果が FALSE の例 :
		`<$if "cart" like "car"$>`
		結果が TRUE の例 :
		`<$if "cart" like "car?"$>`
		結果が TRUE の例 :
		`<$if "carton" like "car*"$>`
		結果が TRUE の例 :
		`<$if "Carton" like "car*"$>`

|	**文字列論理和演算子**は、複数のオプションを区切り、論理和（OR）関数を実行します。	結果が TRUE の例 : `<$if "car" like "car\|truck\|van"$>`

たとえば、変数 a について、接頭辞が *car* かどうか、または部分文字列 *truck* が含まれるかどうかを確認するには、次の式を使用できます。

```
<$if a like "car*|*truck*"$>
```

重要 : 拡張文字列操作を実行するには、strEquals（6-766 ページ）、strReplace（6-775 ページ）またはその他の文字列関連のグローバル関数を使用します。完全なリストは、5-39 ページの「文字列」を参照してください。

ワイルドカード

like 演算子は、次のワイルドカード記号を認識します。

ワイルドカード	説明	例
*	0 個以上の文字に一致します。	• *grow** の場合、*grow*、*grows*、*growth* および *growing* は一致します。 • **car* の場合、*car*、*scar* および *motorcar* は一致します。 • *s*o* の場合、*so*、*solo* および *soprano* は一致します。
?	1 個の文字に一致します。	• *grow?* の場合、*grows* および *growl* は一致し、*growth* は一致しません。 • *grow??* の場合、*growth* は一致し、*grows* や *growing* は一致しません。 • *b?d* の場合、*bad*、*bed*、*bid* および *bud* は一致します。

数値演算子

次の数値演算子を使用して、算術演算を実行します。これらの演算子を使用すると、整数の結果は整数になり、浮動小数点数の結果は浮動小数点数になります。

演算子	説明	例
+	加算演算子。	`<$a=(b+2)$>`
-	減算演算子。	`<$a=(b-2)$>`
*	乗算演算子。	`<$a=(b*2)$>`
/	除算演算子。	`<$a=(b/2)$>`
%	係数演算子。2 つの値を除算した余りを提供します。	`<$a=(b%2)$>`

ブール演算子

ブール演算子を使用して、論理演算を行います。

演算子	説明	例
and	❖ 2つのオペランドの値がともに0以外または true の場合、結果は 1 です。 ❖ どちらかのオペランドが 0 に等しいまたは false の場合、結果は 0 です。	`<$if 3>2 and 4>3$>` 結果は 1
or	❖ 2 つのオペランドの値のどちらかが 0 以外または true の場合、結果は 1 です。 ❖ 2つのオペランドがともに0に等しいまたは false の場合、結果は 0 です。	`<$if 3>2 or 3>4$>` 結果は 1
not	❖ オペランドが 0 に等しいまたは false の場合、結果は 1 です。 ❖ オペランドの値が0以外またはtrueの場合、結果は 0 です。	`<$if not 3=4$>` 結果は 1

注意：ブール演算子は、左から右へと評価します。最初のオペランドの値によって演算の結果が確定した場合、2 番目のオペランドは評価されません。

メタデータ・フィールド

この項の内容は次のとおりです。

- メタデータ・フィールドのネーミング（3-26 ページ）
- 標準メタデータ・フィールド（3-27 ページ）
- オプション・リスト（3-33 ページ）
- 動的サーバー・ページでのメタデータの参照（3-36 ページ）

メタデータ・フィールドのネーミング

コンテンツ・サーバーの各メタデータ・フィールドは内部フィールド名を持ち、コード内で使用されます。また、多くのフィールドには、Web ページに表示されるキャプション（説明）があります。

- メタデータをユーザーに表示する場合は、フィールドのキャプションを使用します。
- ファイルのバッチ・ロードまたは動的サーバー・ページ（.hcst、.hcsp および .hcsf ページ）のスクリプトでは、内部フィールド名を使用します。
- すべての内部メタデータ・フィールド名は、d または x で始まります。
  - **事前定義済**のフィールド名は d で始まります。たとえば、*dDocAuthor* などがあります。
  - **カスタム**・フィールド名は x で始まります。たとえば、*xDepartment* などがあります。
- Configuration Manager でカスタム・メタデータ・フィールドを作成すると、フィールド名の先頭に x が自動的に追加されます。

重要：すべての場合において、内部メタデータ・フィールド名では大 / 小文字が区別されます。

標準メタデータ・フィールド

この項では、Content Server に保存される各コンテンツ・アイテムの標準メタデータ・フィールドについて説明します。フィールドは次のように分類されています。

- 共通メタデータ・フィールド（3-27 ページ）
- その他のフィールド（3-28 ページ）

共通メタデータ・フィールド

コンテンツ・サーバー・インタフェースのカスタマイズで最もよく使用されるメタデータ・フィールドを次に示します。これらのフィールドは、チェックインおよび検索ページにデフォルトで表示されます。

内部フィールド名	標準フィールド・キャプション	説明
dDocAccount	Account	セキュリティ・アカウント
dDocAuthor	Author	リビジョンをチェックインしたユーザー
xComments	Comments	説明的なコメント
dDocName	Content ID	コンテンツ・アイテムの一意の識別子
dOutDate	Expiration Date	リビジョンの検索や表示ができなくなる日付
dInDate	Release Date	リビジョンの検索および表示が可能になることが予定されている日付（3-28 ページの「dCreateDate」および 3-31 ページの「dReleaseDate」も参照）
dRevLabel	Revision	リビジョン・ラベル（3-32 ページの「dRevisionID」も参照）
dSecurityGroup	Security Group	セキュリティ・グループ
dDocTitle	Title	説明的なタイトル
dDocType	Type	コンテンツ・タイプ

注意：Content ID（*dDocName*）と dID（3-28 ページ）を混同しないでください。*dID* は、コンテンツ・アイテムの特定のリビジョンを参照する、内部で生成される整数です。

その他のフィールド

共通メタデータ・フィールド（3-27 ページ）に加えて、コンテンツ・アイテムでは次のメタデータも保存されます。

内部フィールド名	標準フィールド・キャプション	説明
dCheckoutUser	Checked Out By（「Content Information」ページ）	リビジョンをチェックアウトしたユーザー
dCreateDate	なし	リビジョンがチェックインされた日付
dDocFormats	Formats（「Content Information」ページ）	プライマリ・ファイルおよび代替ファイルのファイル形式
dDocID	なし	レンディションの一意の識別子
dExtension	なし	プライマリ・ファイルのファイル拡張子
dFileSize	なし	プライマリ・ファイルのファイル・サイズ（KB）
dFlag1	なし	未使用
dFormat	Format（チェックイン・ページ、「Allow override format on checkin」が有効）	プライマリ・ファイルのファイル形式
dID	なし	リビジョンの一意の識別子

内部フィールド名	標準フィールド・キャプション	説明
dIndexerState	なし	インデクサ・サイクルでのリビジョンの状態。次の値があります。 **X:** リビジョンは再構築サイクルによって処理されています。 **Y:** リビジョンは再構築サイクルによって処理されています。 **A、B、C または D:** 実行時に生成される値で、次の状態に割り当てられます。 • アクティブな更新サイクルでのリビジョンのロード • アクティブな更新サイクルでのリビジョンの索引付け • 再構築サイクルでのリビジョンのロード • 再構築サイクルでのリビジョンの索引付け これらの値の固有の定義は、<*install_dir*>/search/cyclesymbols.hda ファイルに保存されています。
dIsCheckedOut	なし	リビジョンがチェックアウトされたかどうかを示します。 **0:** チェックアウトされていない **1:** チェックアウト
dIsPrimary	なし	ファイル・タイプがプライマリか代替かを示します。 **0:** 代替ファイル **1:** プライマリ・ファイル
dIsWebFormat	なし	weblayout ディレクトリ内の Web で表示可能なファイルかどうかを示します。 **0:** Web で表示可能なファイルではない **1:** Web で表示可能なファイル
dLocation	なし	未使用。

内部フィールド名	標準フィールド・キャプション	説明
dMessage	なし（「Content Information」ページ）	索引付けまたは変換の成功または失敗原因を示します。
dOriginalName	Get Native File（「Content Information」ページ） Original File（リビジョンのチェックイン・ページ）	ネイティブ・ファイルの元のファイル名。
dProcessingState	なし	リビジョンの変換ステータス。次の値があります。 **I:** 不完全な変換。有効な Web 表示可能ファイルが生成された後に変換でエラーが発生し、ファイルはフルテキスト索引付けされていました。 **Y:** 変換済。リビジョンは正常に変換され、Web 表示可能ファイルが使用可能です。 **P:** リファイナリ・パススルー。Inbound Refinery がリビジョンの変換に失敗し、Web にネイティブ・ファイルを送信しました。 **F:** 失敗。リビジョンは削除されたか、ロックされているか、破損しています。または、索引付けでエラーが発生しました。 **C:** 処理中。Inbound Refinery がリビジョンを変換中です。 **M:** メタデータのみ。フルテキスト索引付けをバイパスしてリビジョンのメタデータのみが索引付けされました。

内部フィールド名	標準フィールド・キャプション	説明
dPublishState	なし	Content Publisher で使用されるリビジョンの公開状態。次の値があります。 **P:** 公開済 **S:** ステージング済 **W:** ワークフロー **null:** 未ステージング、未公開で、ワークフローにもない
dPublishType	なし	Content Publisher と使用するリビジョンのコンテンツ・タイプ。次の値があります。 **H:** ホーム **N:** ナビゲーション **S:** 問合せ結果ページ **P:** ページ **G:** ギャラリ・グラフィック **C:** コントリビュータ埋込みグラフィック **O:** その他
dReleaseDate	なし	リビジョンが実際にリリースされた日付。
dReleaseState	なし	リビジョンのリリース・ステータス。 **N:** 新規、索引付けされていない **E:** ワークフロー **R:** 処理中、索引付けの準備中 **I:** 現在、索引付け中で、ファイル名はリリース後の名前に変更済 **Y:** リリース済 **U:** リリースされたが、索引の更新が必要 **O:** 古いリビジョン
dRendition1	なし	ファイルがサムネイル・レンディションかどうかを示します。次の値があります。 **null:** サムネイル・レンディションではない **T:** サムネイル・レンディション

内部フィールド名	標準フィールド・キャプション	説明
dRendition2	なし	未使用。
dRevClassID	なし	「Content ID」（dDocName（3-27 ページ））に対応する内部整数。問合せのレスポンス時間の向上のために使用されます。
dRevisionID	なし	dRevLabel（3-27 ページ）の値とは無関係に、コンテンツ・アイテムのリビジョンごとに 1 ずつ増加する内部リビジョン番号。
dStatus	Status（「Content Information」ページ）	システムでのリビジョンの状態。次の値があります。 **GENWWW:** ファイルは Web 表示可能形式に変換中か、索引付け中です。または、変換か索引付けに失敗しました。 **DONE:** ファイルは、指定されたリリース日にリリースされるまで待機中です。 **RELEASED:** リビジョンはコンテンツ・サーバーで使用可能です。 **REVIEW:** リビジョンはワークフロー内にあり、レビュー中です。 **EDIT:** リビジョンはワークフローの最初の投稿ステップにあります。 **PENDING:** リビジョンは基本ワークフロー内にあり、ワークフロー内のすべてのリビジョンの承認を待機中です。 **EXPIRED:** リビジョンは、コンテンツ・サーバーで表示できなくなりました。（リビジョンは削除されていませんが、管理者以外はアクセスできません。） **DELETED:** リビジョンは削除され、次の索引付けサイクル中にコンテンツ・サーバーから完全に削除されるまで待機中です。
dWebExtension	なし	Web 表示可能ファイルのファイル拡張子。

オプション・リスト

オプション・リストは、メタデータ・フィールドで選択可能な値のセットです。次の項では、オプション・リストの使用について説明します。

- 内部オプション・リスト（3-33 ページ）
- オプション・リストのスクリプト（3-34 ページ）
- オプション・リストの作成（3-35 ページ）

内部オプション・リスト

Content Server では、次の内部オプション・リストがデフォルトで維持されます。

メタデータ・フィールド	オプション・リスト
Author（dDocAuthor）	docAuthors
Security Group（dSecurityGroup）	securityGroups*
Type（dDocType）	docTypes
Account（dDocAccount）	docAccounts*
Role（dRole）	roles

* *securityGroups* および *docAccounts* オプション・リストは、現行ユーザーの権限に応じてフィルタされます。

オプション・リストのスクリプト

オプション・リストの生成および有効化に使用される Idoc スクリプトの変数および関数は次のとおりです。

変数または関数	説明
optList（6-572 ページ）関数	メタデータ・フィールドのオプション・リストを生成します。
optionListName（6-565 ページ）変数	オプション・リストの名前を指定します。
fieldIsOptionList（6-224 ページ）変数	オプション・リストを持つメタデータ・フィールドを指定します。
fieldOptionListType（6-227 ページ）変数	オプション・リストのタイプを指定します（strict、combo、multi または access）。
hasOptionList（6-285 ページ）変数	*fieldIsOptionList* 変数の値に設定されます。この変数は条件文で使用されます。
defaultOptionListScript（6-142 ページ）変数	標準オプション・リスト・フィールドを表示する Idoc スクリプトのピースを定義します。
optionListScript（6-568 ページ）変数	*defaultOptionListScript* 変数で定義された標準オプション・リスト・フィールドの実装を上書きします。
optionsAllowPreselect（6-571 ページ）変数	メタデータ・フィールドのオプション・リストに、前回のオプション・リストの値を事前に含めることが可能かどうかを指定します。
addEmptyOption（6-10 ページ）変数	オプション・リストの最初の値が空白であることを指定します。
optionListResultSet（6-567 ページ）変数	オプション・リストの値が含まれる ResultSet を指定します。
optionListKey（6-564 ページ）変数	オプション・リストの値が含まれる ResultSet の列名を指定します。
optionListValueInclude（6-569 ページ）変数	オプション・リストの値を定義するインクルードを指定します。

オプション・リストの作成

オプション・リストを作成するには、次のいずれかの方法を使用します。

- optList（6-572 ページ）関数を使用して、基本オプション・リストを生成します。

  たとえば、次のコードは、予想される作成者のリストを HTML オプション・リストとして表示します。

```
<select name="dDocAuthors">
    <$optList docAuthors$>
</select>
```

- rsMakeFromList（6-677 ページ）関数を使用してオプション・リストから ResultSet を作成し、ResultSet に対するループ処理を行います。

  たとえば、次のコードは、*docAuthors* オプション・リストから *Authors* という ResultSet を作成し、ResultSet に対するループ処理を行って HTML オプション・リストを作成します。（*rsMakeFromList* のパラメータとして列名が指定されていないため、列名はデフォルトで row になります。）

```
<$rsMakeFromList("Authors","docAuthors")$>
<select name="dDocAuthors">
    <$loop Authors$>
        <option><$row$>
    <$endloop$>
</select>
```

技術ヒント：前述のコード・サンプルは同等です。通常、リスト・オプションを解析または評価する場合は、*rsMakeFromList* 関数を使用します。

動的サーバー・ページでのメタデータの参照

動的サーバー・ページでは、*ref:* という接頭辞を付けて格納されるいくつかのメタデータ値があります。（これにより、動的サーバー・ページによって ResultSet が損われないようにします。）

動的サーバー・ページで次のメタデータ値を参照する場合は、接頭辞 *ref:* を含める必要があります。

- hasDocInfo
- dDocName
- dExtension
- dSecurityGroup
- isLatestRevision
- dDocType
- dID

たとえば、次の文はドキュメント・タイプが Page かどうかを確認します。

```
<$if strEquals(ref:dDocType,"Page"))$>
```

注意：詳細は、『Dynamic Server Pages Guide』を参照してください。

VERITY INTEGRATION での日付 / 時間

注意：Verity Integration は、Content Server とともに使用できるアドイン製品であり、購入することができます。

Verity を使用して日付 / 時間を処理する場合は、いくつかの書式に関する考慮事項を理解しておく必要があります。ほとんどの場合、日付 / 時間の文字列で単一逆向き引用符（ASCII 文字コード 96 番）を使用する必要があります。間違って単一引用符（ASCII 文字コード 39 番）を使用すると、Verity による日付の読み違いが生じます。

次に例を示します。

```
dInDate > `10/1/03`
```

単一逆向き引用符が使用されている点に注意してください。この書式を使用すると、コンテンツ・サーバーが日付を解析し、Verity で正しく処理される書式で日付を再作成します。この例では、`10/1/2003` を 2003 年 10 月 1 日として解釈します。

または、次の書式を使用します。

```
dInDate > '10/1/03'
```

単一引用符が使用されている点に注意してください。この書式を使用すると、Verity では、UTC タイムゾーンと YMD（年の後に月と日が続く）を使用して日付を評価します。したがって、'10/1/03' は、2010 年 1 月 3 日として解釈されます。

技術ヒント：Verity の句を切り取り、貼り付けるには、WebLayout Editor Query Design を使用します。

MERGEINCLUDE を使用したレスポンスのフォーマット

MergeInclude を使用すると、テンプレート・ページ全体ではなく、IdocScript のインクルードに基づいて、コンテンツ・サーバー・リクエストからの結果をフォーマットできます。

MergeInclude は、多くの場合、IdcCommandX ActiveX モジュールを使用して ASP ページをコンテンツ・サーバーと統合するために使用される機能です。コンテンツ・サーバーのアーキテクチャは、Web インタフェースを最適化するように設計されていますが、基本的には、複数のインタフェースを持つモジュラー型のセキュアなサービス・ベースのアプリケーションです。GET_SEARCH_RESULTS などのサービスは、渡された QueryString およびユーザーのセキュリティ資格証明に基づいてレスポンス・データを生成します。このレスポンス・データは、内部で HDA ファイルの形式で表現されます。これを実行中に表示するには、コンテンツ・サーバーを使用して検索を実行し、URL に IsJava=1 を追加します。これにより、内部でレスポンス・データがどのように表現されるかを確認できます。

この HDA 形式は、Web ベースのユーザーには実用的でないため、IdocScript のインクルードおよびテンプレートを使用して、レスポンスを判読可能な HTML ページにフォーマットします。ユーザーはコンポーネントを使用して、テンプレートまたはいくつかのリソースのインクルードを変更して、HTML の表示を変更できます。

ただし、この検索結果のほんの一部のみを取得する必要がある場合（たとえば、コードの大部分が IdocScript でない ASP、JSP または PHP ページに表示する場合）や、IFRAME または DIV 要素をポップアップさせて結果を表示する場合、または結果の表示方法を動的に変更する場合は、次のパラメータを URL に追加できます。

```
MergeInclude=my_custom_include&IsJava=1
```

これにより、コンテンツ・サーバーは、サービスに指定されたテンプレートによるレスポンスのフォーマットをバイパスします。かわりに、コンテンツ・サーバーは、'my_custom_include' の IdocScript に基づいてレスポンスをフォーマットします。たとえば、検索を実行した後、前述の行を URL に追加し、コンポーネントに次のようなインクルードがあるとします。

```
<@dynamichtml my_custom_include@>
<html>
<table width=300>
<tr>
    <td><b>Name</b></td>
    <td><b>Title (Author)</b></td>
</tr>
<$loop SearchResults$>
<tr><td><a href="<$URL$>"><$dDocName$></a></td>
    <td><$dDocTitle$> (<$dDocAuthor$>)</td></tr>
<$endloop$>
```

```
</table>
</html>
<@end@>
```

これにより、不要な画像やフォーマットのない検索結果ページが表示されます。このように、任意の IdocScript インクルードを使用して、コンテンツ・サーバーのレスポンスをフォーマットできます。理論上、IdocScript には、XML、WML またはプレーン・テキストなど、任意の形式を含めることができます。

たとえば、検索結果を Excel スプレッドシートで読取り可能な形式にする場合は、カンマ区切りのエントリ・リストを返すようにリソース・インクルードを作成できます。これにより、返されたファイルをハード・ドライブに保存した後、Excel でオープンできます。また、もう 1 つの便利な方法として、リソース・インクルードを作成して、IdcCommandX ユーティリティまたは BatchLoader がファイルとして読み込むことが可能なレコード・セットにレスポンスをフォーマットすることもできます。このようなインクルードを検索結果や Weblayout Editor で作成された Active Report とともに使用することで、コンテンツ・サーバーのデータベースや検索索引に対する任意の問合せのための特別なバッチ・ファイルを作成することもできます。

- MergeInclude の変数は、通常のリソース・インクルードとは異なる方法でキャッシュされます。このため、リソース・インクルードに変更を加えた場合は、コンテンツ・サーバーを再起動する必要があります。ただし、docLoadResourceInclude() 関数を実行して MergeInclude 内から異なるインクルードを動的にロードする場合は、この再起動を回避できます。
- バージョン 6.0 以降では、MergeInclude により返されるデータに関して、データのコンテンツ・タイプはテキスト /html ではなく、テキスト / プレーンです。一部のクライアントでは（IE および多くのバージョンの Netscape など）、プレーン・テキストが html として表示されますが、そうでないクライアントもあります。問題がある場合は、リンクの際にコンテンツ・タイプを手動で設定することもできます。

有効範囲付のローカル変数

有効範囲付のローカル変数は、ページへのメタデータの表示方法を上書きするために使用される特殊なローカル変数です。これらの変数はコロンで区切られ、変数の有効範囲は特定のメタデータ・フィールドに制限されています。

たとえば、タイトルとコメント・フィールドを非表示にするには、次のフラグを設定します。

```
dDocTitle:isHidden=1
xComments:isHidden=1
```

これらのフラグは、URL で設定するか、インクルード std_doc_page_definitions を上書きして、ページの初めに設定する必要があります。

次のリストで、すべてのフラグはフィールド xFieldName の表示に影響を与えます。

- xFieldName:groupHeader: このフィールドがグループ内の最初のフィールドの場合、「Content Profiles」で設定します。グループ・ヘッダーに使用する HTML および IdocScript が含まれます。
- xFieldName:hasOptionList: デフォルトのオプション・リストのかわりに、フィールドに含まれるカスタム・オプション・リストを使用できます。xFieldName:optionListName 変数または xFieldName:optionListScript 変数とともに使用する必要があります。
- xFieldName:include: fieldInclude の値をカスタム・リソース・インクルードの名前に設定します。このリソースは、JavaScript および HTML を含め、ページ全体で使用されます。このフラグは、ほとんど使用されません。必要な場合は、std_namevalue_field インクルード・ファイルを、カスタム・インクルードの作成の指針として使用します。
- xFieldName:isExcluded: true に設定すると、ページからフィールドが完全に除外されます。フィールドとしても、非表示の入力フィールドとしても表示されません。フィールドは、ページの表示から完全に排除されます。
- xFieldName:isHidden: TRUE に設定すると、フィールドがページで非表示になります。フォームのポストを伴うページでは、フィールドはページに存在します。ただし、非表示の入力フィールドとしてのみ存在します。xFieldName または fieldValue が定義されていないかぎり、フィールドの値は空白になります。これにより、変更できないデフォルト値を持つページを作成できます。
- xFieldName:isInfoOnly: TRUE に設定すると、フィールドの値のみが表示されます。これを xFieldName:isHidden のかわりに使用して、発行されるデフォルト値をユーザーに表示できます。

- xFieldName:isRelocated: TRUE に設定すると、HTML ページにフィールドが自動的に表示されるのを止めることができます。デフォルトでは、ページのすべてのフィールドには特定の順序があります。順序を変更するには、このフラグを設定して、フィールドを手動で表示する必要があります。

```
<!-- hide the comments field -->
<$xComments:isRelocated = 1$>
<$loop DocMetaDefinition$>
<$strTrimWs(inc("std_meta_field_display"))$>
<$endloop$>
<!-- now turn off relocation, and display it -->
<$xComments:isRelocated = ""$>
<$fieldName="xComments", fieldCaption="Comments", fieldType="Memo"$>
<$include std_display_field$>
```

- xFieldName:isRequired: TRUE に設定すると、このフィールドは必須フィールドになります。このフラグは、JavaScript 検証コードがページに導入される前に、std_doc_page_definitions で設定する必要があります。

- xFieldName:maxLength: fieldWidth と同様に、テキスト入力フィールドの最大長を設定します。通常では fieldWidth よりも長く、また、データベース内のフィールドの幅より短くする必要があります。

- xFieldName:noSchema: TRUE に設定すると、フィールドのスキーマ・オプション・リストが無効化されます。カスタム・オプション・リストを動的な方法で生成する場合に必要です。

- xFieldName:optionListName: このフラグは、フィールドがオプション・リストである場合にのみ設定できます。値の表示に使用されるオプション・リストを上書きできます。

```
<$xCountry:hasOptionList = 1$>
<$xCountry:noSchema = 1$>
<$xCountry:optionListName = "securityGroups"$>
<$loop DocMetaDefinition$>
<$strTrimWs(inc("std_meta_field_display"))$>
<$endloop$>
```

- xFieldName:optionListScript: 明示的に定義されたオプション・リストのかわりに、IdocScript をレンダリングできる点を除いて、optionListName と同じです。ResultSet によるオプション・リストの導入が可能になります。

```
<$xCountry:hasOptionList = 1$>
<$xCountry:noSchema = 1$>
<$xCountry:optionListScript =
"<$rsMakeFromList('GROUPS', 'securityGroups')$>" &
"<select>\n" &
"<$loop GROUPS$>" &
" <option><$row$>" &
"<$endloop$>\n" &
"</select>"$>
<$loop DocMetaDefinition$>
<$strTrimWs(inc("std_meta_field_display"))$>
<$endloop$>
```

- `xFieldName:rowClass: std_nameentry_row` で使用されます。このフィールドを含む表の行に Cascading Style Sheet クラスを設定します。

```
<$xComments:rowClass="xuiPageTitleText"$>
<$loop DocMetaDefinition$>
<$strTrimWs(inc("std_meta_field_display"))$>
<$endloop$>
```

- `xFieldName:rowStyle: rowClass` と同じですが、これを使用するとインライン・スタイルを作成できます。たとえば、DHTML で「Comments」フィールドを非表示にするには、次のコードを使用します。

```
<$xComments:rowStyle="display:none"$>
<$loop DocMetaDefinition$>
<$strTrimWs(inc("std_meta_field_display"))$>
<$endloop$>
```

  ページをリロードせずに、動的にフィールドを非表示および表示する場合に役立ちます。

4

タイプ別 Idoc スクリプト

この章では、異なるタイプの Idoc スクリプト変数および関数について説明し、変数および関数をタイプ別に分類します。

- 条件付き動的変数（4-2 ページ）
- 構成変数（4-3 ページ）
- 動的変数（4-20 ページ）
- グローバル関数（4-21 ページ）
- ページ変数（4-25 ページ）
- 読取り専用変数（4-31 ページ）
- 設定可能変数（4-34 ページ）
- 値変数（4-34 ページ）
- Web サーバー変数（4-36 ページ）
- ワークフロー・スクリプト（4-37 ページ）

条件付き動的変数

一部の動的変数は条件変数であり、if、while、elseif または loop などの条件文でのみ使用できます。これらの変数には、次の特徴があります。

- 条件変数は、「設定可能」ではないが、「取得可能」な内部フラグです。
- 条件変数はブール・レスポンスのみを提供し、文字列や整数などの値を返しません。
- 条件変数は、*#active* キーワード接頭辞を受け入れません。このため、変数が見つからない場合、コンテンツ・サーバーのデバッグ出力にエラー・レポートを表示します。

次は、Idoc スクリプトの条件付き動的変数の完全なリストです。

AdminAtLeastOneGroup（6-16 ページ）
AfterLogin（6-21 ページ）
AllowCheckin（6-24 ページ）
AllowCheckout（6-25 ページ）
AllowReview（6-35 ページ）
dcShowExportLink（6-125 ページ）
EmptyAccountCheckinAllowed（6-191 ページ）
HasPredefinedAccounts（6-287 ページ）
HasUrl（6-289 ページ）
HeavyClient（6-290 ページ）
IsCheckinPreAuthed（6-380 ページ）
isComponentEnabled（6-381 ページ）
IsContributor（6-383 ページ）
IsCriteriaSubscription（6-384 ページ）
IsCurrentNav（6-385 ページ）
IsDynamic（6-388 ページ）
IsExternalUser（6-393 ページ）
IsFilePresent（6-401 ページ）
IsFullTextIndexed（6-405 ページ）
IsLocalSearchCollectionID（6-416 ページ）
IsLoggedIn（6-417 ページ）
IsMac（6-418 ページ）
IsMaxRows（6-419 ページ）
IsMultiPage（6-421 ページ）
IsNotSyncRev（6-424 ページ）
IsPromptingForLogin（6-429 ページ）
IsRequestError（6-434 ページ）
IsSubAdmin（6-440 ページ）
IsSun（6-441 ページ）
IsSysManager（6-442 ページ）

IsUploadSockets（6-448 ページ）
IsUserEmailPresent（6-449 ページ）
IsWindows（6-454 ページ）
IsWorkflow（6-455 ページ）
SingleGroup（6-748 ページ）
UserIsAdmin（6-862 ページ）
UserLanguageID（6-863 ページ）
UserLocaleId（6-864 ページ）
UseXmlUrl（6-891 ページ）

構成変数

一般に、構成変数は、構成ファイルでの設定に使用される事前定義済の Idoc スクリプト変数です。これらの変数を Idoc スクリプト内で使用して、構成設定が有効かどうかを検出したり、構成設定の値を返したりすることができます。

- ブール値を渡す構成変数は、TRUE（1）または FALSE（0）の値にのみ設定できます。
- 別途指定がないかぎり、ブール・タイプの構成値のデフォルトは FALSE（0）であり、文字列タイプの構成値のデフォルトは空の文字列です。

この項では、構成変数の最も一般的な保存先となる構成ファイル別に構成変数をリストします（通常は .cfg または .hda）。実際のアプリケーションでは、インストールの際の基準および選択に応じて異なります。

- Collaboration Manager の構成変数（4-4 ページ）
- Content Server の構成変数（4-5 ページ）
- Inbound Refinery の構成変数（4-16 ページ）
- Pop-Up Calendar の構成変数（4-19 ページ）
- Record Manager および Retention Manager の構成変数（4-20 ページ）

Collaboration Manager の構成変数

Component Manager を使用して追加された構成変数は、次のファイルに追加されます。

❖ <*install_dir*>/<*instancename*>/data/components/CollaborationManager/install.cfg

構成変数が編集可能な場合（Component Manager で指定）、次のファイルにも書き込まれます。

❖ <*install_dir*>/<*instancename*>/data/components/CollaborationManager/config.cfg

システムをロードする際に、構成変数は次の順序で読み込まれます。

1. <*install_dir*>/<*instancename*>/config/config.cfg
2. <*install_dir*>/<*instancename*>/data/components/CollaborationManager/install.cfg
3. <*install_dir*>/<*instancename*>/data/components/CollaborationManager/config.cfg

Colaboration Manager は、サーバーの複数のインスタンスにインストールできます。その場合、インスタンス名のパスは <*instancename*> で示されます。

次の Collaboration Manager の構成変数がドキュメントされています。

actionItemInterval（6-9 ページ）
AllowMixedCaseACL（6-30 ページ）
cmDiscussionDocType（6-78 ページ）
cmMemoFieldSize（6-79 ページ）
cmZonedSecurityFields（6-80 ページ）
deleteActionItemInterval（6-145 ページ）
deleteWatchdogInterval（6-146 ページ）
DisableMeetingCenter（6-158 ページ）
prjUseCustomRouting（6-597 ページ）
prjWizardTokens（6-598 ページ）
prjWizardWorkflowField（6-599 ページ）
SelectAccessMemberAllowManualEntry（6-723 ページ）
SelectAccessMemberResultLimit（6-724 ページ）
ShowWebDavUrlInBrowserRequest（6-747 ページ）
UserSelectAppletColumnOption（6-868 ページ）
UserSelectAppletDatabaseCaseSensitive（6-869 ページ）
UserSelectAppletDatabaseIso8859DictionaryOrder（6-870 ページ）
UserSelectAppletExternalAliasDefaultPermissions（6-871 ページ）
UserSelectAppletExternalUserDefaultPermissions（6-872 ページ）
UserSelectAppletForceEntriesToLowerCase（6-873 ページ）
UserSelectAppletInternalAliasDefaultPermissions（6-874 ページ）
UserSelectAppletInternalUserDefaultPermissions（6-875 ページ）
UseUserSelectApplet（6-882 ページ）
watchdogInterval（6-907 ページ）

Content Server の構成変数

次の構成ファイルは、コンテンツ・サーバーのインストール・ディレクトリ（<*install_dir*>）にあります。

<install_dir>/admin/bin/intradoc.cfg（4-5 ページ）
<install_dir>/admin/config/config.cfg（4-5 ページ）
<install_dir>/bin/intradoc.cfg（4-6 ページ）
<install_dir>/config/config.cfg（4-7 ページ）
<install_dir>/custom/folders/folders_environment.cfg（4-14 ページ）
<install_dir>/data/providers/provider_name/provider.hda（4-14 ページ）
<install_dir>/data/users/config/filter.hda（4-15 ページ）
<install_dir>/search/search.cfg（4-15 ページ）

<install_dir>/admin/bin/intradoc.cfg

次の構成変数は、<*install_dir*>/admin/bin/intradoc.cfg ファイルにあります。これらの変数は、Admin Server のディレクトリ・パスおよびグローバル設定を定義します。

IntradocDir（6-368 ページ）
JvmCommandLine（6-469 ページ）
WeblayoutDir（6-909 ページ）

<install_dir>/admin/config/config.cfg

次の構成変数は、<*install_dir*>/admin/config/config.cfg ファイルにあります。これらの変数は、Admin Server のコンテンツ・サーバー・インスタンスの設定を定義します。

CgiFileName（6-67 ページ）
HttpRelativeWebRoot（6-321 ページ）
HttpServerAddress（6-322 ページ）
IdcAdminServerHostName（6-327 ページ）
IdcAdminServerPort（6-329 ページ）

<install_dir>/bin/intradoc.cfg

次の構成変数は、<*install_dir*>/bin/intradoc.cfg ファイルにあります。これらの変数は、コンテンツ・サーバーのディレクトリ・パスおよびその他のグローバル設定を定義します。

AllowConcurrentUpdate（6-26 ページ）
ARCHIVE_COLLECTION（6-41 ページ）
BatchLoaderPath（6-53 ページ）
BatchLoaderUserName（6-54 ページ）
CLASSPATH（6-71 ページ）
CleanUp（6-72 ページ）
CollectionLocation（6-84 ページ）
DataDir（6-120 ページ）
EnableErrorFile（6-195 ページ）
FileEncoding（6-232 ページ）
HTMLEditorPath（6-296 ページ）
IdcAnalyzeLogDir（6-330 ページ）
IdcCommandFile（6-332 ページ）
IdcCommandLog（6-333 ページ）
IdcCommandUser（6-335 ページ）
IndexerPath（6-354 ページ）
IndexerStyleDir（6-358 ページ）
IntradocDir（6-368 ページ）
isPhysicallySplitDir（6-427 ページ）
JvmCommandLine（6-469 ページ）
LogMessagePrefix（6-508 ページ）
MaxErrorsAllowed（6-518 ページ）
PreviewOutputExtension（6-592 ページ）
PreviewPath（6-593 ページ）
SharedDir（6-739 ページ）
VaultDir（6-898 ページ）
WebBrowserPath（6-908 ページ）
WeblayoutDir（6-909 ページ）

<install_dir>/config/config.cfg

次の構成変数は、*<install_dir>*/config/config.cfg ファイルにあります。これらの変数は、コンテンツ・サーバーのほとんどの構成設定を定義します。

AdditionalIndexBuildParams（6-11 ページ）
AdditionalSubscriptionCollatedFieldList（6-14 ページ）
AdminDataDir（6-17 ページ）
AllowAllRedirectUrl（6-22 ページ）
AllowAlternateMetaFile（6-23 ページ）
AllowMatchesInDateCheck（6-29 ページ）
AllowPrimaryMetaFile（6-33 ページ）
AllowSamePrimaryAlternateExtensions（6-36 ページ）
AppletChunkSize（6-39 ページ）
AppletChunkThreshold（6-40 ページ）
ARCHIVE_COLLECTION（6-41 ページ）
ArchiverDoLocks（6-42 ページ）
ArchiverTouchMonitorInterval（6-43 ページ）
ArchiverTraceLockMaintenance（6-44 ページ）
ArchiverTraceLocks（6-45 ページ）
AuthorDelete（6-48 ページ）
AutoMaxMergeTrace（6-50 ページ）
AutoNumberPrefix（6-51 ページ）
AutoNumberWidth（6-52 ページ）
BinDir（6-55 ページ）
CachedResultRowCount（6-59 ページ）
CGI_DEBUG（6-64 ページ）
CGI_RECEIVE_DUMP（6-65 ページ）
CGI_SEND_DUMP（6-66 ページ）
CgiFileName（6-67 ページ）
CharMap（6-68 ページ）
CheckMkvdkDocCount（6-69 ページ）
ChunkedRequestTrace（6-70 ページ）
CollectionLocation（6-84 ページ）
ColumnMapFile（6-96 ページ）
ConfigDir（6-102 ページ）
ConversionCacheExpirationPeriodInDays（6-105 ページ）
CookieAuthTimeoutInDays（6-107 ページ）
CookieProxyPassword（6-108 ページ）
createAlternateMetaFile（6-110 ページ）
createPrimaryMetaFile（6-112 ページ）
DatabasePreserveCase（6-118 ページ）
DatabaseWildcards（6-119 ページ）

DatedCacheIntervalDays（6-122 ページ）
DateOutputFormat（6-123 ページ）
DCMaxFileSize（6-124 ページ）
DCTimeOut（6-126 ページ）
DCViewFormat（6-127 ページ）
Default Accounts（6-130 ページ）
DefaultAuth（6-131 ページ）
DefaultFilterInputFormat（6-133 ページ）
DefaultFilterOutputFormat（6-134 ページ）
DefaultHtmlConversion（6-136 ページ）
DefaultHtmlRefreshTimeoutInSeconds（6-137 ページ）
DefaultPasswordEncoding（6-143 ページ）
DirectoryLockingLogPath（6-148 ページ）
DisableAmberLayouts（6-149 ページ）
DisableClassicLayoutPublishing（6-150 ページ）
DisableCollegiateSkinPublishing（6-151 ページ）
DisableContentProfiles（6-152 ページ）
DisableDBLongIntegerField（6-153 ページ）
DisableForcedConversions（6-154 ページ）
DisableHttpUploadChunking（6-156 ページ）
DisableKeepingForcedConversions（6-157 ページ）
DisableOptimizedLatestReleaseQuerying（6-159 ページ）
DisableSharedCacheChecking（6-160 ページ）
DisableStellent05SkinPublishing（6-161 ページ）
DisableStellentSkinPublishing（6-162 ページ）
DisableTopMenusPublishing（6-163 ページ）
DisableTraysPublishing（6-164 ページ）
DisableWindowsPublishing（6-165 ページ）
DoAutoMaxMerge（6-170 ページ）
DoDocNameOrder（6-179 ページ）
DomainControllerName（6-180 ページ）
DownloadApplet（6-181 ページ）
DtmPort（6-187 ページ）
EnableDocumentHighlight（6-193 ページ）
EnableEnvironmentPackager（6-194 ページ）
EnableExpirationNotifier（6-196 ページ）
EnableOptimizedLatestReleaseQuerying（6-198 ページ）
EnableRebuildForcesReconversion（6-199 ページ）
EnableSchemaPublishing（6-200 ページ）
EnterpriseSearchAsDefault（6-203 ページ）
ExclusiveCheckout（6-208 ページ）
ExpireRevisionOnly（6-213 ページ）

FILTER_DEBUG（6-234 ページ）
ForcedConversionRules（6-236 ページ）
ForceDistinctRevLabels（6-237 ページ）
ForceDocTypeChoice（6-238 ページ）
ForceJdbcSync（6-241 ページ）
ForceSecurityGroupChoice（6-242 ページ）
FormInheritMetaDataFieldList（6-254 ページ）
GetCopyAccess（6-261 ページ）
HasExternalUsers（6-282 ページ）
HasGlobalUsers（6-283 ページ）
HelpDir（6-291 ページ）
HighlightBegin（6-293 ページ）
HtmlDataInputFilterLevel（6-295 ページ）
HighlightEnd（6-294 ページ）
HTMLEditorPath（6-296 ページ）
HttpRelativeWebRoot（6-321 ページ）
HttpServerAddress（6-322 ページ）
IdcAdminServerHostName（6-328 ページ）
IdcAuthExtraRequestParams（6-331 ページ）
IdcCommandServerHost（6-334 ページ）
IdcHttpHeaderVariables（6-336 ページ）
IdcLocale（6-338 ページ）
IDC_Name（6-326 ページ）
IdcServerBindAddress（6-340 ページ）
IdcServerSocketQueueDepth（6-341 ページ）
IndexableRenditions（6-351 ページ）
IndexerAutoWorkInterval（6-352 ページ）
IndexerLargeFileSize（6-353 ページ）
IndexerRebuildStagingManualCopy（6-355 ページ）
IndexerRebuildStagingPath（6-356 ページ）
IndexerTouchMonitorInterval（6-359 ページ）
IndexerTraceCycleListChanges（6-360 ページ）
IndexerTraceLockFileChanges（6-361 ページ）
IndexerTraceStartStop（6-362 ページ）
IndexVaultFile（6-363 ページ）
InstanceDescription（6-366 ページ）
InstanceMenuLabel（6-367 ページ）
IntradocRealm（6-369 ページ）
IntradocServerHostName（6-370 ページ）
IntradocServerPort（6-371 ページ）
IsAllowIndexRenditions（6-373 ページ）
IsAllowIndexVault（6-374 ページ）

IsAutoArchiver（6-375 ページ）
IsAutoNumber（6-376 ページ）
IsAutoQueue（6-377 ページ）
IsAutoSearch（6-378 ページ）
IsContentRefineryPresent（6-382 ページ）
IsDpSubmitErrorFatal（6-387 ページ）
IsWebServerPagesOnly（6-453 ページ）
IsDynamicConverterEnabled（6-389 ページ）
IsFormsPresent（6-402 ページ）
rule:IsFragmentOnly（6-404 ページ）
IsIntranetAuthOnly（6-409 ページ）
IsJdbc（6-411 ページ）
IsJdbcLockTrace（6-412 ページ）
IsJdbcQueryTrace（6-413 ページ）
IsJspServerEnabled（6-414 ページ）
IsOverrideFormat（6-426 ページ）
IsProxiedServer（6-430 ページ）
isRepromptLogin（6-433 ページ）
IsSavedWfCompanionFile（6-437 ページ）
isValidateFile（6-451 ページ）
isZoneField（6-458 ページ）
JdbcConnectionString（6-459 ページ）
JdbcDriver（6-460 ページ）
JdbcPassword（6-461 ページ）
JdbcPasswordEncoding（6-462 ページ）
JdbcUser（6-463 ページ）
JspAdminQuery（6-466 ページ）
JspDefaultIndexPage（6-467 ページ）
JspEnabledGroups（6-468 ページ）
LatestReleasedTimestampTimeout（6-471 ページ）
LocaleArchiverDateFormat（6-504 ページ）
LogDir（6-507 ページ）
LogName（6-509 ページ）
MacSupportsSignedApplets（6-511 ページ）
MailServer（6-512 ページ）
MajorRevSeq（6-513 ページ）
MaxArchiveErrorsAllowed（6-514 ページ）
MaxCollectionSize（6-515 ページ）
MaxConversionCacheSizeInMegs（6-516 ページ）
MaxDocIndexErrors（6-517 ページ）
MaxHitCount（6-519 ページ）
MaxIntegerAllowed（6-520 ページ）

MaxMergeBaseCount（6-522 ページ）
MaxQueryRows（6-525 ページ）
MaxRequestThreadCount（6-526 ページ）
MaxResults（6-527 ページ）
MaxSavedSearchResults（6-528 ページ）
MaxSearchConnections（6-529 ページ）
MaxSizeResourceFileCacheMillions（6-530 ページ）
MemoFieldSize（6-532 ページ）
MinIntegerAllowed（6-533 ページ）
MinMemoFieldSize（6-534 ページ）
MinorRevSeq（6-535 ページ）
MultiUpload（6-539 ページ）
NeverTimeoutForcedConversionsTemplateChange（6-541 ページ）
NoAutomation（6-542 ページ）
noInfoColumn（6-543 ページ）
noRevisionsColumn（6-546 ページ）
NoSearchConnectionCloseOnDocChange（6-547 ページ）
NotificationIntervalInDays（6-549 ページ）
NotificationMaximum（6-550 ページ）
NotificationQuery（6-551 ページ）
NotifyExtras（6-553 ページ）
NotifyTime（6-554 ページ）
NtlmSecurityEnabled（6-555 ページ）
NumAdditionalRenditions（6-556 ページ）
NumConnections（6-557 ページ）
OdbcDateFormat（6-558 ページ）
PLUGIN_DEBUG（6-584 ページ）
PrimaryWorkQueueTimeout（6-594 ページ）
ProfileCacheTimeoutInMins（6-617 ページ）
ProviderDir（6-621 ページ）
QueueDir（6-634 ページ）
RedirectHostsFilter（6-637 ページ）
RequestAuditAdditionalVerboseFieldsList（6-644 ページ）
RequestAuditIntervalSeconds1（6-645 ページ）
RequestAuditIntervalSeconds2（6-646 ページ）
RequestAuditListDepth1（6-647 ページ）
RequestAuditListDepth2（6-648 ページ）
RequestAuditSubClassificationField（6-649 ページ）
SchemaAllowRecursiveFileMove（6-696 ページ）
SchemaBuildDigestFile（6-697 ページ）
SchemaPublishCheckInterval（6-698 ページ）
SchemaPublishInterval（6-699 ページ）

SchemaPublishMaximumInterval（6-700 ページ）
SchemaPublishMinimumInterval（6-701 ページ）
SchemaPublishRenameRetryInitialInterval（6-702 ページ）
SchemaPublishRetryInterval（6-704 ページ）
SchemaPublishRetryIntervalMax（6-705 ページ）
SchemaPublishRenameRetryIterations（6-703 ページ）
SchemaVerifyPublishing（6-706 ページ）
SchemaWin32RenameDelayInterval（6-707 ページ）
SearchCacheCleanerInterval（6-711 ページ）
SearchCacheHardLimit（6-712 ページ）
SearchCacheMaxItemAge（6-713 ページ）
SearchCacheSoftLimit（6-714 ページ）
SearchCacheTrace（6-715 ページ）
SearchConnectionWaitTimeout（6-716 ページ）
SearchDebugLevel（6-717 ページ）
SearchDir（6-718 ページ）
SearchIndexerEngineName（6-719 ページ）
SearchQueryFormat（6-720 ページ）
SearchQueryTrace（6-721 ページ）
SelfRegisteredAccounts（6-725 ページ）
SelfRegisteredRoles（6-726 ページ）
SharedWeblayoutDir（6-740 ページ）
ShowCreateExternalMenu（6-741 ページ）
showDocActions（6-742 ページ）
ShowOnlyKnownAccounts（6-744 ページ）
ShowRevisionsColumn（6-745 ページ）
showSearchResultsCheckoutColumn（6-746 ページ）
SkipMozillaMailToWorkaround（6-749 ページ）
SmtpPort（6-750 ページ）
SocketHostAddressSecurityFilter（6-751 ページ）
SocketHostNameSecurityFilter（6-752 ページ）
SpecialAuthGroups（6-755 ページ）
StdPageWidth（6-758 ページ）
StrConfineOverflowChars（6-765 ページ）
SubjectsDir（6-784 ページ）
SubscriptionMethod（6-786 ページ）
SubscriptionCollatedFieldList（6-785 ページ）
SysAdminAddress（6-787 ページ）
SystemDateFormat（6-788 ページ）
SystemLocale（6-789 ページ）
SystemReportsDir（6-791 ページ）
SystemResourcesDir（6-792 ページ）

SystemTemplatesDir（6-793 ページ）
SystemTimeZone（6-794 ページ）
TempDir（6-795 ページ）
ThumbnailHeight（6-804 ページ）
ThumbnailWidth（6-809 ページ）
TimeoutChangedTemplateForcedConversionsInMins（6-810 ページ）
TimeoutMismatchedForcedConversionsInMins（6-811 ページ）
TimeoutPerOneMegInSec（6-812 ページ）
TraceIsVerbose（6-817 ページ）
TraceConversionCacheCleanup（6-816 ページ）
TraceResourceConflict（6-818 ページ）
TraceResourceLoad（6-819 ページ）
TraceResourceOverride（6-820 ページ）
TraceSectionsList（6-821 ページ）
UniversalSearchAllowFullText（6-825 ページ）
UploadApplet（6-826 ページ）
UseAccounts（6-829 ページ）
UseAdsi（6-831 ページ）
UseBellevueLook（6-835 ページ）
UseCascadingStyleSheets（6-836 ページ）
UseCustomModaling（6-837 ページ）
UseDCLApplet（6-839 ページ）
UseFourDigitYear（6-840 ページ）
UseMicrosoftVM（6-846 ページ）
UseNtlm（6-848 ページ）
UserCache（6-852 ページ）
UserCacheTimeout（6-853 ページ）
UseRedirectedOutput（6-855 ページ）
UseRevisionCreateDate（6-856 ページ）
UserFilteredSubjects（6-857 ページ）
UserPublishCacheDir（6-866 ページ）
UseSearchCache（6-876 ページ）
UseSelfRegistration（6-879 ページ）
UseSSL（6-881 ページ）
UseVdk4CustomStyle（6-883 ページ）
UseXpedioLook（6-892 ページ）
VerityEncoding（6-903 ページ）
VerityLocale（6-905 ページ）
WarnAboutTimeZone（6-906 ページ）
WebBrowserPath（6-908 ページ）
WebProxyAdminServer（6-910 ページ）
WebServerAuthOnly（6-911 ページ）

WorkflowDir（6-943 ページ）
WorkflowIntervalHours（6-944 ページ）
xml（6-945 ページ）
XMLEncodingMode（6-946 ページ）

<install_dir>/custom/folders/*folders_environment.cfg*

CollectionDeleteEnabled（6-81 ページ）
CollectionHiddenMeta（6-82 ページ）
CollectionInhibitUpdateMeta（6-83 ページ）
CollectionMeta（6-85 ページ）
CollectionMoveEnabled（6-86 ページ）
CollectionReadOnlyMarkedFolders（6-87 ページ）
CollectionReadOnlyMeta（6-88 ページ）
CollectionReleasedOnly（6-89 ページ）
CollectionSecurityReadOnly（6-90 ページ）
CollectionTrashDeleteDate（6-91 ページ）
CollectionTrashDeleteLocation（6-92 ページ）
CollectionTrashDeleteOldName（6-93 ページ）
CollectionTrashDeleter（6-94 ページ）
CollectionWebDAVServer（6-95 ページ）
InitialColID（6-365 ページ）

<install_dir>/data/providers/*provider_name*/provider.hda

次の構成変数は、*<install_dir>*/data/providers/*provider_name*/provider.hda ファイルにあります。これらの変数は、コンテンツ・サーバー・プロバイダの設定を定義します。

AcctPermDelim（6-5 ページ）
AcctPrefix（6-7 ページ）
AttributeMap（6-46 ページ）
DefaultNetworkAccounts（6-140 ページ）
IsPrimaryUserBase（6-428 ページ）
LdapAdminDN（6-475 ページ）
LdapAdminPassword（6-476 ページ）
LdapPort（6-477 ページ）
LdapServer（6-478 ページ）
LdapSuffix（6-479 ページ）
NumConnections（6-557 ページ）
Priority（6-596 ページ）
ProviderClass（6-618 ページ）
ProviderConfig（6-619 ページ）
ProviderConnection（6-620 ページ）
ProviderType（6-622 ページ）

RolePrefix（6-657 ページ）
SourcePath（6-754 ページ）
UseGroupFilter（6-842 ページ）
UseNetscape（6-847 ページ）
UseSecureLdap（6-878 ページ）

<install_dir>/data/users/config/filter.hda

次の構成変数は、*<install_dir>*/data/users/config/filter.hda ファイルにあります。これらの変数は、Web サーバーのフィルタ設定を定義します。

AccountMapPrefix（6-4 ページ）
AcctPermDelim（6-5 ページ）
AcctPrefix（6-7 ページ）
AdsSimpleAuth（6-18 ページ）
AdsUserName（6-19 ページ）
AdsUserPassword（6-20 ページ）
AttributeMap（6-46 ページ）
CGI_DEBUG（6-64 ページ）
CGI_RECEIVE_DUMP（6-65 ページ）
CGI_SEND_DUMP（6-66 ページ）
DefaultAuth（6-131 ページ）
DefaultMasterDomain（6-138 ページ）
DefaultNetworkAccounts（6-140 ページ）
FILTER_DEBUG（6-234 ページ）
LocalGroupServer（6-505 ページ）
NetworkAdminGroup（6-540 ページ）
PLUGIN_DEBUG（6-584 ページ）
ProfileCacheTimeoutInMins（6-617 ページ）
RolePrefix（6-657 ページ）
UseFullGroupName（6-841 ページ）
UseGroupFilter（6-842 ページ）
UseLocalGroups（6-845 ページ）
UseShortNamesAlways（6-880 ページ）

<install_dir>/search/search.cfg

次の構成変数は、*<install_dir>*/search/search.cfg ファイルにあります。これらの変数は、Verity Integration の設定を定義します。Verity Integration は、Content Server とともに使用できるアドオン製品であり、購入することができます。

VerityAppName（6-901 ページ）
VerityAppSignature（6-902 ページ）
VerityInstallDir（6-904 ページ）

<stub_dir>/bin/intradoc.cfg

ClusterBinDirRule（6-75 ページ）
ClusterNodeAddress（6-76 ページ）
ClusterNodeName（6-77 ページ）

Inbound Refinery の構成変数

次の構成ファイルは、Inbound Refinery のインストール・ディレクトリ（*<install_dir>*/IdcRefinery/）にあります。

- <install_dir>/IdcRefinery/connections/main/intradoc.cfg（4-16 ページ）
- <install_dir>/IdcRefinery/shared/idcrefinery.cfg（4-18 ページ）

重要：Inbound Refinery に関連するディレクトリまたはファイル・パスの定義文字列を返す構成変数は、Inbound Refinery インストール・ディレクトリに対する相対パスを返します。HTM ページ内の情報として渡される場合、通常、これらのパスは、コンテンツ・サーバーに認識されず、ユーザーにとって使用可能ではありません。

<install_dir>/IdcRefinery/connections/*main*/intradoc.cfg

次の構成変数は、*<install_dir>*/IdcRefinery/connections/*main*/intradoc.cfg ファイルにあります。このファイルの設定は、1 つの Inbound Refinery インスタンスのみに影響します（たとえば、*main* ディレクトリの intradoc.cfg ファイルは、*main* 接続インスタンスにのみ影響します）。

AllowMrSidRetryCompressionLevel（6-31 ページ）
AllowSecondaryConversion（6-37 ページ）
AllowSkippedHyperlinktoCauseIncomplete（6-38 ページ）
CompressionMinHeight（6-97 ページ）
CompressionMinWidth（6-98 ページ）
ConnectionName（6-103 ページ）
ConvertXmlAsWebviewable（6-106 ページ）
CreatePDFThumbnails（6-111 ページ）
CreateXMLRendition（6-113 ページ）
CustomConversionWaitTime（6-116 ページ）
CustomConverterPath（6-117 ページ）
DocConverterEngineDir（6-171 ページ）
EnableCopyConvertCopy（6-192 ページ）
ExcludeConversionFilter（6-207 ページ）
ExclusiveIncludeConversionFilter（6-209 ページ）
FileEncoding（6-232 ページ）
IncludeXmlFlexionDocType（6-349 ページ）
IntradocDir（6-368 ページ）

JvmCommandLine（6-469 ページ）
MaxNumberConvertedHistoryShow（6-523 ページ）
MrSidLowestCompressionLevel（6-536 ページ）
OpenOfficeListenerPort（6-560 ページ）
OpenOfficeListenerURL（6-561 ページ）
OpenOfficePath（6-562 ページ）
OptimizePDF（6-563 ページ）
PdfVersion（6-583 ページ）
PostprocessPDFPath（6-587 ページ）
PowerPointPrintOutput（6-589 ページ）
PreconversionPath（6-590 ページ）
PreconversionWaitTime（6-591 ページ）
ProcessExcelUrlHyperlinks（6-600 ページ）
ProcessInDesignBookMarks（6-601 ページ）
ProcessInDesignHyperlinks（6-602 ページ）
ProcessInDesignInternalPDFThumbnail（6-603 ページ）
ProcessInDesignLinkUpdate（6-604 ページ）
ProcessInDesignPDFCompatibility50（6-605 ページ）
ProcessPowerpointHyperlinks（6-606 ページ）
ProcessVisioInternalHyperlinks（6-607 ページ）
ProcessVisioUrlHyperlinks（6-608 ページ）
ProcessWordBookMarks（6-609 ページ）
ProcessWordComments（6-610 ページ）
ProcessWordEmailLinks（6-611 ページ）
ProcessWordFootEndNote（6-612 ページ）
ProcessWordHeadings（6-613 ページ）
ProcessWordHyperlinks（6-614 ページ）
ProcessWordToc（6-615 ページ）
ProcessWordUrlHyperlinks（6-616 ページ）
SecondaryConversionStepName（6-722 ページ）
SetPdfVersion（6-736 ページ）
SharedDir（6-739 ページ）
ThumbnailDPI（6-803 ページ）
ThumbnailJPEGQuality（6-805 ページ）
ThumbnailSizeLimit（6-806 ページ）
ThumbnailSizingMethod（6-807 ページ）
UseLightweightToPostscriptStep（6-844 ページ）
UseSearchML（6-877 ページ）
UseXmlEmbeddings（6-884 ページ）
UseXmlExtensibleDTD（6-885 ページ）
UseXmlNoDTD（6-886 ページ）
UseXmlNoReference（6-887 ページ）

UseXmlPStyleNames（6-888 ページ）
UseXmlSchema（6-889 ページ）
UseXmlSchemaDTD（6-890 ページ）
VaultDir（6-898 ページ）
VerboseMode（6-900 ページ）
WeblayoutDir（6-909 ページ）
XMLTimeout（6-947 ページ）

<install_dir>/IdcRefinery/shared/idcrefinery.cfg

次の構成変数は、*<install_dir>*/IdcRefinery/shared/idcrefinery.cfg ファイルにあります。このファイル内の設定は、特定のマシン上のすべての Inbound Refinery インスタンスに影響します。

AcadUseLISPInterface（6-3 ページ）
AdjustPrinterMargins（6-15 ページ）
AllowPassthru（6-32 ページ）
AutoCad2000PlotterFilePath（6-49 ページ）
ComputerName（6-101 ページ）
DebugMode（6-128 ページ）
DebugStdConversion（6-129 ページ）
DefaultGraphicTimeout（6-135 ページ）
DefaultNativeTimeout（6-139 ページ）
DefaultPostscriptTimeout（6-144 ページ）
DisplayPSOrigin（6-166 ページ）
DistillerNormJobSetting（6-167 ページ）
DistillerOptJobSetting（6-168 ページ）
DistillerPath（6-169 ページ）
FrameMakerCyclesAllowed（6-255 ページ）
FrameMakerexePath（6-256 ページ）
FrameMakerTimeout（6-257 ページ）
HelpDir（6-291 ページ）
ImageAlchemyExePath（6-342 ページ）
IsNTCompressionPresent（6-425 ページ）
IsThumbnailPresent（6-443 ページ）
LogWithConnectionName（6-510 ページ）
MaxNumRecursiveStepDefinitions（6-524 ページ）
MSPubexePath（6-538 ページ）
PageMakerExePath（6-574 ページ）
PostConversionTimeOut（6-586 ページ）
PostscriptPrinterName（6-588 ページ）
PrinterPortPath（6-595 ページ）
ShowHyperlinkBox（6-743 ページ）

ThumbnailTemplate（6-808 ページ）
UseAdobeDistiller（6-830 ページ）
UseAlchemy111（6-832 ページ）
UseAutoCad2000（6-833 ページ）
UseAutocadModelSpace（6-834 ページ）

Link Manager の構成変数

次の変数は、Link Manager アプリケーションの構成に使用されます。これらの変数は、<*install_dir*>/config/config.cfg ファイルに設定されます。

AllowForceDelete（6-27 ページ）
HasSiteStudio（6-288 ページ）
LkDisableOnRebuild（6-480 ページ）
LkDisableReferenceInfo（6-481 ページ）
LkDisallowConversionFormats（6-482 ページ）
LkIsSecureSearch（6-484 ページ）
LkReExtractOnRebuild（6-485 ページ）
LkRefreshBatchSize（6-486 ページ）
LkRefreshErrorsAllowed（6-489 ページ）
LkRefreshErrorPercent（6-488 ページ）
LkRefreshErrorThreshold（6-490 ページ）

Pop-Up Calendar の構成変数

次の変数は、Pop-Up Calendar アプリケーションの構成に使用されます。これらの変数は、<*install_dir*>/config/config.cfg ファイルに設定されます。

pucDefaultTime（6-625 ページ）
pucEnableOnSearch（6-626 ページ）
pucUseMonthLocale（6-628 ページ）
pucUseWeekdayLocale（6-629 ページ）
pucXOffset（6-630 ページ）
pucYOffset（6-631 ページ）
pucZIndex（6-632 ページ）

Record Manager および Retention Manager の構成変数

AllowRetentionPeriodWithoutCutoff（6-34 ページ）
HideVitalReview（6-292 ページ）
RecordsManagementNumberOverwriteOnDelete（6-636 ページ）
RecordsManagementDenyAuthorFreePassOnRMSecurity（6-635 ページ）
RmaAddDocWhereClauseForScreening（6-652 ページ）
RMAHideExternalFieldsFromCheckInUpdate（6-653 ページ）
RMAHideExternalFieldsFromSearchInfo（6-654 ページ）
RmaNotifyDispReviewerAndCatAuthor（6-655 ページ）
RmaNotifyReviewerAndAlternateReviewer（6-656 ページ）

動的変数

動的変数は、変数が出現するたびに評価されます。したがって、変数が検出されるたびに、コードから値を再計算します。（これに対して、値変数は、サービス・コールの最初に一度評価され、サービス・コール全体にわたってその値が使用されます。4-34 ページの「値変数」を参照してください。）通常、動的変数は、文字列または整数などの値を返します。

次は、Idoc スクリプトの動的変数の完全なリストです。

AllowIntranetUsers（6-28 ページ）
computeDocUrl（6-99 ページ）
getFieldConfigValue（6-264 ページ）
getHelpPage（6-268 ページ）
getOptionListSize（6-269 ページ）
HttpIgnoreWebServerInternalPortNumber（6-316 ページ）
parseClbraProject（6-576 ページ）

関連項目

– 値変数（4-34 ページ）

グローバル関数

Idoc スクリプトには、多数の組込みグローバル関数があります。関数は、文字列比較ルーチン、文字列操作ルーチン、日付フォーマット、ResultSet 操作などのアクションを実行します。一部の関数は、結果も返します（計算や比較の結果など）。

関数に情報を渡すには、関数名の後に情報をカッコで囲んで指定します。関数に渡す情報のことをパラメータと呼びます。パラメータを取らない関数、1 つのパラメータを取る関数、および複数のパラメータを取る関数があります。また、関数の使用方法によってパラメータの数が異なる関数もあります。

次は、Idoc スクリプトのグローバル関数のリストです。

abortToErrorPage（6-2 ページ）
break（6-56 ページ）
cacheInclude（6-60 ページ）
clearSchemaData（6-73 ページ）
computeRenditionUrl（6-100 ページ）
dateCurrent（6-121 ページ）
docLoadResourceIncludes（6-172 ページ）
docRootFilename（6-175 ページ）
docUrlAllowDisclosure（6-178 ページ）
dpGet（6-183 ページ）
dpPromote（6-184 ページ）
dpPromoteRs（6-185 ページ）
dpSet（6-186 ページ）
encodeHtml（6-201 ページ）
eval（6-205 ページ）
executeService（6-211 ページ）
forceExpire（6-239 ページ）
formatDate（6-243 ページ）
formatDateDatabase（6-245 ページ）
formatDateDisplay（6-246 ページ）
formatDateOnly（6-247 ページ）
formatDateOnlyDisplay（6-249 ページ）
formatDateOnlyFull（6-250 ページ）
formatDateWithPattern（6-251 ページ）
formatTimeOnly（6-252 ページ）
formatTimeOnlyDisplay（6-253 ページ）
getDebugTrace（6-262 ページ）
getErrorTrace（6-263 ページ）
getFieldViewDisplayValue（6-265 ページ）
getFieldViewValue（6-266 ページ）
getFreeMemory（6-267 ページ）

getParentValue（6-270 ページ）
getRequiredMsg（6-271 ページ）
getTextFile（6-272 ページ）
getTotalMemory（6-273 ページ）
getUserValue（6-274 ページ）
getValue（6-275 ページ）
getValueForSpecifiedUser（6-278 ページ）
getViewValue（6-279 ページ）
getViewValueResultSet（6-280 ページ）
hasAppRights（6-281 ページ）
inc（6-343 ページ）
incDynamicConversionByRule（6-345 ページ）
incGlobal（6-346 ページ）
incTemplate（6-350 ページ）
indexerSetCollectionValue（6-357 ページ）
isActiveTrace（6-372 ページ）
isFalse（6-396 ページ）
isLayoutEnabled（6-415 ページ）
isTrue（6-444 ページ）
isUserOverrideSet（6-450 ページ）
js（6-464 ページ）
jsFilename（6-465 ページ）
lc（6-472 ページ）
lcCaption（6-474 ページ）
LMDefaultLayout（6-491 ページ）
LMDefaultSkin（6-492 ページ）
lmGetLayout（6-493 ページ）
lmGetSkin（6-494 ページ）
loadCollectionInfo（6-495 ページ）
loadDocMetaDefinition（6-496 ページ）
loadDocumentProfile（6-497 ページ）
loadEnterpriseSearchCollections（6-499 ページ）
loadEnterpriseSearchResults（6-500 ページ）
loadSchemaData（6-501 ページ）
loadSearchOperatorTables（6-502 ページ）
loadUserMetaDefinition（6-503 ページ）
optList（6-572 ページ）
parseDataEntryDate（6-577 ページ）
parseDate（6-578 ページ）
parseDateWithPattern（6-580 ページ）
pneNavigation（6-585 ページ）
proxiedBrowserFullCgiWebUrl（6-623 ページ）

proxiedCgiWebUrl（6-624 ページ）
regexMatches（6-638 ページ）
regexReplaceAll（6-639 ページ）
regexReplaceFirst（6-640 ページ）
rptDisplayMapValue（6-659 ページ）
rsAddFields（6-660 ページ）
rsAddRowCountColumn（6-661 ページ）
rsAppend（6-662 ページ）
rsAppendNewRow（6-663 ページ）
rsAppendRowvalues（6-664 ページ）
rsCopyFiltered（6-666 ページ）
rsCreateResultSet（6-667 ページ）
rsDeleteRow（6-668 ページ）
rsDocInfoRowAllowDisclosure（6-669 ページ）
rsExists（6-670 ページ）
rsFieldByIndex（6-671 ページ）
rsFieldExists（6-672 ページ）
rsFindRowPrimary（6-673 ページ）
rsFirst（6-674 ページ）
rsInsertNewRow（6-675 ページ）
rsIsRowPresent（6-676 ページ）
rsMakeFromList（6-677 ページ）
rsMakeFromString（6-680 ページ）
rsMerge（6-682 ページ）
rsMergeDelete（6-683 ページ）
rsMergeReplaceOnly（6-684 ページ）
rsNext（6-685 ページ）
rsNumFields（6-686 ページ）
rsNumRows（6-687 ページ）
rsRemove（6-688 ページ）
rsRename（6-689 ページ）
rsRenameField（6-690 ページ）
rsSetRow（6-691 ページ）
rsSort（6-692 ページ）
rsSortTree（6-693 ページ）
setContentType（6-731 ページ）
setExpires（6-733 ページ）
setHttpHeader（6-734 ページ）
setMaxAge（6-735 ページ）
setValue（6-738 ページ）
setResourceInclude（6-737 ページ）
stdSecurityCheck（6-759 ページ）

strCenterPad（6-760 ページ）
strCommaAppendNoDuplicates（6-762 ページ）
strConfine（6-763 ページ）
strEquals（6-766 ページ）
strEqualsIgnoreCase（6-767 ページ）
strGenerateRandom（6-768 ページ）
strIndexOf（6-769 ページ）
strLeftFill（6-770 ページ）
strLeftPad（6-771 ページ）
strLength（6-772 ページ）
strLower（6-773 ページ）
strRemoveWs（6-774 ページ）
strReplace（6-775 ページ）
strReplaceIgnoreCase（6-776 ページ）
strRightFill（6-777 ページ）
strRightPad（6-778 ページ）
strSubstring（6-779 ページ）
strTrimWs（6-781 ページ）
strUpper（6-782 ページ）
toInteger（6-813 ページ）
trace（6-814 ページ）
url（6-827 ページ）
urlEscape7Bit（6-828 ページ）
userHasAccessToAccount（6-859 ページ）
userHasGroupPrivilege（6-860 ページ）
userHasRole（6-861 ページ）
utGetValue（6-893 ページ）
utLoad（6-894 ページ）
utLoadDocumentProfiles（6-895 ページ）
utLoadResultSet（6-896 ページ）
xml（6-945 ページ）

ページ変数

ページ変数は、特定のコンテンツ・サーバー Web ページに設定され、特定のページ属性または機能を有効にします。ページ変数は、変数が設定されたページにのみ適用されます。

この項の内容は次のとおりです。

❖ ページ表示変数（4-25 ページ）

❖ フィールド表示変数（4-26 ページ）

ページ表示変数

通常、ページの表示に影響を与えるページ変数は、ページの先頭近くに設定されます。

警告：ページ表示変数は、読取り専用変数として使用します。通常、これらの変数値を設定または変更すると、ページでのメタデータの表示が変更され、ほとんどの場合は好ましくない結果が生じます。

次は、Idoc スクリプトのページ表示変数の完全なリストです。

generateUniqueId（6-259 ページ）
isCheckin（6-379 ページ）
isDocPage（6-386 ページ）
isEditMode（6-390 ページ）
isFormSubmit（6-403 ページ）
isInfo（6-407 ページ）
isQuery（6-431 ページ）
isUpdate（6-446 ページ）
isUploadFieldScript（6-447 ページ）
localPageType（6-506 ページ）
noMCPrefill（6-545 ページ）

フィールド表示変数

フィールド表示変数は、次のタイプに分類できます。

- フィールド情報変数（4-26 ページ）
- 共通フィールド表示変数（4-27 ページ）
- その他のフィールド表示変数（4-29 ページ）

フィールド情報変数

次の変数は、メタデータ・フィールドの情報を定義します。メタデータ・フィールドごとに変数値がロードまたは計算されます。

fieldCaption（6-216 ページ）
fieldDefault（6-219 ページ）
fieldIsOptionList（6-224 ページ）
fieldName（6-226 ページ）
fieldOptionListType（6-227 ページ）
fieldType（6-228 ページ）
fieldValue（6-229 ページ）
fieldWidth（6-231 ページ）
isRequired（6-435 ページ）
requiredMsg（6-650 ページ）
valueStyle（6-897 ページ）

例

<install_dir>/shared/config/resources/std_page.htm リソース・ファイルの *std_prepare_metafield_include* インクルードは、現在のメタデータ・フィールドの表示に備えて、ローカル・データから多数のフィールド情報変数をロードします。

```
<@dynamichtml std_prepare_metafield_include@>
<!--Prepare for presenting field-->
<$fieldName=dName, fieldCaption=dCaption, fieldDefault=dDefaultValue$>
<$fieldType=dType, fieldIsOptionList=dIsOptionList,
fieldOptionListType=dOptionListType$>
<@end@>
```

関連項目

– 共通フィールド表示変数（4-27 ページ）
– その他のフィールド表示変数（4-29 ページ）

共通フィールド表示変数

コンテンツ・サーバー・ページで、メタデータ・フィールドの表示に影響するページ変数には、共通に使用されるいくつかの変数があります。これらの変数は、使用方法に応じて、ページ上の異なる場所で、異なる構文を使用して設定できます。

次の書式を使用して、特殊なフィールド表示変数を設定できます。

- **名前/値ペア:** 標準の`variable=value`書式を使用して変数を設定します。たとえば、`isHidden=1` のように設定します。一般に、この書式は、ループ処理のコードによってフィールドが生成されるページ部分で、現在のメタデータ・フィールドの表示を設定するために使用されます。
- **フィールド名 : 変数の書式:** 変数が適用されるメタデータ・フィールドのパラメータとして定義して、変数を設定します。たとえば、`myMetadata:isHidden` のように設定します。一般に、この書式はページの先頭で使用され、特定のメタデータ・フィールドのグローバル表示を設定します。

技術ヒント : 共通のフィールド表示変数をテンプレート・ページの先頭で設定する場合は、<HEAD> タグより前に変数を置く必要があります。変数を <HEAD></HEAD> セクション内またはセクション後に置くと、フィールドは予定どおりに表示されます（または非表示になります）が、ヘッダーの JavaScript 検証コードが評価されるため、チェックイン・ページを表示しようとすると、「...is not an object」というエラーがスローされます。

共通フィールド表示変数のリストを次に示します。

isExcluded（6-392 ページ）
isHidden（6-406 ページ）
isInfoOnly（6-408 ページ）
isRelocated（6-432 ページ）
maxLength（6-521 ページ）
optionListScript（6-568 ページ）

技術ヒント：特殊な include 変数の使用

これらの共通フィールド表示変数によって必要な柔軟性が十分に得られない場合は、かわりに使用するリソース・インクルードの名前をフィールド変数に設定して、メタデータ・フィールドの実装全体を置き換えることができます（たとえば、`myField:include=customInclude`）。

標準の実装は、変数 *defaultFieldInclude* によって参照されます。この変数の値は、チェックイン / 更新、問合せまたは情報ページでフィールドが生成されたかどうかに応じて異なります。また、表示されるフィールドのタイプによっても大きく異なります。標準のフィールド・インクルードを上書きする場合は、新しい実装で、JavaScript 検証および Upload アプレットなど、異なるページに関するすべての問題を考慮する必要があります。

このアプローチは、最後の手段としてのみ使用してください。既存の機能を拡張し、ローカル変数を設定してカスタム機能を持つことがより好ましい方法です。

この方法で include タグを使用して、特殊なメタデータ・フィールドのカスタム HTML コードを挿入する場合は、ページの </HEAD> タグの後に include 文を置く必要があります。</HEAD> タグより前に置くと、コンテンツ・サーバーはカスタム HTML コードをヘッダー内に挿入し、JavaScript として読取りを試みます。

例

共通変数の使用方法の例は、4-29 ページの「その他のフィールド表示変数」を参照してください。

関連項目

– フィールド情報変数（4-26 ページ）
– その他のフィールド表示変数（4-29 ページ）

その他のフィールド表示変数

前述の変数に加えて、メタデータ・フィールドの表示に影響を与える多数の変数があります。通常、これらの変数は、現在生成されているフィールドおよび関連する共通フィールド表示変数の値に応じて、メタデータ・フィールドの表示を定義するために使用されます。

その他のフィールド表示変数のリストを次に示します。

addEmptyOption（6-10 ページ）
captionEntryWidth（6-62 ページ）
captionFieldWidth（6-63 ページ）
defaultFieldInclude（6-132 ページ）
defaultOptionListScript（6-142 ページ）
fieldCaptionInclude（6-217 ページ）
fieldCaptionStyle（6-218 ページ）
fieldEditWidth（6-220 ページ）
fieldEntryInclude（6-221 ページ）
fieldExtraScriptInclude（6-222 ページ）
fieldInclude（6-223 ページ）
fieldMaxLength（6-225 ページ）
fieldValueStyle（6-230 ページ）
hasOptionList（6-285 ページ）
isFieldExcluded（6-397 ページ）
isFieldHidden（6-398 ページ）
isFieldInfoOnly（6-399 ページ）
isFieldMemo（6-400 ページ）
isMultiOption（6-420 ページ）
isStrictList（6-439 ページ）
NotationForRequiredFields（6-548 ページ）
optionListKey（6-564 ページ）
optionListName（6-565 ページ）
optionListResultSet（6-567 ページ）
optionListValueInclude（6-569 ページ）
optionsAllowPreselect（6-571 ページ）
StyleForRequiredFields（6-783 ページ）

例

この例では、*<install_dir>/shared/config/resources/std_page.htm* リソース・ファイルの *compute_std_field_overrides* インクルードが、現在生成中のフィールドをどのような方法で非表示のフィールドか、参考情報か、除外するフィールドか、あるいは再配置されるフィールドかを判断する方法を示します。このコードは、ページの各メタデータ・フィールドの生成時にループ処理されます。

```
<@dynamichtml compute_std_field_overrides@>
<$isCustomHidden = getValue("#active", fieldName & ":isHidden")$>
<$if isHidden or isCustomHidden$>
    <$isFieldHidden = 1$>
<$else$>
    <$isFieldHidden = ""$>
<$endif$>
<$isCustomInfo = getValue("#active", fieldName & ":isInfoOnly")$>
<$if isInfo or isCustomInfo or isFieldHidden or isInfoOnly$>
    <$isFieldInfoOnly = 1$>
<$else$>
    <$isFieldInfoOnly = ""$>
<$endif$>
<$isCustomExcluded = getValue("#active", fieldName & ":isExcluded")$>
<$isCustomRelocated = getValue("#active", fieldName & ":isRelocated")$>
<$if isCustomExcluded or (isCustomRelocated and not isRelocated) or isExcluded
or (isFieldHidden and not isFormSubmit)$>
    <$isFieldExcluded = 1$>
<$endif$>
<@end@>
```

関連項目

– フィールド情報変数（4-26 ページ）
– 共通フィールド表示変数（4-27 ページ）

読取り専用変数

読取り専用変数を使用して、現在のテンプレート、現在のログイン・ユーザーやその他の現在の設定に関する情報を収集できます。これらの変数は読取り専用であり、値を割り当てることはできません。

- テンプレート読取り専用変数（4-31 ページ）
- テンプレート読取り専用変数の例（4-32 ページ）
- ユーザー読取り専用変数（4-33 ページ）
- コンテンツ読取り専用変数（4-33 ページ）
- その他の読取り専用変数（4-33 ページ）

テンプレート読取り専用変数

テンプレート関連の読取り専用変数を使用すると、テンプレートのアイデンティティに基づいてテンプレート内に条件付きコンテンツを作成できます。これらの事前定義済の変数により、任意のテンプレートのクラス、ファイル・パス、名前またはタイプをコンテンツ・サーバーの Web ページに表示できます。これは Web サイトを開発する際に特に役立ちます。

次の読取り専用変数は、テンプレートに関連があります。

TemplateClass（6-796 ページ）
TemplateFilePath（6-798 ページ）
TemplateName（6-799 ページ）
TemplateType（6-800 ページ）

関連項目

– テンプレート読取り専用変数の例（4-32 ページ）

テンプレート読取り専用変数の例

この例では、すべてのコンテンツ・サーバー Web ページの左側のサイドバーにある「Administration」リンクの下に、テンプレートの内部名が表示されます。これを表示するために、「Administration」リンクを定義する *pne_nav_admin_links* インクルードに、事前定義済の変数 *TemplateName*（次の例の太字部分）が追加されました。

図 4-1　事前定義済の変数 *TemplateName* を使用して、Web ページに内部テンプレート名を表示する例

```
<$if IsSubAdmin$>
<tr>
    <td>
        <a href="<$HttpCgiPath$>?IdcService=GET_ADMIN_PAGE&Action=
        GetTemplatePage&Page=ADMIN_LINKS"
        OnMouseOver="imgAct('admin')"
        OnMouseOut="imgInact('admin')">
        <img src="<$HttpImagesRoot$>
        <$button_admin_grey_ish_image$>" width="<$navImageWidth$>"
        height="<$navImageHeight$>" name="admin" border="0"
        alt="<$lc("wwProductAdministration", ProductID)$>"></a>
    </td>
        <td>
        <a class=pneHeader href="<$HttpCgiPath$>?IdcService=
        GET_ADMIN_PAGE&Action=GetTemplatePage&Page=ADMIN_LINKS"
        OnMouseOver="imgAct('admin')"
        OnMouseOut="imgInact('admin')">
        <$lc("wwAdministration")$></a>
    </td>
</tr>
<tr>
    <td colspan=2><font color=#FFFFFF style="Arial"
    size="-1"><$TemplateName$></font></td>
</tr>
<$endif$>
```

関連項目

– テンプレート読取り専用変数（4-31 ページ）

ユーザー読取り専用変数

ユーザー関連の読取り専用変数を使用して、現在のユーザーに関する情報を収集できます。

次の読取り専用変数は、ユーザーに関連があります。

DelimitedUserRoles（6-147 ページ）
ExternalUserAccounts（6-214 ページ）
ExternalUserRoles（6-215 ページ）
UserAccounts（6-849 ページ）
UserAddress（6-850 ページ）
UserAppRights（6-851 ページ）
UserDefaultAccount（6-854 ページ）
UserFullName（6-858 ページ）
UserName（6-865 ページ）
UserRoles（6-867 ページ）

コンテンツ読取り専用変数

1 つのコンテンツ関連の読取り専用変数 SourceID（6-753 ページ）により、現在の動的サーバー・ページのコンテンツ ID を取得できます。

この変数によって返される値は、`ref:dID` と同じです。3-36 ページの「動的サーバー・ページでのメタデータの参照」を参照してください。

その他の読取り専用変数

次の変数は、コンテンツ・サーバーの内部フラグとしてのみ設定されます。そのため、取得することは可能ですが、直接に設定できません。

SafeDir（6-695 ページ）

設定可能変数

設定可能変数は、スクリプト内に設定するか、CGI 文字列内で使用できます。たとえば、変数 *ScriptDebugTrace* をサービス・コールのパラメータとして使用して、デバッグ・トレース情報をページに表示できます。これらの変数の 1 つを設定して、ページのコンテンツを変更できます。

次は、Idoc スクリプトの設定可能変数の完全なリストです。

ClientControlled（6-74 ページ）
coreContentOnly（6-109 ページ）
getCookie（6-260 ページ）
HasLocalCopy（6-284 ページ）
IsJava（6-410 ページ）
IsSavedQuery（6-436 ページ）
IsSoap（6-438 ページ）
IsXml（6-456 ページ）
ScriptDebugTrace（6-709 ページ）
ScriptErrorTrace（6-710 ページ）
setCookie（6-732 ページ）

値変数

値変数は、サービス・コールの最初に一度評価され、サービス・コール全体にわたってその値が使用されます。変数は、新しいサービス・コールごとに再評価されます。対照的に、動的変数は、変数が出現するたびに評価されます。たとえば、値変数 *isNew* は、チェックインの際に、そのコンテンツ・アイテムが新規であるか、あるいはリビジョンであるかを評価します。この評価は、チェックイン・サービスのコールを通して使用されます。

次は、Idoc スクリプトの値変数の完全なリストです。

AuthorAddress（6-47 ページ）
BrowserVersionNumber（6-57 ページ）
CURRENT_DATE（6-114 ページ）
CURRENT_ROW（6-115 ページ）
DocTypeSelected（6-176 ページ）
DocUrl（6-177 ページ）
DownloadSuggestedName（6-182 ページ）
fileUrl（6-233 ページ）
FIRSTREV（6-235 ページ）
HasOriginal（6-286 ページ）
htmlRefreshTimeout（6-297 ページ）
htmlRefreshUrl（6-298 ページ）

HttpAbsoluteCgiPath（6-307 ページ）
HttpAbsoluteCgiRoot（6-308 ページ）
HttpAbsoluteWebRoot（6-309 ページ）
HttpAdminCgiPath（6-310 ページ）
HttpBrowserFullCgiPath（6-311 ページ）
HttpCgiPath（6-312 ページ）
HttpCommonRoot（6-313 ページ）
HttpEnterpriseCgiPath（6-314 ページ）
HttpHelpRoot（6-315 ページ）
HttpImagesRoot（6-317 ページ）
HttpLayoutRoot（6-318 ページ）
HttpRelativeAdminRoot（6-319 ページ）
HttpRelativeCgiRoot（6-320 ページ）
HttpSharedRoot（6-323 ページ）
HttpSystemHelpRoot（6-324 ページ）
HttpWebRoot（6-325 ページ）
IdcRegistrationPath（6-339 ページ）
IsEditRev（6-391 ページ）
IsFailedConversion（6-394 ページ）
IsFailedIndex（6-395 ページ）
isNew（6-422 ページ）
IsNotLatestRev（6-423 ページ）
MSIE（6-537 ページ）
NoMatches（6-544 ページ）
OneMatch（6-559 ページ）
PageParent（6-575 ページ）
ResultsTitle（6-651 ページ）
StatusCode（6-756 ページ）
StatusMessage（6-757 ページ）
UseHtmlOrTextHighlightInfo（6-843 ページ）

WEB サーバー変数

Web サーバー変数は、サーバーがゲートウェイ・プログラムを実行するときに設定される CGI 環境変数です。情報リクエストに関するデータをサーバーからスクリプトに渡すために、サーバーでは、コマンドライン引数および環境変数を使用します。これらの環境変数は、ログ・ファイルへの情報の出力や、Idoc スクリプトの文中および評価の一部として使用できます。

たとえば、次の Idoc スクリプトの文は、リモート・ホスト・アドレスが特定の文字列と一致するかどうかを評価します。

```
<$if strEquals("207.0.0.1",REMOTE_HOST)$>
```

次の HTML および Idoc スクリプトのマークアップは、ページに Web サーバー環境情報のリストを表示します。

```
<P>HTTP_INTERNETUSER=<$HTTP_INTERNETUSER$></P>
<P>REMOTE_HOST=<$REMOTE_HOST$></P>
<P>SCRIPT_NAME=<$SCRIPT_NAME$></P>
```

次は、Web サーバー変数の完全なリストです。

CONTENT_LENGTH（6-104 ページ）
GATEWAY_INTERFACE（6-258 ページ）
HTTP_ACCEPT（6-299 ページ）
HTTP_ACCEPT_ENCODING（6-300 ページ）
HTTP_ACCEPT_LANGUAGE（6-301 ページ）
HTTP_COOKIE（6-302 ページ）
HTTP_HOST（6-303 ページ）
HTTP_INTERNETUSER（6-304 ページ）
HTTP_REFERER（6-305 ページ）
HTTP_USER_AGENT（6-306 ページ）
PATH_INFO（6-581 ページ）
PATH_TRANSLATED（6-582 ページ）
QUERY_STRING（6-633 ページ）
REMOTE_ADDR（6-641 ページ）
REMOTE_HOST（6-642 ページ）
REQUEST_METHOD（6-643 ページ）
SCRIPT_NAME（6-708 ページ）
SERVER_NAME（6-727 ページ）
SERVER_PORT（6-728 ページ）
SERVER_PROTOCOL（6-729 ページ）
SERVER_SOFTWARE（6-730 ページ）

ワークフロー・スクリプト

Idoc スクリプトには、特にワークフローで使用される事前定義済の関数および変数が含まれます。

- 4-37 ページの「ワークフローの概要」では、ワークフローでの Idoc スクリプトの使用の概要を説明します。
- ワークフロー関数（4-38 ページ）は、ワークフローに関連するアクションを実行または結果を返します。
- ワークフロー変数（4-39 ページ）では、ワークフローに関連する変数の値を設定できます。

ワークフローの概要

Content Server でのワークフローの実装方法の詳細は、『ワークフロー・インプリメンテーション・ガイド』を参照してください。

次にワークフローでの Idoc スクリプトの使用を概説します。

- ワークフロー・ジャンプは、特定のステップ・イベント（エントリ、更新または終了）のために定義された Idoc スクリプトの評価によって開始されます。
- リビジョンが各ステップに移動すると、ワークフロー内のリビジョンの状態に関する情報を保持するコンパニオン・ファイルが作成されます。ユーザー定義のオプションに加えて、実行されたステップ、最後のエントリ時間、特定のワークフロー・ステップのエントリ回数など、リビジョンの履歴も保持されます。
  - ステップ間でのリビジョンの移動に伴い、グローバルな状態情報が維持されます。
  - ローカライズされた状態情報がステップとともに保存され、そのステップでリビジョンを処理するときに利用できます。
- コンパニオン・ファイルでは、キーを使用してワークフローの状態情報を追跡します。次にキーの構文を示します。

  ```
  step_name@workflow_name.variable=value
  ```

  たとえば、*Marketing* というワークフローの *Editor* ステップに関するエントリ・カウント変数と最終エントリ変数の値を定義するキーは次のようになります。

  ```
  Editor@Marketing.entryCount=1
  Editor@Marketing.lastEntryTs={ts '2002-05-28 16:57:00'}
  ```

- すべてのワークフロー・スクリプトの評価は、データベース・トランザクション内で発生します。したがって、重大なエラーまたは中断が発生しても、データベースやコンパニオン・ファイルは変更されません。これは、Idoc スクリプト・ワークフロー関数の所要時間を短くする必要があることを意味します。このため、外部プロセスをトリガーするには、Idoc スクリプト関数を専用のスレッドで実行するように作成する必要があります。

警告 : Idoc スクリプトまたはカスタム・コンポーネントを使用してワークフロー情報をローカル・データにロードする場合、データが破損する可能性があることに注意してください。これは特に、現在のリビジョンとは異なるリビジョンの情報をロードする場合に重要です。

ワークフロー関数

ワークフロー関数は、ワークフローに関連するアクションを実行または結果を返します。次は、ワークフロー関数の完全なリストです。

wfAddActionHistoryEvent（6-913 ページ）
wfAddUser（6-915 ページ）
wfComputeStepUserList（6-916 ページ）
wfCurrentGet（6-917 ページ）
wfCurrentSet（6-918 ページ）
wfCurrentStep（6-919 ページ）
wfDisplayCondition（6-920 ページ）
wfExit（6-921 ページ）
wfGet（6-922 ページ）
wfIsFinishedDocConversion（6-925 ページ）
wfIsNotifyingUsers（6-926 ページ）
wfIsReleasable（6-927 ページ）
wfLoadDesign（6-933 ページ）
wfNotify（6-936 ページ）
wfSet（6-939 ページ）
wfSetIsNotifyingUsers（6-940 ページ）
wfUpdateMetaData（6-942 ページ）

ワークフロー変数

ワークフロー変数を使用して、ワークフローに関連する値を設定します。次は、ワークフロー変数の完全なリストです。

dWfName（6-188 ページ）
dWfStepName（6-189 ページ）
entryCount（6-204 ページ）
lastEntryTs（6-470 ページ）
wfAction（6-912 ページ）
wfAdditionalExitCondition（6-914 ページ）
wfGetStepTypeLabel（6-923 ページ）
wfJumpEntryNotifyOff（6-928 ページ）
wfJumpMessage（6-929 ページ）
wfJumpName（6-930 ページ）
wfJumpReturnStep（6-931 ページ）
wfJumpTargetStep（6-932 ページ）
wfMailSubject（6-934 ページ）
wfMessage（6-935 ページ）
wfParentList（6-937 ページ）
wfReleaseDocument（6-938 ページ）
WfStart（6-941 ページ）

5

使用方法別 Idoc スクリプト

この章では、Idoc スクリプトの変数および関数を一般的な使用方法に応じて分類します。

- アーカイバ（5-3 ページ）
- バッチ・ローダー（5-3 ページ）
- クライアント（5-4 ページ）
- クラスタリング（5-5 ページ）
- Collaboration Manager（5-5 ページ）
- コンテンツ・アイテム（5-6 ページ）
- コンテンツ・プロファイル（5-7 ページ）
- コンテンツ・サーバー（5-8 ページ）
- 変換（5-11 ページ）
  - Inbound Refinery（5-11 ページ）
  - PDF Converter（5-12 ページ）
  - XML Converter（5-16 ページ）
  - Dynamic Converter（5-17 ページ）
  - Compression（5-18 ページ）
- データベース（5-18 ページ）
- 日付と時間（5-19 ページ）
- デバッグ（5-20 ページ）
- ディレクトリおよびパス（5-22 ページ）
- Enterprise Search（5-24 ページ）

- フィールド表示（5-24 ページ）
- Folders および WebDAV（5-26 ページ）
- HTML プレビュー（5-26 ページ）
- Idoc スクリプト（5-27 ページ）
- 索引付け（5-28 ページ）
- JSP（5-29 ページ）
- ローカライゼーション（5-29 ページ）
- ページ表示（5-30 ページ）
- パーソナライズ（5-31 ページ）
- Pop-Up Calendar（5-31 ページ）
- Record Manager および Retention Manager（5-32 ページ）
- ResultSet（5-32 ページ）
- スキーマ（5-33 ページ）
- 検索（5-34 ページ）
- セキュリティ（5-36 ページ）
  - 内部セキュリティ（5-36 ページ）
  - 外部セキュリティ（5-37 ページ）
- 文字列（5-39 ページ）
- テンプレート（5-40 ページ）
- ユーザー（5-40 ページ）
- Web サーバー（5-42 ページ）
- ワークフロー（5-44 ページ）

アーカイバ

次の Idoc スクリプト変数は、コンテンツ・サーバー・アーカイバに関連があります。

構成変数

AllowMatchesInDateCheck（6-29 ページ）
ARCHIVE_COLLECTION（6-41 ページ）
ArchiverDoLocks（6-42 ページ）
ArchiverTouchMonitorInterval（6-43 ページ）
ArchiverTraceLockMaintenance（6-44 ページ）
ArchiverTraceLocks（6-45 ページ）
CollectionLocation（6-84 ページ）
IsAutoArchiver（6-375 ページ）
LocaleArchiverDateFormat（6-504 ページ）
MaxArchiveErrorsAllowed（6-514 ページ）
OdbcDateFormat（6-558 ページ）

バッチ・ローダー

次の Idoc スクリプト変数は、コンテンツ・サーバー・バッチ・ローダーに関連があります。

構成変数

BatchLoaderPath（6-53 ページ）
BatchLoaderUserName（6-54 ページ）
CleanUp（6-72 ページ）
EnableErrorFile（6-195 ページ）
MaxErrorsAllowed（6-518 ページ）

クライアント

次の Idoc スクリプト変数は、コンテンツ・サーバーと連携するクライアント・アプリケーションに関連があります。

構成変数

DownloadApplet（6-181 ページ）
MacSupportsSignedApplets（6-511 ページ）
MultiUpload（6-539 ページ）
showDocActions（6-742 ページ）
UploadApplet（6-826 ページ）

その他の変数

BrowserVersionNumber（6-57 ページ）
ClientControlled（6-74 ページ）
HasLocalCopy（6-284 ページ）
HeavyClient（6-290 ページ）
HttpBrowserFullCgiPath（6-311 ページ）
IsMac（6-418 ページ）
IsSun（6-441 ページ）
IsUploadSockets（6-448 ページ）
IsWindows（6-454 ページ）
MSIE（6-537 ページ）

クラスタリング

次の Idoc スクリプト変数は、クラスタ環境での Content Server の使用に関連があります。

構成変数

AdminDataDir（6-17 ページ）
ClusterBinDirRule（6-75 ページ）
ClusterNodeAddress（6-76 ページ）
ClusterNodeName（6-77 ページ）
ConfigDir（6-102 ページ）
DataDir（6-120 ページ）
DisableSharedCacheChecking（6-160 ページ）
IdcServerBindAddress（6-340 ページ）
IdcServerSocketQueueDepth（6-341 ページ）
SharedDir（6-739 ページ）
UserPublishCacheDir（6-866 ページ）

COLLABORATION MANAGER

次の Idoc スクリプト変数は、Stellent Collaboration Manager の使用に関連があります。

actionItemInterval（6-9 ページ）
AllowMixedCaseACL（6-30 ページ）
cmDiscussionDocType（6-78 ページ）
cmMemoFieldSize（6-79 ページ）
cmZonedSecurityFields（6-80 ページ）
deleteActionItemInterval（6-145 ページ）
deleteWatchdogInterval（6-146 ページ）
DisableMeetingCenter（6-158 ページ）
parseClbraProject（6-576 ページ）
prjUseCustomRouting（6-597 ページ）
prjWizardTokens（6-598 ページ）
prjWizardWorkflowField（6-599 ページ）
SelectAccessMemberAllowManualEntry（6-723 ページ）
SelectAccessMemberResultLimit（6-724 ページ）
ShowWebDavUrlInBrowserRequest（6-747 ページ）
UserSelectAppletColumnOption（6-868 ページ）
UserSelectAppletDatabaseCaseSensitive（6-869 ページ）
UserSelectAppletDatabaseIso8859DictionaryOrder（6-870 ページ）

UserSelectAppletExternalAliasDefaultPermissions（6-871 ページ）
UserSelectAppletExternalUserDefaultPermissions（6-872 ページ）
UserSelectAppletForceEntriesToLowerCase（6-873 ページ）
UserSelectAppletInternalAliasDefaultPermissions（6-874 ページ）
UserSelectAppletInternalUserDefaultPermissions（6-875 ページ）
UseUserSelectApplet（6-882 ページ）
watchdogInterval（6-907 ページ）

コンテンツ・アイテム

次の Idoc スクリプト変数および関数は、コンテンツ・アイテムに関連があります。

構成変数

AllowAlternateMetaFile（6-23 ページ）
AllowPrimaryMetaFile（6-33 ページ）
AllowSamePrimaryAlternateExtensions（6-36 ページ）
AutoNumberPrefix（6-51 ページ）
AutoNumberWidth（6-52 ページ）
computeDocUrl（6-99 ページ）
createAlternateMetaFile（6-110 ページ）
createPrimaryMetaFile（6-112 ページ）
EnableExpirationNotifier（6-196 ページ）
ExpireRevisionOnly（6-213 ページ）
ForceDistinctRevLabels（6-237 ページ）
IsAutoNumber（6-376 ページ）
MajorRevSeq（6-513 ページ）
MinorRevSeq（6-535 ページ）
NotificationIntervalInDays（6-549 ページ）
NotificationMaximum（6-550 ページ）
NotificationQuery（6-551 ページ）
NotifyExtras（6-553 ページ）
NotifyTime（6-554 ページ）

グローバル関数

computeRenditionUrl（6-100 ページ）

その他の変数

CONTENT_LENGTH（6-104 ページ）
DocTypeSelected（6-176 ページ）
DocUrl（6-177 ページ）
DownloadSuggestedName（6-182 ページ）
fileUrl（6-233 ページ）
FIRSTREV（6-235 ページ）
FormInheritMetaDataFieldList（6-254 ページ）
HasLocalCopy（6-284 ページ）
HasOriginal（6-286 ページ）
HasUrl（6-289 ページ）
IsCriteriaSubscription（6-384 ページ）
IsEditRev（6-391 ページ）
IsFailedConversion（6-394 ページ）
IsFailedIndex（6-395 ページ）
IsFilePresent（6-401 ページ）
IsFullTextIndexed（6-405 ページ）
IsLocalSearchCollectionID（6-416 ページ）
isNew（6-422 ページ）
IsNotLatestRev（6-423 ページ）
IsNotSyncRev（6-424 ページ）
NumAdditionalRenditions（6-556 ページ）
SingleGroup（6-748 ページ）
SourceID（6-753 ページ）
VDKSUMMARY（6-899 ページ）

コンテンツ・プロファイル

次の関数を使用して、コンテンツ・プロファイルを管理します。

dpGet（6-183 ページ）
dpPromote（6-184 ページ）
dpPromoteRs（6-185 ページ）
dpSet（6-186 ページ）
DisableContentProfiles（6-152 ページ）
getFieldViewValue（6-266 ページ）
getRequiredMsg（6-271 ページ）
IsDpSubmitErrorFatal（6-387 ページ）
loadDocumentProfile（6-497 ページ）
utLoadDocumentProfiles（6-895 ページ）

コンテンツ・サーバー

次の Idoc スクリプト変数は、コンテンツ・サーバーに関連があります。

構成変数

AdditionalSubscriptionCollatedFieldList（6-14 ページ）
AppletChunkSize（6-39 ページ）
AppletChunkThreshold（6-40 ページ）
ChunkedRequestTrace（6-70 ページ）
CookieAuthTimeoutInDays（6-107 ページ）
CookieProxyPassword（6-108 ページ）
DefaultHtmlRefreshTimeoutInSeconds（6-137 ページ）
DefaultPasswordEncoding（6-143 ページ）
DisableAmberLayouts（6-149 ページ）
DisableClassicLayoutPublishing（6-150 ページ）
DisableCollegiateSkinPublishing（6-151 ページ）
DisableDBLongIntegerField（6-153 ページ）
DisableHttpUploadChunking（6-156 ページ）
DisableSharedCacheChecking（6-160 ページ）
DisableStellent05SkinPublishing（6-161 ページ）
DisableStellentSkinPublishing（6-162 ページ）
DisableTopMenusPublishing（6-163 ページ）
DisableTraysPublishing（6-164 ページ）
DisableWindowsPublishing（6-165 ページ）
DoDocNameOrder（6-179 ページ）
DownloadApplet（6-181 ページ）
FileEncoding（6-232 ページ）
getFieldConfigValue（6-264 ページ）
HasExternalUsers（6-282 ページ）
HasGlobalUsers（6-283 ページ）
HttpAbsoluteCgiRoot（6-308 ページ）
HttpAbsoluteWebRoot（6-309 ページ）
HttpLayoutRoot（6-318 ページ）
HttpRelativeAdminRoot（6-319 ページ）
HttpRelativeCgiRoot（6-320 ページ）
HttpSharedRoot（6-323 ページ）
HttpSystemHelpRoot（6-324 ページ）
IDC_Name（6-326 ページ）
IdcAdminServerHostName（6-327 ページ）
IdcAdminServerHostName（6-328 ページ）
IdcAdminServerPort（6-329 ページ）

IdcCommandFile（6-332 ページ）
IdcCommandLog（6-333 ページ）
IdcCommandServerHost（6-334 ページ）
IdcCommandUser（6-335 ページ）
IdcServerSocketQueueDepth（6-341 ページ）
InstanceDescription（6-366 ページ）
InstanceMenuLabel（6-367 ページ）
IntradocServerHostName（6-370 ページ）
IntradocServerPort（6-371 ページ）
isComponentEnabled（6-381 ページ）
IsContributor（6-383 ページ）
IsFormsPresent（6-402 ページ）
isLayoutEnabled（6-415 ページ）
IsProxiedServer（6-430 ページ）
IsWebServerPagesOnly（6-453 ページ）
isValidateFile（6-451 ページ）
JvmCommandLine（6-469 ページ）
LMDefaultLayout（6-491 ページ）
LMDefaultSkin（6-492 ページ）
lmGetLayout（6-493 ページ）
lmGetSkin（6-494 ページ）
LogMessagePrefix（6-508 ページ）
LogName（6-509 ページ）
MacSupportsSignedApplets（6-511 ページ）
MailServer（6-512 ページ）
MaxIntegerAllowed（6-520 ページ）
MaxQueryRows（6-525 ページ）
MaxRequestThreadCount（6-526 ページ）
MaxSizeResourceFileCacheMillions（6-530 ページ）
MaxStandardDatabaseResults（6-531 ページ）
MinIntegerAllowed（6-533 ページ）
MultiUpload（6-539 ページ）
NoAutomation（6-542 ページ）
PrimaryWorkQueueTimeout（6-594 ページ）
ProviderClass（6-618 ページ）
ProviderConfig（6-619 ページ）
ProviderConnection（6-620 ページ）
ProviderType（6-622 ページ）
ShowCreateExternalMenu（6-741 ページ）
SkipMozillaMailToWorkaround（6-749 ページ）
SmtpPort（6-750 ページ）
SubscriptionCollatedFieldList（6-785 ページ）

SubscriptionMethod（6-786 ページ）
SysAdminAddress（6-787 ページ）
UploadApplet（6-826 ページ）
UseCustomModaling（6-837 ページ）
UseDCLApplet（6-839 ページ）
UseMicrosoftVM（6-846 ページ）
UserCacheTimeout（6-853 ページ）
UseRevisionCreateDate（6-856 ページ）
WebProxyAdminServer（6-910 ページ）
XMLEncodingMode（6-946 ページ）

その他の変数

HttpAbsoluteCgiPath（6-307 ページ）
HttpAdminCgiPath（6-310 ページ）
HttpCgiPath（6-312 ページ）
HttpCommonRoot（6-313 ページ）
HttpEnterpriseCgiPath（6-314 ページ）
HttpHelpRoot（6-315 ページ）
HttpImagesRoot（6-317 ページ）
IsPromptingForLogin（6-429 ページ）
IsRequestError（6-434 ページ）
StatusCode（6-756 ページ）
StatusMessage（6-757 ページ）

変換

この項では、様々な変換製品に関連する Idoc スクリプトの変数および関数について説明します。

- Inbound Refinery（5-11 ページ）
- PDF Converter（5-12 ページ）
- XML Converter（5-16 ページ）
- Dynamic Converter（5-17 ページ）
- AutoCAD（5-14 ページ）
- Compression（5-18 ページ）

Inbound Refinery

次の Idoc スクリプト変数は、Inbound Refinery に関連があります。

構成変数

AllowPassthru（6-32 ページ）
AllowSecondaryConversion（6-37 ページ）
ComputerName（6-101 ページ）
ConnectionName（6-103 ページ）
CustomConversionWaitTime（6-116 ページ）
CustomConverterPath（6-117 ページ）
DefaultFilterInputFormat（6-133 ページ）
DefaultFilterOutputFormat（6-134 ページ）
DefaultGraphicTimeout（6-135 ページ）
DefaultNativeTimeout（6-139 ページ）
DocConverterEngineDir（6-171 ページ）
EnableCopyConvertCopy（6-192 ページ）
ExcludeConversionFilter（6-207 ページ）
ExclusiveIncludeConversionFilter（6-209 ページ）
FileEncoding（6-232 ページ）
IsAutoQueue（6-377 ページ）
IsContentRefineryPresent（6-382 ページ）
JvmCommandLine（6-469 ページ）
LogWithConnectionName（6-510 ページ）
MaxNumberConvertedHistoryShow（6-523 ページ）
MaxNumRecursiveStepDefinitions（6-524 ページ）
QueueDir（6-634 ページ）

PostConversionTimeOut（6-586 ページ）
PreconversionPath（6-590 ページ）
PreconversionWaitTime（6-591 ページ）
SecondaryConversionStepName（6-722 ページ）
SharedDir（6-739 ページ）
ThumbnailSizingMethod（6-807 ページ）
VaultDir（6-898 ページ）
VerboseMode（6-900 ページ）
WeblayoutDir（6-909 ページ）

その他の変数

IsFailedConversion（6-394 ページ）

PDF Converter

この項では、PDF Converter に関連する Idoc スクリプトの変数および関数のリストを示します。

❖ 全般（5-13 ページ）
❖ サムネイル（5-13 ページ）
❖ Microsoft Office（5-14 ページ）
❖ AutoCAD（5-14 ページ）
❖ InDesign（5-15 ページ）
❖ その他の形式（5-15 ページ）

全般

次の Idoc スクリプト変数は、PDF Converter に関連があります。

構成変数

DefaultPostscriptTimeout（6-144 ページ）
DisplayPSOrigin（6-166 ページ）
DistillerNormJobSetting（6-167 ページ）
DistillerOptJobSetting（6-168 ページ）
DistillerPath（6-169 ページ）
NumAdditionalRenditions（6-556 ページ）
OptimizePDF（6-563 ページ）
PdfVersion（6-583 ページ）
PostprocessPDFPath（6-587 ページ）
PostscriptPrinterName（6-588 ページ）
PrinterPortPath（6-595 ページ）
SetPdfVersion（6-736 ページ）
ShowHyperlinkBox（6-743 ページ）
UseAdobeDistiller（6-830 ページ）
UseLightweightToPostscriptStep（6-844 ページ）

サムネイル

次の Idoc スクリプト変数は、PDF Converter のサムネイル機能の使用に関連があります。

構成変数

CreatePDFThumbnails（6-111 ページ）
ImageAlchemyExePath（6-342 ページ）
IsThumbnailPresent（6-443 ページ）
ThumbnailDPI（6-803 ページ）
ThumbnailHeight（6-804 ページ）
ThumbnailJPEGQuality（6-805 ページ）
ThumbnailSizeLimit（6-806 ページ）
ThumbnailTemplate（6-808 ページ）
ThumbnailWidth（6-809 ページ）
UseAlchemy111（6-832 ページ）

Microsoft Office

次の Idoc スクリプト変数は、Microsoft Office アプリケーションの PDF Converter との使用に関連があります。

構成変数

AdjustPrinterMargins（6-15 ページ）
AllowSkippedHyperlinktoCauseIncomplete（6-38 ページ）
PowerPointPrintOutput（6-589 ページ）
ProcessExcelUrlHyperlinks（6-600 ページ）
ProcessPowerpointHyperlinks（6-606 ページ）
ProcessVisioInternalHyperlinks（6-607 ページ）
ProcessVisioUrlHyperlinks（6-608 ページ）
ProcessWordBookMarks（6-609 ページ）
ProcessWordComments（6-610 ページ）
ProcessWordEmailLinks（6-611 ページ）
ProcessWordFootEndNote（6-612 ページ）
ProcessWordHeadings（6-613 ページ）
ProcessWordHyperlinks（6-614 ページ）
ProcessWordToc（6-615 ページ）
ProcessWordUrlHyperlinks（6-616 ページ）

AutoCAD

次の Idoc スクリプト変数は、AutoCAD の PDF Converter との使用に関連があります。

構成変数

AcadUseLISPInterface（6-3 ページ）
AutoCad2000PlotterFilePath（6-49 ページ）
UseAutoCad2000（6-833 ページ）
UseAutocadModelSpace（6-834 ページ）

InDesign

次の Idoc スクリプト変数は、Adobe InDesign の PDF Converter との使用に関連があります。

構成変数

ProcessInDesignBookMarks（6-601 ページ）
ProcessInDesignHyperlinks（6-602 ページ）
ProcessInDesignInternalPDFThumbnail（6-603 ページ）
ProcessInDesignLinkUpdate（6-604 ページ）
ProcessInDesignPDFCompatibility50（6-605 ページ）

その他の形式

次の Idoc スクリプト変数は、様々なネイティブ・アプリケーションの PDF Converter との使用に関連があります。

構成変数

FrameMakerCyclesAllowed（6-255 ページ）
FrameMakerexePath（6-256 ページ）
FrameMakerTimeout（6-257 ページ）
MSPubexePath（6-538 ページ）
OpenOfficeListenerPort（6-560 ページ）
OpenOfficeListenerURL（6-561 ページ）
OpenOfficePath（6-562 ページ）
PageMakerExePath（6-574 ページ）

XML Converter

次の Idoc スクリプト変数は、XML Converter に関連があります。

構成変数

ConvertXmlAsWebviewable（6-106 ページ）
CreateXMLRendition（6-113 ページ）
IncludeXmlFlexionDocType（6-349 ページ）
UseSearchML（6-877 ページ）
UseXmlEmbeddings（6-884 ページ）
UseXmlExtensibleDTD（6-885 ページ）
UseXmlNoDTD（6-886 ページ）
UseXmlNoReference（6-887 ページ）
UseXmlPStyleNames（6-888 ページ）
UseXmlSchema（6-889 ページ）
UseXmlSchemaDTD（6-890 ページ）
XMLTimeout（6-947 ページ）

Dynamic Converter

次の Idoc スクリプト変数は、Dynamic Converter に関連があります。

構成変数

ConversionCacheExpirationPeriodInDays（6-105 ページ）
DatedCacheIntervalDays（6-122 ページ）
DCMaxFileSize（6-124 ページ）
dcShowExportLink（6-125 ページ）
DCTimeOut（6-126 ページ）
DCViewFormat（6-127 ページ）
DefaultFilterInputFormat（6-133 ページ）
DefaultFilterOutputFormat（6-134 ページ）
DefaultHtmlConversion（6-136 ページ）
DisableForcedConversions（6-154 ページ）
DisableKeepingForcedConversions（6-157 ページ）
DisableOptimizedLatestReleaseQuerying（6-159 ページ）
DynamicConverterHtmlFlavor（6-190 ページ）
EnableOptimizedLatestReleaseQuerying（6-198 ページ）
EnableRebuildForcesReconversion（6-199 ページ）
ForcedConversionRules（6-236 ページ）
HighlightBegin（6-293 ページ）
HighlightEnd（6-294 ページ）
IsDynamicConverterEnabled（6-389 ページ）
rule:IsFragmentOnly（6-404 ページ）
LatestReleasedTimestampTimeout（6-471 ページ）
MaxConversionCacheSizeInMegs（6-516 ページ）
NeverTimeoutForcedConversionsTemplateChange（6-541 ページ）
TimeoutChangedTemplateForcedConversionsInMins（6-810 ページ）
TimeoutMismatchedForcedConversionsInMins（6-811 ページ）
TraceConversionCacheCleanup（6-816 ページ）
UseCascadingStyleSheets（6-836 ページ）

グローバル関数

incDynamicConversionByRule（6-345 ページ）

Compression

次の Idoc スクリプト変数は、Compression に関連があります。

構成変数

AllowMrSidRetryCompressionLevel（6-31 ページ）
CompressionMinHeight（6-97 ページ）
CompressionMinWidth（6-98 ページ）
IsNTCompressionPresent（6-425 ページ）
MrSidLowestCompressionLevel（6-536 ページ）

データベース

次の Idoc スクリプトの変数および関数は、データベースに関連があります。

構成変数

CLASSPATH（6-71 ページ）
ColumnMapFile（6-96 ページ）
DatabasePreserveCase（6-118 ページ）
DatabaseWildcards（6-119 ページ）
ForceJdbcSync（6-241 ページ）
IsJdbc（6-411 ページ）
IsJdbcLockTrace（6-412 ページ）
IsJdbcQueryTrace（6-413 ページ）
JdbcConnectionString（6-459 ページ）
JdbcDriver（6-460 ページ）
JdbcPassword（6-461 ページ）
JdbcPasswordEncoding（6-462 ページ）
JdbcUser（6-463 ページ）
MemoFieldSize（6-532 ページ）
MinMemoFieldSize（6-534 ページ）
NumConnections（6-557 ページ）
UseDatabaseShortIndexName（6-838 ページ）

グローバル関数

formatDateDatabase（6-245 ページ）

日付と時間

次の Idoc スクリプトの変数および関数は、コンテンツ・サーバーでの日付と時間の書式設定および操作に関連があります。

構成変数

DateOutputFormat（6-123 ページ）
SystemDateFormat（6-788 ページ）
SystemTimeZone（6-794 ページ）
UseFourDigitYear（6-840 ページ）
WarnAboutTimeZone（6-906 ページ）

グローバル関数

dateCurrent（6-121 ページ）
formatDate（6-243 ページ）
formatDateDatabase（6-245 ページ）
formatDateOnly（6-247 ページ）
formatDateOnlyDisplay（6-249 ページ）
formatDateOnlyFull（6-250 ページ）
formatDateWithPattern（6-251 ページ）
formatTimeOnly（6-252 ページ）
parseDataEntryDate（6-577 ページ）
parseDate（6-578 ページ）
parseDateWithPattern（6-580 ページ）

その他の変数

CURRENT_DATE（6-114 ページ）
lastEntryTs（6-470 ページ）

デバッグ

次の Idoc スクリプトの変数および関数は、デバッグに関連があります。

構成変数

ArchiverTraceLocks（6-45 ページ）
ArchiverTraceLockMaintenance（6-44 ページ）
AutoMaxMergeTrace（6-50 ページ）
CGI_DEBUG（6-64 ページ）
CGI_RECEIVE_DUMP（6-65 ページ）
CGI_SEND_DUMP（6-66 ページ）
ChunkedRequestTrace（6-70 ページ）
DebugMode（6-128 ページ）
DebugStdConversion（6-129 ページ）
DirectoryLockingLogPath（6-148 ページ）
DisplayPSOrigin（6-166 ページ）
EnableEnvironmentPackager（6-194 ページ）
EnableErrorFile（6-195 ページ）
FILTER_DEBUG（6-234 ページ）
ForceJdbcSync（6-241 ページ）
getFreeMemory（6-267 ページ）
getTotalMemory（6-273 ページ）
IdcAnalyzeLogDir（6-330 ページ）
IndexerTraceCycleListChanges（6-360 ページ）
IndexerTraceLockFileChanges（6-361 ページ）
IndexerTraceStartStop（6-362 ページ）
isActiveTrace（6-372 ページ）
IsJdbcLockTrace（6-412 ページ）
IsJdbcQueryTrace（6-413 ページ）
isVerboseTrace（6-452 ページ）
IsXml（6-456 ページ）
LogDir（6-507 ページ）
LogMessagePrefix（6-508 ページ）
LogName（6-509 ページ）
LogWithConnectionName（6-510 ページ）
PLUGIN_DEBUG（6-584 ページ）
RequestAuditAdditionalVerboseFieldsList（6-644 ページ）
RequestAuditIntervalSeconds1（6-645 ページ）
RequestAuditIntervalSeconds2（6-646 ページ）
RequestAuditListDepth1（6-647 ページ）
RequestAuditListDepth2（6-648 ページ）

RequestAuditSubClassificationField（6-649 ページ）
SearchCacheTrace（6-715 ページ）
SearchDebugLevel（6-717 ページ）
SearchQueryTrace（6-721 ページ）
TempDir（6-795 ページ）
TraceIsVerbose（6-817 ページ）
TraceResourceConflict（6-818 ページ）
TraceResourceLoad（6-819 ページ）
TraceResourceOverride（6-820 ページ）
TraceSectionsList（6-821 ページ）
UseRedirectedOutput（6-855 ページ）
VerboseMode（6-900 ページ）

グローバル関数

getDebugTrace（6-262 ページ）
getErrorTrace（6-263 ページ）
trace（6-814 ページ）

その他の変数

IsJava（6-410 ページ）
IsRequestError（6-434 ページ）
IsSoap（6-438 ページ）
SafeDir（6-695 ページ）
ScriptDebugTrace（6-709 ページ）
ScriptErrorTrace（6-710 ページ）

ディレクトリおよびパス

次の Idoc スクリプトの変数および関数は、コンテンツ・サーバーのディレクトリおよびファイル・パスに関連があります。

構成変数

AdminDataDir（6-17 ページ）
BinDir（6-55 ページ）
CollectionLocation（6-84 ページ）
ConfigDir（6-102 ページ）
DataDir（6-120 ページ）
HelpDir（6-291 ページ）
HTMLEditorPath（6-296 ページ）
IdcAnalyzeLogDir（6-330 ページ）
IntradocDir（6-368 ページ）
isPhysicallySplitDir（6-427 ページ）
LogDir（6-507 ページ）
ProviderDir（6-621 ページ）
SearchDir（6-718 ページ）
SharedDir（6-739 ページ）
SharedWeblayoutDir（6-740 ページ）
SubjectsDir（6-784 ページ）
SystemReportsDir（6-791 ページ）
SystemResourcesDir（6-792 ページ）
SystemTemplatesDir（6-793 ページ）
TempDir（6-795 ページ）
UserCache（6-852 ページ）
VaultDir（6-898 ページ）
WebBrowserPath（6-908 ページ）
WeblayoutDir（6-909 ページ）
WorkflowDir（6-943 ページ）

グローバル関数

docRootFilename（6-175 ページ）
proxiedBrowserFullCgiWebUrl（6-623 ページ）
proxiedCgiWebUrl（6-624 ページ）

その他の変数

DownloadSuggestedName（6-182 ページ）
HttpAbsoluteCgiPath（6-307 ページ）
HttpAbsoluteCgiRoot（6-308 ページ）
HttpAbsoluteWebRoot（6-309 ページ）
HttpAdminCgiPath（6-310 ページ）
HttpBrowserFullCgiPath（6-311 ページ）
HttpCgiPath（6-312 ページ）
HttpCommonRoot（6-313 ページ）
HttpEnterpriseCgiPath（6-314 ページ）
HttpHelpRoot（6-315 ページ）
HttpImagesRoot（6-317 ページ）
HttpLayoutRoot（6-318 ページ）
HttpRelativeAdminRoot（6-319 ページ）
HttpRelativeCgiRoot（6-320 ページ）
HttpRelativeWebRoot（6-321 ページ）
HttpServerAddress（6-322 ページ）
HttpSharedRoot（6-323 ページ）
HttpSystemHelpRoot（6-324 ページ）
HttpWebRoot（6-325 ページ）
IdcRegistrationPath（6-339 ページ）
SafeDir（6-695 ページ）

ENTERPRISE SEARCH

次の Idoc スクリプト変数は、Enterprise Search に関連があります。

構成変数

EnterpriseSearchAsDefault（6-203 ページ）

フィールド表示

次の Idoc スクリプトの変数および関数は、コンテンツ・サーバー Web ページでのメタデータ・フィールドの表示に関連があります。

構成変数

ForceDocTypeChoice（6-238 ページ）
ForceSecurityGroupChoice（6-242 ページ）
IsOverrideFormat（6-426 ページ）
MemoFieldSize（6-532 ページ）
MinMemoFieldSize（6-534 ページ）
NotationForRequiredFields（6-548 ページ）
ShowOnlyKnownAccounts（6-744 ページ）
ShowRevisionsColumn（6-745 ページ）
StyleForRequiredFields（6-783 ページ）
UseDCLApplet（6-839 ページ）

グローバル関数

optList（6-572 ページ）

その他の変数

addEmptyOption（6-10 ページ）
captionEntryWidth（6-62 ページ）
captionFieldWidth（6-63 ページ）
defaultFieldInclude（6-132 ページ）
defaultOptionListScript（6-142 ページ）
fieldCaption（6-216 ページ）
fieldCaptionInclude（6-217 ページ）
fieldCaptionStyle（6-218 ページ）
fieldDefault（6-219 ページ）
fieldEditWidth（6-220 ページ）
fieldEntryInclude（6-221 ページ）
fieldExtraScriptInclude（6-222 ページ）
fieldInclude（6-223 ページ）
fieldIsOptionList（6-224 ページ）
fieldMaxLength（6-225 ページ）
fieldName（6-226 ページ）
fieldOptionListType（6-227 ページ）
fieldType（6-228 ページ）
fieldValue（6-229 ページ）
fieldValueStyle（6-230 ページ）
fieldWidth（6-231 ページ）
hasOptionList（6-285 ページ）
isExcluded（6-392 ページ）
isFieldExcluded（6-397 ページ）
isFieldHidden（6-398 ページ）
isFieldInfoOnly（6-399 ページ）
isFieldMemo（6-400 ページ）
isHidden（6-406 ページ）
isInfoOnly（6-408 ページ）
isMultiOption（6-420 ページ）
isRelocated（6-432 ページ）
isRequired（6-435 ページ）
isStrictList（6-439 ページ）
maxLength（6-521 ページ）
optionListKey（6-564 ページ）
optionListName（6-565 ページ）
optionListResultSet（6-567 ページ）
optionListScript（6-568 ページ）
optionListValueInclude（6-569 ページ）
optionsAllowPreselect（6-571 ページ）
requiredMsg（6-650 ページ）
valueStyle（6-897 ページ）

関連項目

– ページ表示（5-30 ページ）

FOLDERS および WEBDAV

次の Idoc スクリプト変数は、Folders および WebDAV のインストールおよび使用に関連があります。

CollectionDeleteEnabled（6-81 ページ）
CollectionHiddenMeta（6-82 ページ）
CollectionInhibitUpdateMeta（6-83 ページ）
CollectionMeta（6-85 ページ）
CollectionMoveEnabled（6-86 ページ）
CollectionReadOnlyMarkedFolders（6-87 ページ）
CollectionReadOnlyMeta（6-88 ページ）
CollectionReleasedOnly（6-89 ページ）
CollectionSecurityReadOnly（6-90 ページ）
CollectionTrashDeleteDate（6-91 ページ）
CollectionTrashDeleteLocation（6-92 ページ）
CollectionTrashDeleteOldName（6-93 ページ）
CollectionTrashDeleter（6-94 ページ）
CollectionWebDAVServer（6-95 ページ）
InitialColID（6-365 ページ）

HTML プレビュー

次の Idoc スクリプト変数は、HTML プレビューに関連があります。

構成変数

DtmPort（6-187 ページ）
PreviewOutputExtension（6-592 ページ）
PreviewPath（6-593 ページ）

IDOC スクリプト

次の Idoc スクリプトの関数およびキーワードは、特定のコンテンツ・サーバー機能とは関連がありません。

グローバル関数

break（6-56 ページ）
cacheInclude（6-60 ページ）
docLoadResourceIncludes（6-172 ページ）
eval（6-205 ページ）
executeService（6-211 ページ）
forceExpire（6-239 ページ）
getTextFile（6-272 ページ）
getUserValue（6-274 ページ）
getValue（6-275 ページ）
inc（6-343 ページ）
incGlobal（6-346 ページ）
incTemplate（6-350 ページ）
isFalse（6-396 ページ）
isTrue（6-444 ページ）
setExpires（6-733 ページ）
setHttpHeader（6-734 ページ）
setMaxAge（6-735 ページ）
setResourceInclude（6-737 ページ）

キーワード

c（6-58 ページ）
exec（6-210 ページ）
include（6-347 ページ）

索引付け

次の Idoc スクリプト変数は、索引付けに関連があります。

構成変数

AdditionalIndexBuildParams（6-11 ページ）
AllowConcurrentUpdate（6-26 ページ）
AutoMaxMergeTrace（6-50 ページ）
CheckMkvdkDocCount（6-69 ページ）
DateOutputFormat（6-123 ページ）
DoAutoMaxMerge（6-170 ページ）
IndexableRenditions（6-351 ページ）
IndexerAutoWorkInterval（6-352 ページ）
IndexerLargeFileSize（6-353 ページ）
IndexerPath（6-354 ページ）
IndexerRebuildStagingManualCopy（6-355 ページ）
IndexerRebuildStagingPath（6-356 ページ）
indexerSetCollectionValue（6-357 ページ）
IndexerStyleDir（6-358 ページ）
IndexerTouchMonitorInterval（6-359 ページ）
IndexerTraceCycleListChanges（6-360 ページ）
IndexerTraceLockFileChanges（6-361 ページ）
IndexerTraceStartStop（6-362 ページ）
IndexVaultFile（6-363 ページ）
IsAllowIndexRenditions（6-373 ページ）
IsAllowIndexVault（6-374 ページ）
MaxCollectionSize（6-515 ページ）
MaxDocIndexErrors（6-517 ページ）
MaxMergeBaseCount（6-522 ページ）
TimeoutPerOneMegInSec（6-812 ページ）
UseVdk4CustomStyle（6-883 ページ）
VerityAppName（6-901 ページ）
VerityAppSignature（6-902 ページ）
VerityEncoding（6-903 ページ）
VerityInstallDir（6-904 ページ）
VerityLocale（6-905 ページ）

その他の変数

IsFailedIndex（6-395 ページ）
IsFullTextIndexed（6-405 ページ）
VDKSUMMARY（6-899 ページ）

JSP

次の Idoc スクリプト変数は、JSP サポートに関連があります。

構成変数

IsJspServerEnabled（6-414 ページ）
JspAdminQuery（6-466 ページ）
JspDefaultIndexPage（6-467 ページ）
JspEnabledGroups（6-468 ページ）

ローカライゼーション

次の Idoc スクリプトの変数および関数は、ローカライゼーションに関連があります。

構成変数

CharMap（6-68 ページ）
DefaultFilterInputFormat（6-133 ページ）
DefaultFilterOutputFormat（6-134 ページ）
FileEncoding（6-232 ページ）
IdcLocale（6-338 ページ）
lcCaption（6-474 ページ）
LocaleArchiverDateFormat（6-504 ページ）
rptDisplayMapValue（6-659 ページ）
SystemDateFormat（6-788 ページ）
SystemLocale（6-789 ページ）
SystemTimeZone（6-794 ページ）
UserLanguageID（6-863 ページ）
UserLocaleId（6-864 ページ）
VerityEncoding（6-903 ページ）
VerityLocale（6-905 ページ）
WarnAboutTimeZone（6-906 ページ）

グローバル関数

lc（6-472 ページ）

ページ表示

次の Idoc スクリプトの変数および関数は、コンテンツ・サーバー Web ページの表示に関連があります。

グローバル関数

abortToErrorPage（6-2 ページ）
docLoadResourceIncludes（6-172 ページ）
pneNavigation（6-585 ページ）
UseBellevueLook（6-835 ページ）
UseXpedioLook（6-892 ページ）

その他の変数

AfterLogin（6-21 ページ）
coreContentOnly（6-109 ページ）
generateUniqueId（6-259 ページ）
getHelpPage（6-268 ページ）
getOptionListSize（6-269 ページ）
isCheckin（6-379 ページ）
htmlRefreshTimeout（6-297 ページ）
htmlRefreshUrl（6-298 ページ）
IsCurrentNav（6-385 ページ）
isDocPage（6-386 ページ）
IsDynamic（6-388 ページ）
isEditMode（6-390 ページ）
isFormSubmit（6-403 ページ）
isInfo（6-407 ページ）
IsMaxRows（6-419 ページ）
IsMultiPage（6-421 ページ）
isNew（6-422 ページ）
isQuery（6-431 ページ）
IsSavedQuery（6-436 ページ）
isUpdate（6-446 ページ）
isUploadFieldScript（6-447 ページ）
localPageType（6-506 ページ）
noInfoColumn（6-543 ページ）
noMCPrefill（6-545 ページ）

noRevisionsColumn（6-546 ページ）
PageParent（6-575 ページ）
ResultsTitle（6-651 ページ）
showDocActions（6-742 ページ）
showSearchResultsCheckoutColumn（6-746 ページ）
StdPageWidth（6-758 ページ）
TemplateClass（6-796 ページ）
TemplateFilePath（6-798 ページ）
TemplateName（6-799 ページ）
TemplateType（6-800 ページ）

関連項目

– フィールド表示（5-24 ページ）

パーソナライズ

次の Idoc スクリプト関数は、ユーザー・トピック（パーソナライズ）ファイルに関連があります。

グローバル関数

utGetValue（6-893 ページ）
utLoad（6-894 ページ）
utLoadResultSet（6-896 ページ）

関連項目

– パーソナライズ関数（3-10 ページ）
– ユーザー（5-40 ページ）

POP-UP CALENDAR

次の構成変数は、Pop-Up Calendar で使用します。

pucDefaultTime（6-625 ページ）
pucEnableOnSearch（6-626 ページ）
pucHideTimeInDateFields（6-627 ページ）
pucUseMonthLocale（6-628 ページ）
pucUseWeekdayLocale（6-629 ページ）
pucXOffset（6-630 ページ）
pucYOffset（6-631 ページ）
pucZIndex（6-632 ページ）

RECORD MANAGER および RETENTION MANAGER

AllowRetentionPeriodWithoutCutoff（6-34 ページ）
HideVitalReview（6-292 ページ）
RecordsManagementDenyAuthorFreePassOnRMSecurity（6-635 ページ）
RecordsManagementNumberOverwriteOnDelete（6-636 ページ）
RmaAddDocWhereClauseForScreening（6-652 ページ）
RMAHideExternalFieldsFromCheckInUpdate（6-653 ページ）
RMAHideExternalFieldsFromSearchInfo（6-654 ページ）
RmaNotifyDispReviewerAndCatAuthor（6-655 ページ）
RmaNotifyReviewerAndAlternateReviewer（6-656 ページ）

RESULTSET

次の Idoc スクリプトの変数および関数は、ResultSet に関連があります。

グローバル関数

getValue（6-275 ページ）
loadDocMetaDefinition（6-496 ページ）
loadEnterpriseSearchCollections（6-499 ページ）
loadEnterpriseSearchResults（6-500 ページ）
loadSearchOperatorTables（6-502 ページ）
loadUserMetaDefinition（6-503 ページ）
rsAddFields（6-660 ページ）
rsAddRowCountColumn（6-661 ページ）
rsAppend（6-662 ページ）
rsAppendNewRow（6-663 ページ）
rsAppendRowvalues（6-664 ページ）
rsCopyFiltered（6-666 ページ）
rsCreateResultSet（6-667 ページ）
rsDeleteRow（6-668 ページ）
rsDocInfoRowAllowDisclosure（6-669 ページ）
rsExists（6-670 ページ）
rsFieldByIndex（6-671 ページ）
rsFieldExists（6-672 ページ）
rsFindRowPrimary（6-673 ページ）
rsFirst（6-674 ページ）
rsInsertNewRow（6-675 ページ）
rsIsRowPresent（6-676 ページ）
rsMakeFromList（6-677 ページ）

rsMakeFromString（6-680 ページ）
rsMerge（6-682 ページ）
rsMergeDelete（6-683 ページ）
rsMergeReplaceOnly（6-684 ページ）
rsNext（6-685 ページ）
rsNumFields（6-686 ページ）
rsNumRows（6-687 ページ）
rsRemove（6-688 ページ）
rsRename（6-689 ページ）
rsRenameField（6-690 ページ）
rsSetRow（6-691 ページ）
rsSort（6-692 ページ）
rsSortTree（6-693 ページ）
setContentType（6-731 ページ）
setValue（6-738 ページ）
utLoadResultSet（6-896 ページ）

その他の変数

CURRENT_ROW（6-115 ページ）

スキーマ

次の Idoc スクリプト関数は、スキーマに関連があります。

スキーマ公開設定

スキーマの各部分の関係に影響を及ぼす変更が発生すると、スキーマは再公開されます。これらの設定は、公開要素に関連があります。

EnableSchemaPublishing（6-200 ページ）
getFieldViewDisplayValue（6-265 ページ）
getParentValue（6-270 ページ）
getViewValue（6-279 ページ）
getViewValueResultSet（6-280 ページ）
SchemaAllowRecursiveFileMove（6-696 ページ）
SchemaBuildDigestFile（6-697 ページ）
SchemaPublishCheckInterval（6-698 ページ）
SchemaPublishInterval（6-699 ページ）
SchemaPublishMaximumInterval（6-700 ページ）
SchemaPublishMinimumInterval（6-701 ページ）
SchemaPublishRenameRetryInitialInterval（6-702 ページ）

SchemaPublishRenameRetryIterations（6-703 ページ）
SchemaPublishRetryInterval（6-704 ページ）
SchemaPublishRetryIntervalMax（6-705 ページ）
SchemaVerifyPublishing（6-706 ページ）
SchemaWin32RenameDelayInterval（6-707 ページ）

その他のスキーマ設定

clearSchemaData（6-73 ページ）
jsFilename（6-465 ページ）
loadSchemaData（6-501 ページ）

検索

次の Idoc スクリプトの変数および関数は、検索に関連があります。

構成変数

AllowForceDelete（6-27 ページ）
CachedResultRowCount（6-59 ページ）
EnableDocumentHighlight（6-193 ページ）
EnterpriseSearchAsDefault（6-203 ページ）
HasSiteStudio（6-288 ページ）
indexerSetCollectionValue（6-357 ページ）
IsAutoSearch（6-378 ページ）
isZoneField（6-458 ページ）
LkDisableOnRebuild（6-480 ページ）
LkDisableReferenceInfo（6-481 ページ）
LkDisallowConversionFormats（6-482 ページ）
LkIsSecureSearch（6-484 ページ）
LkReExtractOnRebuild（6-485 ページ）
LkRefreshBatchSize（6-486 ページ）
LkRefreshErrorsAllowed（6-489 ページ）
LkRefreshErrorPercent（6-488 ページ）
LkRefreshErrorThreshold（6-490 ページ）
MaxHitCount（6-519 ページ）
MaxResults（6-527 ページ）
MaxSavedSearchResults（6-528 ページ）
MaxSearchConnections（6-529 ページ）
noInfoColumn（6-543 ページ）
noRevisionsColumn（6-546 ページ）
NoSearchConnectionCloseOnDocChange（6-547 ページ）

SearchCacheCleanerInterval（6-711 ページ）
SearchCacheHardLimit（6-712 ページ）
SearchCacheMaxItemAge（6-713 ページ）
SearchCacheSoftLimit（6-714 ページ）
SearchCacheTrace（6-715 ページ）
SearchConnectionWaitTimeout（6-716 ページ）
SearchDebugLevel（6-717 ページ）
SearchDir（6-718 ページ）
SearchIndexerEngineName（6-719 ページ）
SearchQueryFormat（6-720 ページ）
SearchQueryTrace（6-721 ページ）
showDocActions（6-742 ページ）
showSearchResultsCheckoutColumn（6-746 ページ）
UniversalSearchAllowFullText（6-825 ページ）

グローバル関数

loadCollectionInfo（6-495 ページ）

その他の変数

IsCurrentNav（6-385 ページ）
IsFullTextIndexed（6-405 ページ）
IsLocalSearchCollectionID（6-416 ページ）
IsMultiPage（6-421 ページ）
IsSavedQuery（6-436 ページ）
NoMatches（6-544 ページ）
OneMatch（6-559 ページ）
QUERY_STRING（6-633 ページ）
regexMatches（6-638 ページ）
regexReplaceAll（6-639 ページ）
regexReplaceFirst（6-640 ページ）
ResultsTitle（6-651 ページ）
UseHtmlOrTextHighlightInfo（6-843 ページ）
UseSearchCache（6-876 ページ）
UseXmlUrl（6-891 ページ）

セキュリティ

この項では、セキュリティに関連する Idoc スクリプトの変数および関数のリストを示します。

- 内部セキュリティ（5-36 ページ）
- 外部セキュリティ（5-37 ページ）

内部セキュリティ

次の Idoc スクリプトの変数および関数は、コンテンツ・サーバー内部セキュリティに関連があります。

構成変数

AllowAllRedirectUrl（6-22 ページ）
AuthorDelete（6-48 ページ）
Default Accounts（6-130 ページ）
encodeHtml（6-201 ページ）
ExclusiveCheckout（6-208 ページ）
GetCopyAccess（6-261 ページ）
HtmlDataInputFilterLevel（6-295 ページ）
SelfRegisteredAccounts（6-725 ページ）
SelfRegisteredRoles（6-726 ページ）
ShowOnlyKnownAccounts（6-744 ページ）
SpecialAuthGroups（6-755 ページ）
UseAccounts（6-829 ページ）
UserFilteredSubjects（6-857 ページ）
UseSelfRegistration（6-879 ページ）

グローバル関数

docUrlAllowDisclosure（6-178 ページ）
hasAppRights（6-281 ページ）
isUserOverrideSet（6-450 ページ）
rsDocInfoRowAllowDisclosure（6-669 ページ）
stdSecurityCheck（6-759 ページ）
userHasAccessToAccount（6-859 ページ）
userHasGroupPrivilege（6-860 ページ）
userHasRole（6-861 ページ）

その他の変数

AdminAtLeastOneGroup（6-16 ページ）
AllowCheckin（6-24 ページ）
AllowCheckout（6-25 ページ）
DelimitedUserRoles（6-147 ページ）
EmptyAccountCheckinAllowed（6-191 ページ）
HasPredefinedAccounts（6-287 ページ）
IsCheckinPreAuthed（6-380 ページ）
IsLoggedIn（6-417 ページ）
IsSubAdmin（6-440 ページ）
IsSysManager（6-442 ページ）
UserAccounts（6-849 ページ）
UserAppRights（6-851 ページ）
UserDefaultAccount（6-854 ページ）
UserIsAdmin（6-862 ページ）
UserRoles（6-867 ページ）

外部セキュリティ

次の Idoc スクリプト変数は、外部セキュリティに関連があります（Web サーバーおよび NTLM セキュリティ）。

構成変数

AccountMapPrefix（6-4 ページ）
AcctPermDelim（6-5 ページ）
AcctPrefix（6-7 ページ）
AdsSimpleAuth（6-18 ページ）
AdsUserName（6-19 ページ）
AdsUserPassword（6-20 ページ）
AttributeMap（6-46 ページ）
DefaultAuth（6-131 ページ）
DefaultMasterDomain（6-138 ページ）
DefaultNetworkAccounts（6-140 ページ）
DomainControllerName（6-180 ページ）
HasExternalUsers（6-282 ページ）
HasGlobalUsers（6-283 ページ）
IntradocRealm（6-369 ページ）
IsIntranetAuthOnly（6-409 ページ）
IsPrimaryUserBase（6-428 ページ）
LdapAdminDN（6-475 ページ）

LdapAdminPassword（6-476 ページ）
LdapPort（6-477 ページ）
LdapServer（6-478 ページ）
LdapSuffix（6-479 ページ）
LocalGroupServer（6-505 ページ）
NetworkAdminGroup（6-540 ページ）
NtlmSecurityEnabled（6-555 ページ）
Priority（6-596 ページ）
ProviderClass（6-618 ページ）
ProviderConfig（6-619 ページ）
ProviderConnection（6-620 ページ）
ProviderType（6-622 ページ）
RedirectHostsFilter（6-637 ページ）
RolePrefix（6-657 ページ）
SocketHostAddressSecurityFilter（6-751 ページ）
SocketHostNameSecurityFilter（6-752 ページ）
SourcePath（6-754 ページ）
UseAdsi（6-831 ページ）
UseFullGroupName（6-841 ページ）
UseGroupFilter（6-842 ページ）
UseLocalGroups（6-845 ページ）
UseNetscape（6-847 ページ）
UseNtlm（6-848 ページ）
UseSecureLdap（6-878 ページ）
UseShortNamesAlways（6-880 ページ）
UseSSL（6-881 ページ）
WebServerAuthOnly（6-911 ページ）

その他の変数

AllowIntranetUsers（6-28 ページ）
ExternalUserAccounts（6-214 ページ）
ExternalUserRoles（6-215 ページ）
IsPromptingForLogin（6-429 ページ）
IsUploadSockets（6-448 ページ）

文字列

次の Idoc スクリプトの変数および関数は、文字列に関連があります。

構成変数

StrConfineOverflowChars（6-765 ページ）

グローバル関数

js（6-464 ページ）
strCenterPad（6-760 ページ）
strCommaAppendNoDuplicates（6-762 ページ）
strConfine（6-763 ページ）
StrConfineOverflowChars（6-765 ページ）
strEquals（6-766 ページ）
strEqualsIgnoreCase（6-767 ページ）
strGenerateRandom（6-768 ページ）
strIndexOf（6-769 ページ）
strLeftFill（6-770 ページ）
strLeftPad（6-771 ページ）
strLength（6-772 ページ）
strLower（6-773 ページ）
strRemoveWs（6-774 ページ）
strReplace（6-775 ページ）
strReplaceIgnoreCase（6-776 ページ）
strRightFill（6-777 ページ）
strRightPad（6-778 ページ）
strSubstring（6-779 ページ）
strTrimWs（6-781 ページ）
strUpper（6-782 ページ）
toInteger（6-813 ページ）
url（6-827 ページ）
urlEscape7Bit（6-828 ページ）
xml（6-945 ページ）

テンプレート

次の Idoc スクリプト変数は、コンテンツ・サーバーのテンプレートに関連があります。

構成変数

SystemReportsDir（6-791 ページ）
SystemTemplatesDir（6-793 ページ）

その他の変数

TemplateClass（6-796 ページ）
TemplateFilePath（6-798 ページ）
TemplateName（6-799 ページ）
TemplateType（6-800 ページ）

ユーザー

次の Idoc スクリプトの変数および関数は、コンテンツ・サーバー・ユーザーに関連があります。

構成変数

Default Accounts（6-130 ページ）
ExclusiveCheckout（6-208 ページ）
SysAdminAddress（6-787 ページ）
UserCacheTimeout（6-853 ページ）
UseSelfRegistration（6-879 ページ）

グローバル関数

getUserValue（6-274 ページ）
getValueForSpecifiedUser（6-278 ページ）
isUserOverrideSet（6-450 ページ）
userHasAccessToAccount（6-859 ページ）
userHasGroupPrivilege（6-860 ページ）
userHasRole（6-861 ページ）

その他の変数

AdminAtLeastOneGroup（6-16 ページ）
AllowCheckin（6-24 ページ）
AllowCheckout（6-25 ページ）
AuthorAddress（6-47 ページ）
DelimitedUserRoles（6-147 ページ）
ExternalUserAccounts（6-214 ページ）
ExternalUserRoles（6-215 ページ）
HasPredefinedAccounts（6-287 ページ）
IsCheckinPreAuthed（6-380 ページ）
IsExternalUser（6-393 ページ）
IsLoggedIn（6-417 ページ）
IsSubAdmin（6-440 ページ）
IsSysManager（6-442 ページ）
IsUserEmailPresent（6-449 ページ）
UserAccounts（6-849 ページ）
UserAddress（6-850 ページ）
UserAppRights（6-851 ページ）
UserDefaultAccount（6-854 ページ）
UserFullName（6-858 ページ）
UserIsAdmin（6-862 ページ）
UserName（6-865 ページ）
UserRoles（6-867 ページ）
wfAddUser（6-915 ページ）

関連項目

– パーソナライズ（5-31 ページ）

WEB サーバー

次の Idoc スクリプト変数は、Web サーバーおよび Web サーバー・フィルタ・プラグインに関連があります。

構成変数

CGI_DEBUG（6-64 ページ）
CGI_RECEIVE_DUMP（6-65 ページ）
CGI_SEND_DUMP（6-66 ページ）
CgiFileName（6-67 ページ）
DefaultAuth（6-131 ページ）
DefaultMasterDomain（6-138 ページ）
DomainControllerName（6-180 ページ）
FILTER_DEBUG（6-234 ページ）
HttpAbsoluteWebRoot（6-309 ページ）
HttpRelativeWebRoot（6-321 ページ）
HttpServerAddress（6-322 ページ）
IdcAuthExtraRequestParams（6-331 ページ）
IdcHttpHeaderVariables（6-336 ページ）
IntradocRealm（6-369 ページ）
IsIntranetAuthOnly（6-409 ページ）
PATH_INFO（6-581 ページ）
PLUGIN_DEBUG（6-584 ページ）
ProfileCacheTimeoutInMins（6-617 ページ）
RedirectHostsFilter（6-637 ページ）
SocketHostAddressSecurityFilter（6-751 ページ）
SocketHostNameSecurityFilter（6-752 ページ）
UseSSL（6-881 ページ）
WebServerAuthOnly（6-911 ページ）

その他の変数

CONTENT_LENGTH（6-104 ページ）
GATEWAY_INTERFACE（6-258 ページ）
getCookie（6-260 ページ）
HTTP_ACCEPT（6-299 ページ）
HTTP_ACCEPT_ENCODING（6-300 ページ）
HTTP_ACCEPT_LANGUAGE（6-301 ページ）
HTTP_COOKIE（6-302 ページ）
HTTP_HOST（6-303 ページ）
HTTP_INTERNETUSER（6-304 ページ）
HTTP_REFERER（6-305 ページ）
HTTP_USER_AGENT（6-306 ページ）
HttpIgnoreWebServerInternalPortNumber（6-316 ページ）
PATH_TRANSLATED（6-582 ページ）
QUERY_STRING（6-633 ページ）
REMOTE_ADDR（6-641 ページ）
REMOTE_HOST（6-642 ページ）
REQUEST_METHOD（6-643 ページ）
SCRIPT_NAME（6-708 ページ）
SERVER_NAME（6-727 ページ）
SERVER_PORT（6-728 ページ）
SERVER_PROTOCOL（6-729 ページ）
SERVER_SOFTWARE（6-730 ページ）
setCookie（6-732 ページ）

ワークフロー

次の Idoc スクリプトの変数および関数は、ワークフローに関連があります。

構成変数

isRepromptLogin（6-433 ページ）
IsSavedWfCompanionFile（6-437 ページ）
PrimaryWorkQueueTimeout（6-594 ページ）
WorkflowDir（6-943 ページ）
WorkflowIntervalHours（6-944 ページ）

グローバル関数

getValueForSpecifiedUser（6-278 ページ）

ワークフロー関数

wfAddActionHistoryEvent（6-913 ページ）
wfAddUser（6-915 ページ）
wfComputeStepUserList（6-916 ページ）
wfCurrentGet（6-917 ページ）
wfCurrentSet（6-918 ページ）
wfCurrentStep（6-919 ページ）
wfDisplayCondition（6-920 ページ）
wfExit（6-921 ページ）
wfGet（6-922 ページ）
wfGetStepTypeLabel（6-923 ページ）
wfIsFinishedDocConversion（6-925 ページ）
wfIsNotifyingUsers（6-926 ページ）
wfIsReleasable（6-927 ページ）
wfLoadDesign（6-933 ページ）
wfNotify（6-936 ページ）
wfReleaseDocument（6-938 ページ）
wfSet（6-939 ページ）
wfSetIsNotifyingUsers（6-940 ページ）
wfUpdateMetaData（6-942 ページ）

その他の変数

AllowReview（6-35 ページ）
dWfName（6-188 ページ）
dWfStepName（6-189 ページ）
entryCount（6-204 ページ）
IsEditRev（6-391 ページ）
IsWorkflow（6-455 ページ）
lastEntryTs（6-470 ページ）
SingleGroup（6-748 ページ）
wfAction（6-912 ページ）
wfAdditionalExitCondition（6-914 ページ）
wfJumpEntryNotifyOff（6-928 ページ）
wfJumpMessage（6-929 ページ）
wfJumpName（6-930 ページ）
wfJumpReturnStep（6-931 ページ）
wfJumpTargetStep（6-932 ページ）
wfMailSubject（6-934 ページ）
wfMessage（6-935 ページ）
wfParentList（6-937 ページ）
WfStart（6-941 ページ）

6

Idoc スクリプト索引

この項では、Idoc スクリプトの変数および関数をアルファベット順にリストし、各項目の詳細を示します。

ABORTTOERRORPAGE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ページ表示（5-30 ページ）

説明

現在のページを中断し、エラー・メッセージを表示します。

- この関数は StatusCode（6-756 ページ）変数を評価し、負の数値（-1）が返された場合、現在のページの表示が中断されてエラー・ページに置き換えられます。
- StatusMessage（6-757 ページ）変数をエラー・メッセージの文字列として使用できます。

パラメータ

1 つのパラメータを取ります。

- パラメータは、エラー・メッセージの文字列のみです。

出力

エラー・ページのエラー・メッセージ文字列を返します。

例

現在のページを中断し、Access Denied をエラー・メッセージとして表示する例を次に示します。

```
<$abortToErrorPage("Access Denied")$>
```

現在のページを中断し、*StatusMessage* 変数の値をエラー・メッセージとして表示します。

```
<$abortToErrorPage("<$StatusMessage$>")$>
```

関連項目

- executeService（6-211 ページ）
- IsRequestError（6-434 ページ）
- StatusCode（6-756 ページ）
- StatusMessage（6-757 ページ）

ACADUSELISPINTERFACE

タイプおよび使用方法

❖ 構成変数（4-3 ページ）

❖ AutoCAD（5-14 ページ）

説明

LIST インタフェースを使用するように AutoCAD を設定します。これは AutoCAD の不具合を回避します。

❖ TRUE に設定すると、LISP インタフェースが使用されます。変換する前に、*acadunsupp.arx* ファイルを AutoCAD 14.01 に登録しておく必要があります。

❖ デフォルトは、TRUE です。

場所

<*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
AcadUseLISPInterface=true
```

関連項目

– UseAutoCad2000（6-833 ページ）

ACCOUNTMAPPREFIX

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

コンテンツ・サーバー・アカウントにマップする Windows ネットワーク・グループを識別する接頭辞を定義します。

- NTLM セキュリティで使用します。この設定値で開始する Windows グループは、コンテンツ・サーバー・アカウントにマップされます。
- デフォルトは、@ です。

場所

- 「Configure NT Domain Security」ページ→「Account Map Prefix」
- <*install_dir*>/data/users/config/filter.hda

例

次は構成エントリの使用例です。

```
AccountMapPrefix=+
```

ACCTPERMDELIM

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

Active Directory または LDAP グループ名のアカウント名とアカウント権限を区切る文字列を定義します。

- Active Directory または LDAP グループ名がアカウントにマップされ、この部分文字列が含まれている場合、この部分文字列の左側の文字列がアカウント名で、右側の文字列がアカウント権限となります。
- たとえば、デリミタを +（プラス記号）として定義すると、グループ名 *Acct1+rw* は、読取りおよび書込み権限を持つ *Acct1* という名前のアカウントにマップされます。デリミタを _（アンダースコア）として定義すると、グループ名 *Acct1+rw* は、DefaultNetworkAccounts（6-140 ページ）設定により定義される権限を持つ *Acct1+rw* という名前のアカウントにマップされます。
- デフォルトは、_（アンダースコア）です。

場所

Active Directory セキュリティ

- 「Active Directory Configuration」ページ→「Account Permissions Delimiter」
- <*install_dir*>/data/users/config/filter.hda

LDAP セキュリティ

- 「Add/Edit LDAP Provider」ページ→「Account Permissions Delimiter」
- <*install_dir*>/data/providers/*provider_name*/provider.hda

例

次は構成エントリの使用例です。

```
AcctPermDelim=+
```

関連項目

– DefaultNetworkAccounts（6-140 ページ）

ACCTPREFIX

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

コンテンツ・サーバー・アカウントにマップされる Active Directory または LDAP の接頭辞を定義します。

- これはセミコロンで区切られた LDAP スタイル文字列のリストで、これらの文字列は、Active Directory または LDAP グループ名の部分文字列です。いずれかの文字列がグループ名の部分文字列である場合、指定された深さに従って、グループ名の残りの部分がコンテンツ・サーバー・アカウントにマップされます。詳細な例は、「Active Directory Configuration」ページまたは「Add/Edit LDAP Provider」ページを参照してください。
- アカウントを Active Directory または LDAP グループからマップするには、UseGroupFilter（6-842 ページ）設定が TRUE である必要があります。
- デフォルトは、空の文字列です。

場所

Active Directory セキュリティ

- 「Active Directory Configuration」ページ→「Account Prefix」
- <*install_dir*>/data/users/config/filter.hda

LDAP セキュリティ

- 「Add/Edit LDAP Provider」ページ→「Account Prefix」
- <*install_dir*>/data/providers/*provider_name*/provider.hda

例

次は構成エントリの使用例です。

```
AcctPrefix=OU=Accounts,OU=Stellent[1];
```

関連項目

– UseGroupFilter（6-842 ページ）
– RolePrefix（6-657 ページ）

ACTIONITEMINTERVAL

タイプおよび使用方法

- Collaboration Manager の構成変数（4-4 ページ）
- Collaboration Manager（5-5 ページ）

説明

ブラウザが Content Server からのレスポンスをチェックするまでの待機時間をミリ秒単位で指定します。

- この設定により、プログレス・バーの更新間隔も設定されます。
- デフォルトは、100 です。
- Collaboration Manager のインストール中に追加されます。

場所

<*install_dir*>/<*instancename*>/data/components/CollaborationManager/config.cfg

例

```
actionItemInterval=200
```

関連項目

- deleteActionItemInterval（6-145 ページ）
- deleteWatchdogInterval（6-146 ページ）
- watchdogInterval（6-907 ページ）

ADDEMPTYOPTION

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドのオプション・リストで、空白値をリストの最初のオプションとして指定します。

出力

- オプションの最初の値が空白の場合は、TRUE を返します。
- オプションの最初の値が空白でない場合は、FALSE を返します。

例

オプション・リストの最初の値が空白であることを指定する例を次に示します。

```
<$endif$>
```

関連項目

– fieldIsOptionList（6-224 ページ）

ADDITIONALINDEXBUILDPARAMS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）

説明

すべての Verity Indexer（*mkvdk.exe*）の実行にビルド・パラメータを追加します。Verity Integration は、Content Server とともに使用できるアドオン製品であり、購入することができます。

- 使用すると、数回のバルクロードではなく、索引付けのすべてのバルクロードの後に最適化を強制できます。たとえば、コレクションが完全にマージされていないと、コレクションが多数のファイル間に広がり、検索が非効率的になり、多数のファイル・ハンドルが使用されます。
- デフォルトは、空の文字列です。

パラメータ

- 各ビルド・パラメータは、接頭辞としてハイフンが付きます。たとえば、`-repair`。
- ビルド・パラメータが独自のサブパラメータを持つ場合、最初のサブパラメータにはハイフンを付けません。たとえば、`-optimize maxmerge`。
- ビルド・パラメータが複数のサブパラメータを持つ場合は、最初のサブパラメータの後、それに続く各サブパラメータの先頭にハイフンを付けます。たとえば、`-optimize maxmerge -squeeze`。

パラメータ	説明
-repair	コレクションを修復します。
-optimize	次のサブパラメータを 1 つ以上使用して、コレクションを最適化します。 **maxmerge:** 個別のパーティションを 1 つのパーティションにマージします。 **squeeze:** 削除されたドキュメントの領域をリカバリします。 **vdbopt:** コレクションの小さいユニットを構成して最適化された VDB（Verity データベース）を構築します。 **spanword:** すべてのパーティションにわたって単語リストを作成します。 **ngramindex:** 全域にわたる単語リストに *ngram* 索引を作成します。 **maxclean:** 古いコレクション・ファイルを削除します。このパラメータは、コレクションがオフライン（または読取り専用）の場合にのみ使用できます。 **readonly:** コレクションを読取り専用にします。 **tuneup:** 読取り / 書込み用としてコレクションを完全に最適化します。 **publish:** 読取り専用としてコレクションを完全に最適化します。
-servlev	サービス・レベルを上書きしてコレクションの機能を有効にします。次の 1 つ以上のパラメータを使用します。 **search:** 検索および取得を有効にします。 **optimize:** コレクションの最適化を有効にします。 **assist:** 単語リストの作成を有効にします。 **housekeep:** 不要なファイルのハウスキーピングを有効にします。 **insert:** ドキュメントの追加および更新を有効にします。 **delete:** ドキュメントの削除を有効にします。 **backup:** バックアップを有効にします。 **purge:** バックグラウンドでのパージを有効にします。 **repair:** コレクションの修復を有効にします。 **index:** *insert -delete* と同じです。 **dataprep:** *search -optimize -assist -housekeep -index* と同じです。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
AdditionalIndexBuildParams=-repair

AdditionalIndexBuildParams=-optimize maxclean

AdditionalIndexBuildParams=-optimize maxmerge -squeeze
```

関連項目

– DoAutoMaxMerge（6-170 ページ）

ADDITIONALSUBSCRIPTIONCOLLATEDFIELDLIST

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

サブスクリプションに表示される追加のフィールドのリストを定義します。

- デフォルトのフィールドは、SubscriptionCollatedFieldList に定義されています。

場所

<*install_dir*>/config/config.cfg

例

```
AdditionalSubscriptionCollatedFieldList=dOriginalName
```

関連項目

– SubscriptionCollatedFieldList（6-785 ページ）

ADJUSTPRINTERMARGINS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

Microsoft PowerPoint ファイルの変換時に、必要に応じてプリンタの余白を調整します。

- TRUE に設定すると、Microsoft PowerPoint ファイルの変換時に、必要に応じてプリンタの余白が調整されます。
- FALSE に設定すると、Microsoft PowerPoint ファイルの変換時に、プリンタの余白は調整されません。
- デフォルトは、TRUE です。

場所

<*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
AdjustPrinterMargins=false
```

ADMINATLEASTONEGROUP

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- 内部セキュリティ（5-36 ページ）
- ユーザー（5-40 ページ）

説明

現在のユーザーが、1 つ以上のセキュリティ・グループに対する admin ロールを持つかどうかをチェックします。

出力

- ユーザーが 1 つ以上のセキュリティ・グループの管理者である場合、TRUE を返します。
- admin ロールを割り当てられたユーザーでない場合は、FALSE を返します。

例

管理者向けのオプションのプレゼンテーションを実行するために使用できます。

```
<$if (AdminAtLeastOneGroup)$>
    <a href="<$redirect$>">
<$endif$>
```

関連項目

- UserAppRights（6-851 ページ）
- UserIsAdmin（6-862 ページ）

ADMINDATADIR

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ディレクトリおよびパス（5-22 ページ）
- クラスタリング（5-5 ページ）

説明

管理サーバーの data ディレクトリのパスを定義します。

- デフォルトは、<*install_dir*>/admin/data/ です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
AdminDataDir=c:/stellent/admin/data/
```

ADSSIMPLEAUTH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

コンテンツ・サーバーのロールおよびアカウントが、Active Directory グループから導出されるかどうかを指定します。

- TRUE に設定すると、ユーザーのパスワードのみが Active Directory から認証されます。ロールおよびアカウントは、別のユーザー・ベース（LDAP サーバーなど）から導出されます。
- FALSE に設定すると、ロールおよびアカウントは、Active Directory グループから導出されます。
- デフォルトは、FALSE です。

場所

<*install_dir*>/data/users/config/filter.hda

例

次は構成エントリの使用例です。

```
AdsSimpleAuth=true
```

ADSUSERNAME

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

Active Directory に対する読取り権限を持つユーザー名を指定します。

- この設定は、Web サーバーが、Active Directory に対する読取り権限のない Windows サーバーで実行されている場合にのみ必要です。
- ユーザー名は、*DOMAIN_NAME¥user_name* の形式にする必要があります。
- この設定および AdsUserPassword（6-20 ページ）設定が指定されていない場合、デフォルトは Web サーバーを実行するユーザー名です。

場所

- 「Active Directory Configuration」ページ→「User Name」
- <*install_dir*>/data/users/config/filter.hda

例

次は構成エントリの使用例です。

```
AdsUserName=CORP_DOMAIN¥admin_user
```

関連項目

– AdsUserPassword（6-20 ページ）

ADSUSERPASSWORD

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

AdsUserName（6-19 ページ）設定によって指定された、Active Directory に対する読取り権限を持つユーザー名のパスワードを指定します。

- この設定は、Web サーバーが、Active Directory に対する読取り権限のない Windows サーバーで実行されている場合にのみ必要です。
- この設定および AdsUserName（6-19 ページ）設定が指定されていない場合、デフォルトは Web サーバーを実行するユーザー名です。

場所

- 「Active Directory Configuration」ページ→「User Password」
- <*install_dir*>/data/users/config/filter.hda

例

次は構成エントリの使用例です。

```
AdsUserPassword=idc
```

関連項目

– AdsUserName（6-19 ページ）

AFTERLOGIN

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- ページ表示（5-30 ページ）

説明

現在のページが、ログインの直後に作成されたかどうかを指定します。

出力

- 現在のページがログインの直後に作成された場合は、TRUE を返します。
- 現在のページがログインの直後に作成されなかった場合は、FALSE を返します。

例

ログインの直後にページが作成されなかった場合、代替 URL を表示する例を次に示します。

```
<$if not AfterLogin$>
    <a href="<$redirect$>">
<$endif$>
```

AllowAllRedirectUrl

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

RedirectHostsFilter によって定義された機能をリセットし、その変数が設定される前と同様に Content Server を動作させます。この変数を TRUE に設定すると、RedirectHostsFilter によって指定された URL ではなく、任意の URL にユーザーをリダイレクトできます。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
AllowAllRedirectUrl=true
```

関連項目

– RedirectHostsFilter（6-637 ページ）

AllowAlternateMetaFile

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

ユーザーがメタデータ専用のコンテンツを代替ファイルとして発行できるようにします。

- メタデータ専用ファイルの作成機能を有効にします。この場合、コントリビュータは代替ファイルを指定しません。かわりに、テンプレートおよびコンテンツ・アイテムのメタデータを使用して、システムによって代替ファイルが作成されます。
- config.cfg ファイルで TRUE に設定した場合、チェックイン・ページにチェック・ボックスが追加されます。このチェック・ボックスを選択すると、createAlternateMetaFile（6-110 ページ）設定が有効になり、代替ファイルが実際に作成されます。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
AllowAlternateMetaFile=true
```

関連項目

- createAlternateMetaFile（6-110 ページ）
- AllowPrimaryMetaFile（6-33 ページ）

ALLOWCHECKIN

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- 内部セキュリティ（5-36 ページ）
- ユーザー（5-40 ページ）

説明

現在のユーザーが、コンテンツ・アイテムのセキュリティ・グループに対するチェックイン権限のあるユーザーかどうかをチェックします。

出力

- チェックイン権限を持つユーザーの場合は、TRUE を返します。
- チェックイン権限のないユーザーの場合は、FALSE を返します。

例

チェックイン権限を持つユーザー向けのオプションのプレゼンテーションを実行するために使用できます。

```
<$if (AllowCheckin)$>
    <a href="<$redirect$>">
<$endif$>
```

ALLOWCHECKOUT

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- 内部セキュリティ（5-36 ページ）
- ユーザー（5-40 ページ）

説明

現在のユーザーが、コンテンツ・アイテムのセキュリティ・グループに対するチェックアウト権限のあるユーザーかどうかをチェックします。

出力

- チェックアウト権限を持つユーザーの場合は、TRUE を返します。
- チェックアウト権限のないユーザーの場合は、FALSE を返します。

例

チェックアウト権限を持つユーザー向けのオプションのプレゼンテーションを実行するために使用できます。

```
<$if (AllowCheckout)$>
    <a href="<$redirect$>">
<$endif$>
```

ALLOWCONCURRENTUPDATE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）

説明

索引の再構築中に、既存コレクションへの新規ドキュメントの追加を許可します。

場所

<*install_dir*>/bin/intradoc.cfg

例

次は構成エントリの使用例です。

```
AllowConcurrentUpdate=TRUE
```

ALLOWFORCEDELETE

タイプおよび使用方法

- Link Manager の構成変数（4-19 ページ）
- 検索（5-34 ページ）

説明

他のコンテンツ・アイテムが参照または使用しているコンテンツ・アイテムを削除するために使用します。

デフォルトの設定は、`AllowForceDelete=FALSE` です。この構成では、Content Server は、リンクとして参照されているコンテンツ・アイテムの削除リクエストを拒否します。

`AllowForceDelete` を TRUE に設定すると、リンクとして参照されているコンテンツ・アイテムを削除できます。これにより、リンクは無効になります。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
AllowForceDelete=TRUE
```

関連項目

– Link Manager の使用の詳細は、『リポジトリ・コンテンツの管理』を参照してください。

ALLOWINTRANETUSERS

タイプおよび使用方法

- 動的変数（4-20 ページ）
- 外部セキュリティ（5-37 ページ）

説明

Microsoft ネットワーク（NTLM）認証が有効かどうかをチェックします。

出力

- NtlmSecurityEnabled（6-555 ページ）が TRUE に設定されている場合、TRUE を返します。
- NtlmSecurityEnabled（6-555 ページ）が FALSE に設定されている場合、FALSE を返します。

例

Microsoft ネットワークのセキュリティを評価します。

```
<$AllowIntranetUsers$>
```

関連項目

– NtlmSecurityEnabled（6-555 ページ）

ALLOWMATCHESINDATECHECK

タイプおよび使用方法

- 構成変数（4-3 ページ）
- アーカイバ（5-3 ページ）

説明

TRUE に設定すると、前のリビジョンの日付に一致する日付が有効となるように日付検証コードが変更されます。

- この変数を設定すると、アーカイブのインポートおよびバッチ・ロード・プロセスで、同じ作成日またはリリース日を持つコンテンツ・アイテムのリビジョンをインポートできます。
- 通常、アーカイバの標準の日付検証は、リビジョンの日付がドキュメントの前のリビジョンと一致するか、1 分以内である場合、そのリビジョンのインポートを許可しません。

場所

<*install_dir*>/config/config.cfg

例

```
AllowMatchesInDateCheck=TRUE
```

ALLOWMIXEDCASEACL

タイプおよび使用方法

- Collaboration Manager の構成変数（4-4 ページ）
- Collaboration Manager（5-5 ページ）

説明

データベース問合せで大 / 小文字が混在するエントリを許可します（または、エントリが小文字になります）。

- Collaboration Manager のインストール中に追加されます。
- TRUE に設定すると、データベース問合せで大文字と小文字の両方を使用できます。
- デフォルトは、TRUE です。

場所

<*install_dir*>/<*instancename*>/data/components/CollaborationManager/config.cfg

例

次は構成エントリの使用例です。

```
AllowMixedCaseACL=true
```

ALLOWMRSIDRETRYCOMPRESSIONLEVEL

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Compression（5-18 ページ）

説明

Compression アドオン・モジュールの再試行機能を有効にします。

- TRUE に設定すると、特定のファイルでズーム・レベルの設定が高すぎる場合、2 番目に低いズーム・レベルで圧縮を試みます。MrSidLowestCompressionLevel（6-536 ページ）設定で指定されたレベルでファイルを圧縮できない場合は、ネイティブ・ファイルを JPG ファイルに変換します。
- FALSE に設定すると、ズーム・レベルの設定が高すぎる場合、圧縮は 1 回目の試行で失敗します。
- デフォルトは、FALSE です。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
AllowMrSidRetryCompressionLevel=true
```

関連項目

– MrSidLowestCompressionLevel（6-536 ページ）

ALLOWPASSTHRU

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Inbound Refinery（5-11 ページ）

説明

変換に失敗したファイルの処理を設定します。

- TRUE に設定すると、変換が失敗またはタイムアウトした場合、ファイルをネイティブな形式でパススルーします。
- FALSE に設定すると、変換が失敗またはタイムアウトした場合、ファイルをパススルーしません。エラー・メッセージが生成され、ファイルは GENWWW ステータスで維持されます。
- デフォルトは、空の文字列です。

場所

<*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
AllowPassthru=false
```

ALLOWPRIMARYMETAFILE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

ユーザーがメタデータ専用のコンテンツをプライマリ・ファイルとして発行できるようにします。

- メタデータ専用ファイルの作成機能を有効にします。この場合、コントリビュータはプライマリ・ファイルを指定しません。かわりに、テンプレートおよびコンテンツ・アイテムのメタデータを使用して、システムによってプライマリ・ファイルが作成されます。
- config.cfg ファイルで TRUE に設定した場合、チェックイン・ページにチェック・ボックスが追加されます。このチェック・ボックスを選択すると、createPrimaryMetaFile（6-112 ページ）設定が有効になり、プライマリ・ファイルが実際に作成されます。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
AllowPrimaryMetaFile=true
```

関連項目

- AllowAlternateMetaFile（6-23 ページ）
- createPrimaryMetaFile（6-112 ページ）

ALLOWRETENTIONPERIODWITHOUTCUTOFF

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Record Manager および Retention Manager（5-32 ページ）

説明

非レコード・コンテンツ・アイテムの保存期間を指定します。TRUE に設定すると、非レコード・コンテンツ・アイテムのトリガーの保存期間が有効化されます。FALSE に設定すると、保存期間が無効化されます。

デフォルトは、TRUE です。

場所

- <*install_dir*>/custom/RecordsManagement/records_management_environments.cfg
- <*install_dir*>/custom/RetentionManager/retentionmanager_environments.cfg

例

次は構成エントリの使用例です。

```
AllowRetentionPeriodWithoutCutoff=false
```

ALLOWREVIEW

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- ワークフロー（5-44 ページ）

説明

現在のユーザーが、現在のワークフロー・アイテムの承認または却下を許可されているかどうかをチェックします。

出力

- ユーザーが現在のワークフロー・ステップのレビューアである場合は、TRUE を返します。
- ユーザーが現在のワークフロー・ステップのレビューアでない場合は、FALSE を返します。

例

ユーザーがレビューアである場合、「Approve」ボタンと「Reject」ボタンを表示する例を次に示します。

```
<$if AllowReview$>
    <$include workflow_doc_action_buttons$>
<$endif$>
```

ALLOWSAMEPRIMARYALTERNATEEXTENSIONS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

プライマリ・ファイルと代替ファイルが同じファイル拡張子を持つことができます。

- TRUE に設定すると、プライマリ・ファイルと代替ファイルが同じファイル拡張子を持つことができます。
- FALSE に設定すると、プライマリ・ファイルと代替ファイルは、同じファイル拡張子を持つことができません。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
AllowSamePrimaryAlternateExtensions=true
```

ALLOWSECONDARYCONVERSION

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Inbound Refinery（5-11 ページ）

説明

ネイティブ変換ステップが失敗した場合、Inbound Refinery でバックアップ変換ステップを使用するかどうかを定義します。

- TRUE に設定すると、変換に失敗した場合、バックアップ変換ステップが使用されます。
- FALSE に設定すると、変換に失敗した場合、バックアップ変換ステップは使用されません。
- デフォルトは、TRUE です。

場所

- 「Local Inbound Refinery Configuration」画面→「General」タブ→「Allow secondary conversion step」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
AllowSecondaryConversion=false
```

関連項目

– SecondaryConversionStepName（6-722 ページ）

ALLOWSKIPPEDHYPERLINKTOCAUSEINCOMPLETE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

ハイパーリンクを PDF に変換できない場合、変換は「incomplete」としてマークされます。

- TRUE に設定すると、ハイパーリンクを変換できない場合、変換は「incomplete」としてマークされます。
- FALSE に設定すると、変換できないハイパーリンクはスキップされ、変換は「converted」としてマークされます。
- デフォルトは、TRUE です。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの例です。

```
AllowSkippedHyperlinktoCauseIncomplete=false
```

APPLETCHUNKSIZE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

チャンク化機能の転送チャンク・サイズを設定します（バイト単位）。

- サイズが AppletChunkThreshold（6-40 ページ）を超過するファイルは、このチャンク・サイズで転送されます。
- デフォルトは、1MB です。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
AppletChunkSize=10000
```

関連項目

- AppletChunkThreshold（6-40 ページ）
- ChunkedRequestTrace（6-70 ページ）
- DisableHttpUploadChunking（6-156 ページ）

APPLETCHUNKTHRESHOLD

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

チャンク化機能の最小ファイル・サイズを設定します（バイト単位）。

- ファイル・サイズがこのサイズ未満の場合、チャンク化機能は使用されません。
- デフォルトは、1MB です。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
AppletChunkThreshold=500000
```

関連項目

- AppletChunkThreshold（6-40 ページ）
- ChunkedRequestTrace（6-70 ページ）
- DisableHttpUploadChunking（6-156 ページ）

ARCHIVE_COLLECTION

タイプおよび使用方法

- 構成変数（4-3 ページ）
- アーカイバ（5-3 ページ）

説明

デフォルトのアーカイブ・コレクション・ディレクトリのパスを定義します。

- デフォルトは、*<install_dir>/archives/* です。

場所

- *<install_dir>*/config/config.cfg
- *<install_dir>*/bin/intradoc.cfg

注意：この設定は、両方の構成ファイルに含める必要があります。

例

次は構成設定の例です。

```
ARCHIVE_COLLECTION=f:/stellent_archives/
```

ARCHIVERDOLOCKS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- アーカイバ（5-3 ページ）

説明

ファイル・システム・レベルで、アーカイブ・ファイルに対するロック・アルゴリズムをアクティブ化します。

- 複数のコンテンツ・サーバーで 1 つのインスタンスを共有して、インポートまたは転送を自動化する場合は、この変数を TRUE に設定します。
- ロード・バランシング構成で、1 つのインスタンスのみでインポートまたは転送を自動化する場合、この変数を設定する必要はありません。
- デフォルトの設定はありません。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
ArchiverDoLocks=true
```

関連項目

– ArchiverTraceLocks（6-45 ページ）
– ArchiverTraceLockMaintenance（6-44 ページ）
– ArchiverTouchMonitorInterval（6-43 ページ）

ARCHIVERTOUCHMONITORINTERVAL

タイプおよび使用方法

- 構成変数（4-3 ページ）
- アーカイバ（5-3 ページ）

説明

競合するコンテンツ・サーバーが、タイムスタンプの変更されていないアーカイブ・ロック・ファイルを失効として処理するまでの待機時間を定義します。また、この期間中にコンテンツ・サーバーは、維持しているロックに対する touch を 2 回実行します。

- アーカイブ・ロックは、適切なディレクトリにロック・ファイルを配置することで作成されます。ロックを持続するには、ファイルのタイムスタンプを定期的に変更する必要があります。そうでない場合、競合するコンテンツ・サーバーは、ロックが期限切れになったとみなします。
- この設定値が小さいほど、ロックのコンテンツ・サーバー間での移動が速くなります。値が大きいほど、ネットワーク・ドライブでの負荷の高いアクティビティによって touch リクエストが何秒間も遅延し、コンテンツ・サーバーが偽ってロックの失効を報告する可能性が高くなります。虚偽のロックの失効は、処理の重複を生じる可能性があります（コンテンツ・サーバーの状態への長期的な影響はありません）。
- デフォルトは、12000 ミリ秒（12 秒）です。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
ArchiverTouchMonitorInterval=15000
```

関連項目

- ArchiverDoLocks（6-42 ページ）
- ArchiverTraceLocks（6-45 ページ）
- ArchiverTraceLockMaintenance（6-44 ページ）

ARCHIVERTRACELOCKMAINTENANCE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- アーカイバ（5-3 ページ）
- デバッグ（5-20 ページ）

説明

未処理のアーカイブ・ロックおよび他のサーバーによって維持されているロックで検出されたタイムスタンプの変更について、トレース情報をコンテンツ・サーバー・コンソールにダンプしてタイムスタンプをメンテナンスします。

- アーカイブ・ロックは、適切なディレクトリにロック・ファイルを配置することで作成されます。ロックを持続するには、ファイルのタイムスタンプを定期的に変更する必要があります。そうでない場合、競合するコンテンツ・サーバーは、ロックが期限切れになったとみなします。
- デフォルトの設定はありません。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
ArchiverTraceLockMaintenance=true
```

関連項目

- ArchiverDoLocks（6-42 ページ）
- ArchiverTraceLocks（6-45 ページ）
- ArchiverTouchMonitorInterval（6-43 ページ）

注意：この変数は、Content Server 7.0 リリース以降では不要になりました。この機能は、「Admin Applets」の「System Audit Information」ページで利用できるようになりました。

ARCHIVERTRACELOCKS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- アーカイバ（5-3 ページ）
- デバッグ（5-20 ページ）

説明

アーカイブのロックを取得および解放するとき、コンテンツ・サーバー・コンソールにトレース情報をダンプします。

- デフォルトの設定はありません。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
ArchiverTraceLocks=true
```

関連項目

- ArchiverDoLocks（6-42 ページ）
- ArchiverTraceLockMaintenance（6-44 ページ）
- ArchiverTouchMonitorInterval（6-43 ページ）

注意：この変数は、Content Server 7.0 リリース以降では不要になりました。この機能は、「Admin Applets」の「System Audit Information」ページで利用できるようになりました。

ATTRIBUTEMAP

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

Active Directory または LDAP ユーザー属性のコンテンツ・サーバー・ユーザー情報フィールドへのマッピングを指定します。

- これはセミコロン区切りのユーザー属性マップのリストです。各値のセットをコロンで区切ります。たとえば、`mail:dEmail`。
- デフォルトは、`mail:dEmail;cn:dFullName;title:dUserType` です。

場所

Active Directory セキュリティ

- 「Active Directory Configuration」ページ→「Attribute Map」
- <*install_dir*>/data/users/config/filter.hda

LDAP セキュリティ

- 「Add/Edit LDAP Provider」ページ→「Attribute Map」
- <*install_dir*>/data/providers/*provider_name*/provider.hda

例

次は構成エントリの使用例です。

```
AttributeMap=department:uUserDepartment;mail:dEmail
```

AUTHORADDRESS

タイプおよび使用方法

- 値変数（4-34 ページ）
- ユーザー（5-40 ページ）

説明

コンテンツ・アイテムの作成者の電子メール・アドレスを指定します。

出力

使用に応じて、文字列またはブール値を返します。

- 標準的な使用 : コンテンツ・アイテムの作成者の電子メール・アドレスを文字列として返します。
- 条件文での使用 :
  - コンテンツ・アイテムの作成者の電子メール・アドレスが定義されている場合は、TRUE を返します。
  - コンテンツ・アイテムの作成者の電子メール・アドレスがない場合は、FALSE を返します。

例

修正が加えられたときに電子メールでコンテンツ・アイテム作成者に通知する場合に使用できます。

```
<$AuthorAddress$>
```

AUTHORDELETE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 内部セキュリティ（5-36 ページ）

説明

セキュリティ・グループに対する削除権限なしに、作成者が自身のリビジョンを削除できるようになります。

- TRUE に設定すると、作成者は、セキュリティ・グループに対する削除権限なしに、自身のリビジョンを削除できます。
- デフォルトは、空の文字列です。

場所

- 「System Properties」→「Content Security」タブ→「Allow author to delete revision」
- 「Admin Server」→「Content Security」→「Allow author to delete revision」
- <*install_dir*>/config/config.cfg

例

次は構成エントリの例です。

```
AuthorDelete=true
```

次はスクリプトの例です。

```
<$if AuthorDelete$>
    <$AuthorDelete$>
<$else$>
    false
<$endif$>
```

AUTOCAD2000PLOTTERFILEPATH

タイプおよび使用方法

❖ 構成変数（4-3 ページ）

❖ AutoCAD（5-14 ページ）

説明

AutoCAD 2000 プロッタ・ファイルのパスを定義します。

❖ ファイル・パスを文字列で返します。

❖ デフォルトは、空の文字列です。

場所

❖ 「Shared Inbound Refinery Configuration」画面→「AutoCAD Options」タブ→「AutoCAD 2000 Plotter File Path」

❖ <*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

構成エントリとして、AutoCAD 2000 プロッタ・ファイルのパスを定義します。

```
AutoCad2000PlotterFilePath=c:/autocad2000/plotter/
```

スクリプトとして、ファイル・パスを文字列で返します。

```
<$AutoCad2000PlotterFilePath$>
```

関連項目

– UseAutoCad2000（6-833 ページ）

AUTOMAXMERGETRACE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）
- デバッグ（5-20 ページ）

説明

Verity Indexer の自動チューンアップ機能のスクリプト・デバッグ・トレースを有効にします。Verity Integration は、Content Server とともに使用できるアドオン製品であり、購入することができます。

- TRUE に設定すると、スクリプト・デバッグ・トレースに自動チューンアップ・パラメータが含まれます。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
AutoMaxMergeTrace=true
```

関連項目

– DoAutoMaxMerge（6-170 ページ）
– ScriptDebugTrace（6-709 ページ）

AUTONUMBERPREFIX

タイプおよび使用方法

❖ 構成変数（4-3 ページ）

❖ コンテンツ・アイテム（5-6 ページ）

説明

自動的に番号付けされるすべてのコンテンツ ID に付加される接頭辞を定義します。

❖ 自動番号付けの接頭辞を返します（構成設定の値を返します）。

❖ 文字列を返します。

場所

❖ 「System Properties」→「Options」タブ→「Auto Name Prefix」

❖ 「Admin Server」→「General Configuration」→「Auto Number Prefix」

❖ <*install_dir*>/config/config.cfg

例

構成設定として、自動番号付けの接頭辞を定義します。

```
AutoNumberPrefix=HR
```

スクリプトとして、構成設定の値を返します。

```
<$AutoNumberPrefix$>
```

関連項目

– IsAutoNumber（6-376 ページ）

AUTONUMBERWIDTH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

6 桁を超えるコンテンツ ID 番号を使用できます。最大 1 億のコンテンツ・アイテム（8 桁）をサポートします。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
AutoNumberWidth=8
```

BATCHLOADERPATH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- バッチ・ローダー（5-3 ページ）

説明

デフォルトのバッチ・ロード・テキスト・ファイルのパスを定義します。

- ファイル・パスを文字列で返します。

場所

<*install_dir*>/bin/intradoc.cfg

例

次は Windows 構成エントリの例です。

```
BatchLoaderPath=c:/stellent/BatchLoader/batchfile.txt
```

次は Solaris/UNIX 構成エントリの例です。

```
BatchLoaderPath=/u1/intradoc3/batLd/batchfile
```

スクリプトとして、ファイル・パスを文字列で返します。

```
<$BatchLoaderPath$>
```

BATCHLOADERUSERNAME

タイプおよび使用方法

- 構成変数（4-3 ページ）
- バッチ・ローダー（5-3 ページ）

説明

バッチ・ローダー・ユーティリティの認可ユーザー名を定義します。ユーザー名は、admin ロールに属している必要があります。

- デフォルトは、sysadmin です。
- バッチ・ローダーのユーザー名を文字列で返します。

場所

<*install_dir*>/bin/intradoc.cfg

例

次は構成エントリの使用例です。

```
BatchLoaderUserName=sysadmin
```

BINDIR

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

コンテンツ・サーバーを起動する実行可能ファイル（IdcServer.exe など）のパスを定義します。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
BinDir=c:/stellent/bin/
```

BREAK

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- Idoc スクリプト（5-27 ページ）

説明

多くの場合、ループを終了するために使用されます。

- break 命令により、最も内側のループが終了します。
- 制御は、ループの end に続く最初の文から再開されます。

パラメータ

なし。

出力

なし。

例

N/A

BROWSERVERSIONNUMBER

タイプおよび使用方法

- 値変数（4-34 ページ）
- クライアント（5-4 ページ）

説明

クライアント・ブラウザのバージョン番号を取得します。

出力

ブラウザのバージョン番号を文字列で返します。

例

Stellent と互換性のあるブラウザ・バージョンかどうかを確認するために使用できます。

```
<$BrowserVersionNumber$>
```

C

タイプおよび使用方法

- Idoc スクリプトのコメント（3-3 ページ）
- Idoc スクリプト（5-27 ページ）

説明

Idoc スクリプト・コードのコメントを指定します。

例

コードにコメントを挿入します。

```
<$c = "Sets the variables to empty strings."$>
```

CACHEDRESULTROWCOUNT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）

説明

検索キャッシュのサイズを設定します。

- 検索キャッシュでは、最近の検索エンジンの問合せをキャッシュして、パフォーマンスが向上します。サイズを増加すると検索時間は短縮されますが、メモリー使用量が増加します。
- デフォルトは、20,000 行です。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
CachedResultRowCount=30000
```

CACHEINCLUDE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- Idoc スクリプト（5-27 ページ）

説明

この関数は、IdocScript 関数 inc と同様に動作します。この関数は、includeName に対応する動的 html インクルードを評価し、ページに表示します。この関数の相違点は、可能な場合、レンダリングされた html を再度評価するかわりに、キャッシュから取り出すことです。

詳細は、3-19 ページの「キーワードと関数の対比」を参照してください。

パラメータ

3 つの必須パラメータと 2 つのオプションのパラメータを取ります。次に例を示します。

```
cacheInclude(includeName, scope, lifeSpan [, cacheName, key])
```

パラメータ	説明
includeName	評価する動的 html 要素の名前。
scope	ユーザーごとに異なる html を表示する場合は session、すべてのユーザーに同じものを表示する場合は application に設定します。
lifeSpan	キャッシュ内のインクルードの存続期間（秒単位）。
cacheName	このデータをデフォルトのキャッシュではなく、名前付きキャッシュに入れる場合のオプションのパラメータ。空の文字列を渡すと、インクルードは、セッションのデフォルト・キャッシュに入れられます。
key	キャッシュの自動名前スコープによる競合を回避するには、一意のキーをオプションで指定できます。

例

各ユーザーの std_page_begin インクルードが 10 分間キャッシュされます。これは、キャッシュ内のユーザー 1 人当たり約 10k となります。

```
<$cacheInclude("std_page_begin", "session", 600)$>
```

関連項目

- inc（6-343 ページ）
- forceExpire（6-239 ページ）
- setExpires（6-733 ページ）
- setHttpHeader（6-734 ページ）
- setMaxAge（6-735 ページ）

CAPTIONENTRYWIDTH

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドの幅をパーセントで指定します。

出力

現在のメタデータ・フィールドの幅をパーセントで返します。

例

スクリプトとしての使用例を次に示します。

```
<$if isInfo$>
    <$captionFieldWidth="30%"$>
    <$captionEntryWidth="70%"$>
<$elseif isEditMode$>
    <$captionFieldWidth="20%"$>
    <$captionEntryWidth="80%"$>
<$endif$>
```

関連項目

– captionFieldWidth（6-63 ページ）

CAPTIONFIELDWIDTH

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドのキャプションの幅をパーセントで指定します。

出力

現在のメタデータ・フィールドのキャプションの幅をパーセントで返します。

例

スクリプトとしての使用例を次に示します。

```
<$if isInfo$>
    <$captionFieldWidth="30%"$>
    <$captionEntryWidth="70%"$>
<$elseif isEditMode$>
    <$captionFieldWidth="20%"$>
    <$captionEntryWidth="80%"$>
<$endif$>
```

関連項目

- fieldCaptionStyle（6-218 ページ）
- captionEntryWidth（6-62 ページ）

CGI_DEBUG

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Web サーバー（5-42 ページ）
- デバッグ（5-20 ページ）

説明

Web サーバー・フィルタを通じて詳細な情報をログできます。

- TRUE に設定すると、Web サーバー・フィルタとコンテンツ・サーバー間で送信されるデータおよびヘッダーのサマリーが、次のログ・ファイルに記録されます。
  - **IIS:** <*install_dir*>/idcplg/idc_cgi_isapi-<*instance*>.dll.log
  - **Apache または NES:** <*install_dir*>/data/users/authfilt.log
- IIS または Content Server v3.72 以下でこの設定を変更する場合、コンテンツ・サーバーおよび Web サーバーを再起動する必要があります。
- Content Server v3.72 以降での Apache および Netscape でこの設定を変更する場合は、コンテンツ・サーバーのみを起動する必要があります。
- デフォルトは、FALSE です。

場所

<*install_dir*>/data/users/config/filter.hda

例

次は構成エントリの使用例です。

```
CGI_DEBUG=true
```

関連項目

– CGI_RECEIVE_DUMP（6-65 ページ）
– CGI_SEND_DUMP（6-66 ページ）
– FILTER_DEBUG（6-234 ページ）
– PLUGIN_DEBUG（6-584 ページ）

CGI_RECEIVE_DUMP

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Web サーバー（5-42 ページ）
- デバッグ（5-20 ページ）

説明

Web サーバー・フィルタを通じてすべての受信データをログできます。

- TRUE に設定すると、Web サーバー・フィルタが受信するすべてのデータが次のログ・ファイルに記録されます。
  - **IIS:** <*install_dir*>/idcplg/idc_cgi_isapi-<*instance*>.dll.log
  - **Apache または NES:** <*install_dir*>/data/users/authfilt.log
- IIS または Content Server v3.72 以下でこの設定を変更する場合、コンテンツ・サーバーおよび Web サーバーを再起動する必要があります。
- Content Server v3.72 以降での Apache および Netscape でこの設定を変更する場合は、コンテンツ・サーバーのみを起動する必要があります。
- デフォルト値はありません。

場所

<*install_dir*>/data/users/config/filter.hda

例

次は構成エントリの使用例です。

```
CGI_RECEIVE_DUMP=true
```

関連項目

- CGI_DEBUG（6-64 ページ）
- CGI_SEND_DUMP（6-66 ページ）
- FILTER_DEBUG（6-234 ページ）
- PLUGIN_DEBUG（6-584 ページ）
- DisableGzipCompression（6-155 ページ）

CGI_SEND_DUMP

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Web サーバー（5-42 ページ）
- デバッグ（5-20 ページ）

説明

Web サーバー・フィルタを通じてすべての送信データをログできます。

- TRUE に設定すると、Web サーバー・フィルタが送信するすべてのデータが次のログ・ファイルに記録されます。
  - **IIS:** <*install_dir*>/idcplg/idc_cgi_isapi-<*instance*>.dll.log
  - **Apache または NES:** <*install_dir*>/data/users/authfilt.log
- IIS または Content Server v3.72 以下でこの設定を変更する場合、コンテンツ・サーバーおよび Web サーバーを再起動する必要があります。
- Content Server v3.72 以降での Apache および Netscape でこの設定を変更する場合は、コンテンツ・サーバーのみを起動する必要があります。
- デフォルト値はありません。

場所

<*install_dir*>/data/users/config/filter.hda

例

次は構成エントリの使用例です。

```
CGI_SEND_DUMP=true
```

関連項目

- CGI_DEBUG（6-64 ページ）
- CGI_RECEIVE_DUMP（6-65 ページ）
- FILTER_DEBUG（6-234 ページ）
- PLUGIN_DEBUG（6-584 ページ）

CGIFILENAME

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Web サーバー（5-42 ページ）

説明

Web サーバー・フィルタのファイル名を定義します。

- この設定は、IIS でのみ使用されます。
- Web サーバー・フィルタのファイル名を文字列で返します。
- デフォルトは、*iis_idc_cgi.dll* です。

場所

- コンテンツ・サーバー : <*install_dir*>/config/config.cfg
- 管理サーバー : <*install_dir*>/admin/config/config.cfg

例

構成設定として、Web サーバー・フィルタのファイル名を指定します。

```
CgiFileName=idc_cgi_isapi.dll
```

スクリプトとして、Web サーバー・フィルタのファイル名を文字列で返します。

```
<$CgiFileName$>
```

関連項目

– HttpCgiPath（6-312 ページ）

CHARMAP

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ローカライゼーション（5-29 ページ）

説明

Verity のキャラクタ・マッピングを設定します。Verity Integration は、Content Server とともに使用できるアドオン製品であり、購入することができます。

- Verity のロケールは、この変数が自動的に設定されます。
- デフォルトは、西ヨーロッパ言語ロケールの 1252 です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
CharMap=sjis
```

CHECKMKVDKDOCCOUNT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）

説明

Verity Integration で使用されます。Verity Integration は、Content Server とともに使用できるアドオン製品であり、購入することができます。

埋込みリンクのある Microsoft Word ファイルが検索索引に含まれるようになります。

- Verity の問題により、埋込みリンクのある Microsoft Word ドキュメントが検索索引に含まれない場合があります。これは、デフォルトでは、検索問合せでこれらのファイルが検索されないことを意味します。この構成設定は、このようなファイルのメタデータ（全テキストではない）を検索索引に含めます。
- TRUE に設定すると、埋込みリンクのある Word ファイルのメタデータ（全テキストではない）が、検索索引に含まれます。
- FALSE に設定すると、埋込みリンクのある Word ファイルは、検索索引に含まれない場合があります。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
ChkMkvdkDocCount=true
```

CHUNKEDREQUESTTRACE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）
- デバッグ（5-20 ページ）

説明

管理サーバーの「Content Server Output」ページに、チャンク・ファイル転送リクエストを出力します。

- TRUE に設定すると、「Content Server Output」ページにチャンク化リクエストが出力されます。
- FALSE に設定すると、チャンク化リクエストは出力されません。
- この設定によってデータを出力するには、DisableHttpUploadChunking（6-156 ページ）を使用してチャンク化機能を有効にする必要があります。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
ChunkedRequestTrace=true
```

関連項目

- AppletChunkSize（6-39 ページ）
- AppletChunkThreshold（6-40 ページ）
- DisableHttpUploadChunking（6-156 ページ）

CLASSPATH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- データベース（5-18 ページ）

説明

Java クラス・ファイルのパスを定義します。

- デフォルトの CLASSPATH は、*classes/*、*shared/classes/* および *shared/classes/server.zip* を指します。デフォルトを使用すると、構成ファイルに変数は表示されません。
- Oracle または Informix データベースを使用する場合、CLASSPATH に、*shared/classes/classes111.zip* などの JDBC ドライバ ZIP ファイルが含まれます。

場所

- 「System Properties」→「Paths」タブ→「Java Classpath」
- <*install_dir*>/bin/intradoc.cfg

例

次は構成エントリの使用例です。

```
CLASSPATH=$COMPUTEDCLASSPATH;crimson.jar;xalan.jar;jaxp.jar
```

CLEANUP

タイプおよび使用方法

- 構成変数（4-3 ページ）
- バッチ・ローダー（5-3 ページ）

説明

バッチ・ロードされるファイルをファイル・システムから削除するかどうかを決定します。

- TRUE に設定すると、バッチ・ロードされた各ファイルは、チェックインまたは更新が成功した後に、ハード・ドライブから削除されます。
- デフォルトは、FALSE です。

場所

<*install_dir*>/bin/intradoc.cfg

例

次は構成エントリの使用例です。

```
CleanUp=true
```

CLEARSCHEMADATA

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- スキーマ（5-33 ページ）

説明

スキーマの結果セットからデータを消去します。

パラメータ

この関数は、パラメータを 0 個、1 個または 2 個取ることができます。

- 0 個の引数を渡された場合は、データ・バインダを消去します。値を返しません。
- 1 個の引数を渡された場合、その引数は、現在の行から値を消去する結果セットの名前です。値を返しません。
- 2 個の引数を渡された場合、最初の引数は結果セットの名前で、2 番目の引数は消去するデータ・オブジェクトを識別するキーです。データが存在しない場合は 0 を、存在する場合は 1 を返します。戻り値に基づいて条件付きスクリプトを実行するには、True() または False() を使用します。

例

```
<$clearSchemaData()$>
```

関連項目

– loadSchemaData（6-501 ページ）

CLIENTCONTROLLED

タイプおよび使用方法

❖ 設定可能変数（4-34 ページ）

❖ クライアント（5-4 ページ）

説明

ODMA クライアントからページがアクセスされたかどうかをチェックします。

❖ この値は、文字列によって渡され、クライアントが提供する更新処理を制御します。

出力

❖ ODMA クライアントからページがアクセスされた場合は、TRUE を返します。

❖ ODMA クライアントからページがアクセスされなかった場合は、FALSE を返します。

例

ODMA クライアントからページがアクセスされたかどうかをチェックします。

```
<$ClientControlled$>
```

関連項目

– HasLocalCopy（6-284 ページ）

– IsNotLatestRev（6-423 ページ）

CLUSTERBINDIRRULE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- クラスタリング（5-5 ページ）

説明

クラスタの場合に必要です。

管理サーバーがクラスタの bin ディレクトリの場所を見つける方法を指定します。ローカル・スタブ・ディレクトリを使用する場合は、この変数を `local` に設定します。Veritas スタイルのクラスタで、クラスタの bin ディレクトリが共有ファイル・システムにある場合は、`shared` に設定します。

場所

- <*stub_dir*>/bin/intradoc.cfg
- <*stub_dir*>/admin/bin/intradoc.cfg

Veritas スタイルのクラスタの場合 :

- stellent/bin-*nodename*/intradoc.cfg
- stellent/admin/bin-*nodename*/intradoc.cfg

例

次は構成変数としての使用例です。

```
ClusterBinDirRule=local
```

関連項目

– ClusterNodeAddress（6-76 ページ）
– ClusterNodeName（6-77 ページ）

CLUSTERNODEADDRESS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- クラスタリング（5-5 ページ）

説明

クラスタの場合に必要です。

特定のクラスタ・ノードの IP アドレスまたは DNS 名を指定します。

別途指定がないかぎり、この変数によって指定された値は、次のすべての構成フラグのデフォルトとなります。

- IdcServerBindAddress
- IdcAdminServerBindAddress
- IntradocServerHostName
- SocketHostAddressSecurityFilter
- IdcAminServerHostName
- SocketServerAddress

場所

- <*stub_dir*>/bin/intradoc.cfg
- <*stub_dir*>/admin/bin/intradoc.cfg

Veritas スタイルのクラスタの場合 :

- stellent/bin-*nodename*/intradoc.cfg
- stellent/admin/bin-*nodename*/intradoc.cfg

例

次は構成変数としての使用例です。

```
ClusterNodeAddress=23.10.1.232
```

関連項目

– ClusterBinDirRule（6-75 ページ）
– ClusterNodeName（6-77 ページ）

CLUSTERNODENAME

タイプおよび使用方法

- 構成変数（4-3 ページ）
- クラスタリング（5-5 ページ）

説明

クラスタの場合に必要です。

クラスタのノードに一意の名前を指定します。空白を含まない短い簡潔な名前で、他のノード名と明確に区別する必要があります。

場所

- <*stub_dir*>/bin/intradoc.cfg
- <*stub_dir*>/admin/bin/intradoc.cfg

Veritas スタイルのクラスタの場合 :

- stellent/bin-*nodename*/intradoc.cfg
- stellent/admin/bin-*nodename*/intradoc.cfg

例

次は構成変数としての使用例です。

```
ClusterNodeName=MSP1
```

関連項目

– ClusterBinDirRule（6-75 ページ）
– ClusterNodeAddress（6-76 ページ）

cmDiscussionDocType

タイプおよび使用方法

- Collaboration Manager の構成変数（4-4 ページ）
- Collaboration Manager（5-5 ページ）

説明

ディスカッション・アイテムの識別に使用するドキュメント・タイプを指定します。

- Collaboration Manager のインストール中に追加されます。
- コンテンツ・タイプの「Type」フィールドに指定可能な任意の値を使用できます。

場所

<*install_dir*>/<*instancename*>/data/components/CollaborationManager/config.cfg

例

次は構成エントリの使用例です。

```
cmDiscussionDocType=DISCUSS
```

CMMEMOFIELDSIZE

タイプおよび使用方法

- Collaboration Manager の構成変数（4-4 ページ）
- Collaboration Manager（5-5 ページ）

説明

メモ・フィールドのサイズを設定します。より大きいアクセス・ファイルを保存するには、フィールド・サイズを増加する必要があります。

- Collaboration Manager のインストール中に追加されます。
- `cmMemoFieldSize` に指定した値は、Content Server の `MemoFieldSize` を設定するために使用され、*<install_dir>*/*<instancename>*/config/config.cfg に書き込まれます。
- この変数は、バージョン 7.0 より前の Collaboration Manager では利用できませんでした。
- デフォルト値は、1000 です。マルチバイト・システムを使用する場合、または Record Manager がインストールされていない場合は、この値を 3000 に増加できます。

場所

<install_dir>/*<instancename>*/data/components/CollaborationManager/install.cfg

例

次は構成エントリの使用例です。

```
cmMemoFieldSize=2000
```

関連項目

– MemoFieldSize（6-532 ページ）

CMZONEDSECURITYFIELDS

タイプおよび使用方法

- Collaboration Manager の構成変数（4-4 ページ）
- Collaboration Manager（5-5 ページ）

説明

検索でゾーン・セキュリティ・フィールドを使用可能にするかどうかを指定します。ゾーン・セキュリティ・フィールドにより、検索のパフォーマンスを向上させることができます。

- Collaboration Manager のインストール中に追加されます。
- TRUE に設定すると、ゾーン・セキュリティ・フィールドが有効になり、`ZonedSecurityFields` が xClbraUserList,xClbraAliasList に設定されます。
- デフォルトは、TRUE です。
- この値を変更した後、検索索引の再構築が必要になります。

場所

<*install_dir*>/<*instancename*>/data/components/CollaborationManager/config.cfg

例

次は構成エントリの使用例です。

```
cmZonedSecurityFields=true
```

COLLECTIONDELETEENABLED

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Folders および WebDAV（5-26 ページ）

説明

ユーザーがフォルダおよびコンテンツ・アイテムを削除できるように「Delete」アイコンを表示するかどうかを指定します。

- TRUE に設定すると、ユーザーが「Exploring」ページからフォルダおよびコンテンツ・アイテムを削除できるように「Delete」アイコンが表示されます。
- FALSE に設定すると、「Delete」アイコンは非表示になります。
- この変数は、インストールの際に指定したオプションに従って、Folders インストール・プログラムによって自動的に設定されます。

場所

<*install_dir*>/custom/folders/folders_environment.cfg

例

次は構成エントリの使用例です。

```
CollectionDeleteEnabled=true
```

COLLECTIONHIDDENMETA

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Folders および WebDAV（5-26 ページ）

説明

仮想フォルダまたはコンテンツ・アイテムの表示と非表示をユーザーが指定できるようにします。

- この設定により、「Exploring」ページに「Hide」/「Unhide」アイコンが追加されます。
- この設定により、「Folder Configuration」ページに**「Show hidden when browsing」**チェック・ボックスが追加されます。
- これはインストールの際に指定されたオプションに従って、Folders インストーラ・プログラムによって自動的に設定されます。

場所

<*install_dir*>/data/components/folders/install.cfg

関連項目

この機能は、手動で有効化できます。詳細は、Folders and WebDAV ガイドを参照してください。

COLLECTIONINHIBITUPDATEMETA

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Folders および WebDAV（5-26 ページ）

説明

メタデータの伝播機能で伝播されないフィールドを定義します。メタデータの伝播により、コントリビュータは、フォルダからサブフォルダおよびコンテンツ・アイテムにメタデータ値をコピーできます。

- この変数は、インストールの際に指定したフィールド名に従って、Folders インストーラ・プログラムによって自動的に設定されます。

場所

<*install_dir*>/custom/folders/folders_environment.cfg

関連項目

この機能は、手動で有効化できます。詳細は、Folders and WebDAV ガイドを参照してください。

COLLECTIONLOCATION

タイプおよび使用方法

- 構成変数（4-3 ページ）
- アーカイバ（5-3 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

デフォルトのアーカイブ・コレクションの場所を定義します。

- デフォルトは、<*install_dir*>/*archives*/ です。

場所

- <*install_dir*>/config/config.cfg
- <*install_dir*>/bin/intradoc.cfg

例

次は構成エントリの使用例です。

```
CollectionLocation=c:/stellent/new/archives/
```

COLLECTIONMETA

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Folders および WebDAV（5-26 ページ）

説明

各仮想フォルダの一意のコレクション ID の保存に使用するメタデータ・フィールドを指定します。

- この設定が構成ファイルに指定されていない場合、デフォルト値は `xCollectionID` です。
- この変数は、インストールの際に指定したフィールド名に従って、Folders インストーラ・プログラムによって自動的に設定されます。

場所

<*install_dir*>/custom/folders/folders_environment.cfg

関連項目

この機能は、手動で有効化できます。詳細は、Folders and WebDAV ガイドを参照してください。

COLLECTIONMOVEENABLED

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Folders および WebDAV（5-26 ページ）

説明

「Exploring」ページに「Move」アイコンを表示するかどうかを指定します。

- TRUE に設定すると、ユーザーがフォルダおよびコンテンツ・アイテムを他のフォルダに移動できるように、「Move」アイコンが表示されます。
- FALSE に設定すると、「Move」アイコンは非表示になります。
- この変数は、インストールの際に指定したオプションに従って、Folders インストール・プログラムによって自動的に設定されます。

場所

<*install_dir*>/custom/folders/folders_environment.cfg

例

次は構成エントリの使用例です。

```
CollectionMoveEnabled=true
```

COLLECTIONREADONLYMARKEDFOLDERS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Folders および WebDAV（5-26 ページ）

説明

システム・レベルのフォルダ（Contribution Server のフォルダやごみ箱など）を変更（読取り専用ステータスに設定）できるかどうかを指定します。

- TRUE に設定すると、システム・レベルのフォルダを変更できないようにします。
- FALSE に設定すると、変更できるようになります。
- デフォルトでは、この変数は TRUE に設定されます。
- FALSE に設定する場合は、権限のある担当者のみに変更が可能となるように、システムレベルのフォルダに適切なセキュリティ・グループを割り当てる必要があります。システムレベルのフォルダの変更は、特に変更する理由がある場合にのみ行います。

場所

<*install_dir*>/custom/folders/folders_environment.cfg

例

次は構成エントリの使用例です。

```
CollectionReadOnlyMarkedFolders=true
```

COLLECTIONREADONLYMETA

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Folders および WebDAV（5-26 ページ）

説明

仮想フォルダとその内容を読取り専用にするかどうかを、ユーザーが指定できるようにします。

- 読取り専用は、インスタンスを実行する動作環境のルールの制約を受けます。
- この設定により、「Exploring」ページに「Read-only」/「Editable」アイコンが追加されます。
- この設定は、現在のフォルダ・レベルにのみ適用されます。サブフォルダは、親からステータスを継承しません。
- この変数は、インストールの際に指定したオプションに従って、Folders インストール・プログラムによって自動的に設定されます。

場所

<*install_dir*>/data/components/folders/install.cfg

関連項目

この機能は、手動で有効化できます。詳細は、Folders and WebDAV ガイドを参照してください。

COLLECTIONRELEASEDONLY

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Folders および WebDAV（5-26 ページ）

説明

コンテンツ・アイテムに対する読取り権限を持つ作成者およびユーザーが、改訂されたドキュメントを表示できるかどうかを指定します。

- TRUE に設定した場合、アイテムが未リリースで、作成者以外のユーザーがアイテムにアクセスした場合、コンテンツ・アイテムはまったく表示されません。
- FALSE に設定した場合、作成者およびコンテンツ・アイテムに対する読取り権限を持つすべてのユーザーは、最新バージョンを表示できます。
- デフォルトでは、この変数は FALSE に設定されます。

場所

<*install_dir*>/custom/folders/folders_environment.cfg

例

次は構成エントリの使用例です。

```
CollectionReleasedOnly=false
```

COLLECTIONSECURITYREADONLY

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Folders および WebDAV（5-26 ページ）

説明

ユーザーが、コンテンツ・アイテムに対する読取り権限のみを持つ場合、そのアイテムを表示できるかどうかを指定します。

- TRUE に設定した場合、読取り権限のみを持つユーザーは、コンテンツ・アイテムを表示できます。
- FALSE に設定した場合、ユーザーがコンテンツ・アイテムを表示するには、書込み権限以上が必要です。
- この変数は、インストールの際に指定したオプションに従って、Folders インストール・プログラムによって自動的に設定されます。

場所

<*install_dir*>/custom/folders/folders_environment.cfg

例

次は構成エントリの使用例です。

```
CollectionSecurityReadOnly=false
```

COLLECTIONTRASHDELETEDATE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Folders および WebDAV（5-26 ページ）

説明

ごみ箱機能の日付メタデータ・フィールドを定義します。

- この変数を設定すると、ごみ箱機能の日付メタデータ・フィールドが定義され、ごみ箱に移動するアイテムの削除日時をメタデータとして記録できるようになります。
- 設定した場合、「Trash」フォルダに削除日時が表示されます。
- インストールの際にごみ箱機能が有効な場合、この変数は Folders インストール・プログラムによって自動的に設定されます。

場所

<*install_dir*>/custom/folders/folders_environment.cfg

関連項目

この機能は、手動で有効化できます。詳細は、Folders and WebDAV ガイドを参照してください。

COLLECTIONTRASHDELETELOCATION

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Folders および WebDAV（5-26 ページ）

説明

ごみ箱機能の親フォルダを指定する整数メタデータ・フィールドを定義します。

- この変数を設定すると、元の親フォルダを、「Trash」フォルダに移動するアイテムのメタデータとして記録できるようになります。
- 設定した場合、ユーザーは「Trash」フォルダから、削除済のアイテムをリストアできます。
- 設定した場合、「Trash」フォルダに削除日時が表示されます。
- インストールの際にごみ箱機能が有効な場合、この変数は Folders インストール・プログラムによって自動的に設定されます。

場所

<*install_dir*>/custom/folders/folders_environment.cfg

関連項目

この機能は、手動で有効化できます。詳細は、Folders and WebDAV ガイドを参照してください。

COLLECTIONTRASHDELETEOLDNAME

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Folders および WebDAV（5-26 ページ）

説明

削除ファイル名を追跡するごみ箱機能のメタデータ・フィールドを定義します。

- この変数を設定すると、「Trash」フォルダで名前の競合のために名前変更されたファイルを、元のファイル名でリストアできます。
- この変数は、ごみ箱機能のファイル名メタデータ・フィールドを定義します。これにより、元のファイル名を、「Trash」フォルダに移動するアイテムのメタデータとして記録できるようになります。
- インストールの際にごみ箱機能が有効な場合、この変数は Folders インストール・プログラムによって自動的に設定されます。

場所

<*install_dir*>/custom/folders/folders_environment.cfg

関連項目

この機能は、手動で有効化できます。詳細は、Folders and WebDAV ガイドを参照してください。

COLLECTIONTRASHDELETER

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Folders および WebDAV（5-26 ページ）

説明

ユーザーによる削除を追跡するごみ箱機能のメタデータ・フィールドを定義します。

- この変数を設定すると、ごみ箱機能のメタデータ・フィールドが定義され、ユーザーのログイン情報が「Trash」フォルダに移動するアイテムのメタデータとして記録されます。
- 設定した場合、ユーザーは、自分が「Trash」フォルダに移動したアイテムのみを表示するか、全ユーザーが「Trash」フォルダに移動したすべてのアイテムを表示するかを選択できます。
- インストールの際にごみ箱機能が有効な場合、この変数は Folders インストール・プログラムによって自動的に設定されます。

場所

<*install_dir*>/custom/folders/folders_environment.cfg

関連項目

この機能は、手動で有効化できます。詳細は、Folders and WebDAV ガイドを参照してください。

COLLECTIONWEBDAVSERVER

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Folders および WebDAV（5-26 ページ）

説明

Windows Explorer の表示の切替えが可能になります。

- URL に設定すると、ユーザーは Content Server または Collaboration Manager の Web ページから Windows Explorer の WebDAV フォルダに表示を切り替えられます。
- この機能を使用するには、WebDAV 機能がコンテンツ・サーバーにインストールされている必要があります。
- Microsoft Internet Explorer 5.0 以降では、仮想探索フォルダに、Web フォルダ・アイコンが追加されます（このアイコンは、他のブラウザには表示されません）。
- Collaboration Manager では、この設定により、情報ページおよびフォルダに、**「Open Web Folder」**アイコンが追加されます。
- インストールの際に WebDAV 機能が有効な場合、この変数は Folders インストール・プログラムによって自動的に設定されます。

場所

<*install_dir*>/custom/folders/folders_environment.cfg

例

```
CollectionWebDAVServer=http://my_computer/root/idcplg/webdav
```

COLUMNMAPFILE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- データベース（5-18 ページ）

説明

列マッピングの HTM リソース・ファイルを指定します。

- コンテンツ・サーバーでは、列マッピング・ファイルを使用して、データベース列名をすべて大文字（Oracle で必要）にマップします。
- パスの参照がない場合、デフォルトのパスは <*install_dir*>/shared/config/resources/ です。
- Content Server で提供されている標準の列マッピング・ファイルは、*upper_clmns_map.htm* です。
- デフォルト値はありません。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
ColumnMapFile=upper_clmns_map.htm
```

COMPRESSIONMINHEIGHT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Compression（5-18 ページ）

説明

Oracle Compression アドオン・モジュールによって処理されるグラフィックの最小の高さをピクセル数で設定します。

- グラフィックが、指定された高さ以上の場合、Compression によって処理されます。
- グラフィックが、指定された高さ未満の場合、Compression によって処理されません。
- デフォルトは、200 です。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
CompressionMinHeight=100
```

COMPRESSIONMINWIDTH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Compression（5-18 ページ）

説明

Oracle Compression アドオン・モジュールによって処理されるグラフィックの最小幅をピクセル数で設定します。

- グラフィックが、指定された幅以上の場合、Compression によって処理されます。
- グラフィックが、指定された幅未満の場合、Compression によって処理されません。
- デフォルトは、200 です。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
CompressionMinWidth=100
```

COMPUTEDOCURL

タイプおよび使用方法

- 動的変数（4-20 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

この関数は、ページのデータに基づいて、コンテンツ・アイテムに対する URL を算定します。

たとえば、アイテムの ResultSet に対するループ処理の際に、この関数を呼び出して、アイテムの URL を生成できます。

次の情報がページに存在する必要があります。

- dDocAccount（オプション）
- dDocName
- dDocType
- dProcessingState
- dRevLabel
- dSecurityGroup
- dWebExtension

パラメータ

1 つのパラメータを取ります。

- 1 つのパラメータは、相対 URL かどうかを示し、TRUE（1）または FALSE（0）に設定されます。

COMPUTERENDITIONURL

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

指定されたレンディションの URL を返します。

パラメータ

パラメータを 3 つ取ります。

- 最初のパラメータは、コンテンツ・アイテムの URL です。
- 2 番目のパラメータは、*dRevLabel* 値です。
- 3 番目のパラメータは、*dRendition1* 値です。*dRendition1* には、次の値が含まれます。
  - **T** = サムネイル・レンディション
  - **X** = XML レンディション

出力

レンディションの完全な URL を文字列として返します。

例

レンディションの URL を文字列として返します。

```
<$computeRenditionUrl(url, dRevLabel, dRendition1)$>
```

COMPUTERNAME

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Inbound Refinery（5-11 ページ）

説明

Inbound Refinery がインストールされたコンピュータに割り当てられている名前を参照します。

- この設定は、Inbound Refinery が実行されている場所を示すためにログ・ファイルで使用されます。ログは起動時に、「DocRefinery running on <ComputerName> for <IDC_Name>」というログ・エントリを発行します。
- 文字列を返します。
- デフォルトは、*ComputerNameNotSet* です。

場所

<*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
ComputerName=maincomputer
```

CONFIGDIR

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ディレクトリおよびパス（5-22 ページ）
- クラスタリング（5-5 ページ）

説明

config ディレクトリのパスを定義します。

- デフォルトは、<*install_dir*>/*config*/ です。

場所

<*install_dir*>/bin/intradoc.cfg

例

次は構成設定の例です。

```
ConfigDir=c:/stellent/config/
```

CONNECTIONNAME

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Inbound Refinery（5-11 ページ）

説明

Inbound Refinery 接続に割り当てられている名前を参照します。

- 接続名を文字列で返します。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
ConnectionName=MAINCONNECTION
```

CONTENT_LENGTH

タイプおよび使用方法

- Web サーバー変数（4-36 ページ）
- Web サーバー（5-42 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

重要：この設定は、バージョン 7.0 以降では不要です。Web サーバー・フィルタで、この情報は送信されなくなりました。

クライアントによって指定された、リクエストされたコンテンツ・アイテムの長さをバイト単位で取得します。

- この変数は、現在のゲートウェイ・プログラム・リクエストに固有のものです。

出力

コンテンツの長さ（バイト数）を返します。

例

次はページまたはログに出力される情報の例です。

```
CONTENT_LENGTH=0
```

次は Idoc スクリプト文または評価の一部としての使用例です。

```
<$if CONTENT_LENGTH$>
<!--statement-->
```

CONVERSIONCACHEEXPIRATIONPERIODINDAYS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Dynamic Converter（5-17 ページ）

説明

Dynamic Converter コンテンツがキャッシュに保持される最小日数を定義します。

- Dynamic Converter によって変換されたコンテンツは、（最終アクセス日に基づいて）この日数が経過するまで、キャッシュから削除されません。次回のクリーンアップ・スキャンの実行時期に応じて、この日数を超えてコンテンツがキャッシュ内に保持されたままになる場合もあります。6-122 ページの「DatedCacheIntervalDays」を参照してください。
- この設定は、現在コンテンツ・サーバーに存在しないファイル、または強制変換や事前変換によって生成されていないファイルにのみ適用されます。キャッシュのクリーンアップ・ルールの詳細は、6-516 ページの「MaxConversionCacheSizeInMegs」を参照してください。
- デフォルトは、7 です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
ConversionCacheExpirationPeriodInDays=14
```

関連項目

- DatedCacheIntervalDays（6-122 ページ）
- MaxConversionCacheSizeInMegs（6-516 ページ）

CONVERTXMLASWEBVIEWABLE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- XML Converter（5-16 ページ）

説明

ファイルが、XML の Web 表示可能なレンディションにのみ変換されるかどうかを定義します。

- TRUE に設定すると、ファイルは XML 形式にのみ変換されます。他の Web で表示可能な形式（PDF など）への変換は起こりません。
- FALSE に設定すると、XML 形式に加えて、他の Web で表示可能な形式にも変換できます。
- デフォルトは、FALSE です。

場所

- 「Local Inbound Refinery Configuration」画面→「XML Options」タブ→「Convert XML as the only web-viewable file」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
ConvertXmlAsWebviewable=true
```

CookieAuthTimeoutInDays

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

ExtranetLook コンポーネントとともに使用されます。この変数は、Web リクエストを監視してリクエストが認証されているかどうかを判断する CookieLoginPlugin とともに使用されます。ブラウザの Cookie ベースの認証をサポートできます。

- `CookieAuthTimeoutInDays` を正の数値に設定すると、Cookie はその日数の間タイムアウトしません。ユーザーは、Cookie をクリアするログアウト・オプションが必要です。
- ゼロ（0）以下に設定すると、Cookie はブラウザ・セッションの間のみ存続します。

場所

<*install_dir*>/config/config.cfg

例

```
CookieAuthTimeoutInDays=2
```

関連項目

– CookieProxyPassword（6-108 ページ）
– IsWebServerPagesOnly（6-453 ページ）

CookieProxyPassword

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

ExtranetLook コンポーネントとともに使用されます。この変数は、ユーザーが Cookie 認証を使用して Web サイトにログインした場合に、パスワードの暗号化に使用されます。この変数は、Web リクエストを監視してリクエストが認証されているかどうかを判断する CookieLoginPlugin とともに使用されます。ブラウザの Cookie ベースの認証をサポートできます。この変数は、任意の値に設定できます。

場所

<*install_dir*>/config/config.cfg

例

```
CookieProxyPassword=x2x
```

関連項目

- CookieAuthTimeoutInDays（6-107 ページ）
- IsWebServerPagesOnly（6-453 ページ）

CORECONTENTONLY

タイプおよび使用方法

- 設定可能変数（4-34 ページ）
- ページ表示（5-30 ページ）

説明

この変数をサービス・リクエストの URL に設定すると、フォームのみが表示されます。`std_page_begin` および `std_page_end` インクルード・ファイルは表示されません。

出力

なし。

CREATEALTERNATEMETAFILE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

ユーザーがメタデータ専用のコンテンツを代替ファイルとして発行できるようにします。

- メタデータ専用の代替ファイルがシステムによって作成されます。この場合、コントリビュータは代替ファイルを指定しません。かわりに、テンプレートおよびコンテンツ・アイテムのメタデータを使用して、システムによって代替ファイルが作成されます。
- AllowAlternateMetaFile（6-23 ページ）を TRUE に設定すると、追加のチェック・ボックスがチェックイン・ページに追加されます。このチェック・ボックスを選択すると、*createAlternateMetaFile* 変数が有効になります。
- バッチ・ローダーで使用する場合は、バッチ・ロード・ファイルで *createAlternateMetaFile* が TRUE に設定されている必要があります。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
createAlternateMetaFile=true
```

関連項目

– AllowAlternateMetaFile（6-23 ページ）
– createPrimaryMetaFile（6-112 ページ）

CREATEPDFTHUMBNAILS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

PDF ファイルのサムネイル・イメージを作成するかどうかを定義します。

- TRUE に設定すると、PDF ファイルにサムネイルが作成されます。
- FALSE に設定すると、PDF ファイルにサムネイルは作成されません。
- デフォルトは、FALSE です。

場所

- 「Local Inbound Refinery Configuration」画面→「Thumbnail Options」タブ→「Create thumbnail images for PDF files created by the Inbound Refinery」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
CreatePDFThumbnails=true
```

関連項目

- ImageAlchemyExePath（6-342 ページ）
- IsThumbnailPresent（6-443 ページ）

CREATEPRIMARYMETAFILE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

ユーザーがメタデータ専用のコンテンツをプライマリ・ファイルとして発行できるようにします。

- メタデータ専用のプライマリ・ファイルがシステムによって作成されます。この場合、コントリビュータはプライマリ・ファイルを指定しません。かわりに、テンプレートおよびコンテンツ・アイテムのメタデータを使用して、システムによってプライマリ・ファイルが作成されます。
- AllowPrimaryMetaFile（6-33 ページ）を TRUE に設定すると、追加のチェック・ボックスがチェックイン・ページに追加されます。このチェック・ボックスを選択すると、*createPrimaryMetaFile* 変数が有効になります。
- バッチ・ローダーで使用する場合は、バッチ・ロード・ファイルで *createPrimaryMetaFile* が TRUE に設定されている必要があります。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
createPrimaryMetaFile=true
```

関連項目

- AllowPrimaryMetaFile（6-33 ページ）
- createAlternateMetaFile（6-110 ページ）

CREATEXMLRENDITION

タイプおよび使用方法

- 構成変数（4-3 ページ）
- XML Converter（5-16 ページ）

説明

変換された XML ファイルをレンディションとして識別するかどうかを定義します。

- サムネイル・イメージを作成する場合は、サムネイルが最初のレンディションです。
- TRUE に設定すると、変換された XML ファイルはレンディションとして識別されます。
- FALSE に設定すると、変換された XML ファイルはレンディションとして識別されません。
- デフォルトは、FALSE です。

場所

- 「Local Inbound Refinery Configuration」画面→「XML Options」タブ→「Create XML renditions for all supported formats」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
CreateXMLRendition=true
```

CURRENT_DATE

タイプおよび使用方法

❖ 値変数（4-34 ページ）

❖ 日付と時間（5-19 ページ）

説明

現在の日付と時間を返します。

❖ この変数は、頻繁に使用される dateCurrent（6-121 ページ）に類似しています。

パラメータ

なし。

出力

ユーザーのロケールに従って書式設定された文字列を返します。

例

スクリプトとしての使用例を次に示します。

```
<$CURRENT_DATE$>
```

関連項目

– dateCurrent（6-121 ページ）

CURRENT_ROW

タイプおよび使用方法

- 値変数（4-34 ページ）
- ResultSet（5-32 ページ）

説明

ResultSet の現在の行がどれであるかを評価します。

- ResultSet の最初の行は、行ゼロ（0）です。

出力

行番号を返します。

例

スクリプトとしての使用例を次に示します。

```
<$CURRENT_ROW$>
```

CUSTOMCONVERSIONWAITTIME

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Inbound Refinery（5-11 ページ）

説明

カスタム変換のタイムアウトを定義します。

- カスタム変換のタイムアウト値を返します。
- この変数の後に、次のいずれかのフラグが必要です。

フラグ	説明
#factor	この値とファイル・サイズを掛け合せて、カスタム変換の許容時間を決定します。（結果の時間は、#max および #min フラグによって制限されます。）デフォルト値は、1 です。ビジーまたは処理の遅いシステムでは、この値を増加します。
#max	カスタム変換が完了するまでの Inbound Refinery の最大待機時間を秒数で指定します。デフォルトは、10 秒です。
#min	カスタム変換が完了するまでの Inbound Refinery の最小待機時間を秒数で指定します。デフォルトは、1 秒です。

場所

- 「Local Inbound Refinery Configuration」画面→「Legacy Conversions」タブ→「Custom Converter Timeout」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
CustomConversionWaitTime#max=60
```

CUSTOMCONVERTERPATH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Inbound Refinery（5-11 ページ）

説明

カスタム変換の実行に使用される実行ファイルのパスを定義します。

- 指定するファイルは、Win32 実行可能ファイルである必要があります。
- デフォルトは、空の文字列です。

場所

- 「Local Inbound Refinery Configuration」画面→「Legacy Conversions」タブ→「Custom Converter Path」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
CustomConverterPath=c:/out/custom.bat
```

DATABASEPRESERVECASE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- データベース（5-18 ページ）

説明

データベースからの大 / 小文字を区別するかどうかを定義します。

- TRUE に設定すると、大 / 小文字が区別されます。
- Oracle および大 / 小文字が区別される他のデータベースでは、TRUE に設定する必要があります。
- デフォルトは、TRUE です。

場所

- 「System Properties」→「Database」タブ→「Enable database preserve case」
- <*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
DatabasePreserveCase=false
```

DATABASEWILDCARDS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- データベース（5-18 ページ）

説明

データベースの問合せで、複数の文字を表すために使用するワイルドカード文字を定義します。

- JDBC データベースの場合、デフォルトは % です。
- Microsoft Access の場合、デフォルトは * です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
DatabaseWildcards=%
```

DATADIR

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ディレクトリおよびパス（5-22 ページ）
- クラスタリング（5-5 ページ）

説明

data ディレクトリのパスを定義します。

- デフォルトは、<*install_dir*>*/data/* です。

場所

<*install_dir*>/bin/intradoc.cfg

例

次は構成エントリの使用例です。

```
DataDir=c:/stellent/data/
```

DATECURRENT

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 日付と時間（5-19 ページ）

説明

現在の日付と時間を返します。

- 現在の日付と時間を返したり、日付の評価を使用してコマンドを作成したりするために使用できます。

パラメータ

1 つのオプションのパラメータを取ります。

- オプションのパラメータは、整数のみです。これにより、現在の日付に関連して、指定された日数だけ日付を調整します。

出力

ユーザーのロケールに従って書式設定された日付を返します。

例

次の例では、日付は、デフォルトの「English-US」ロケールに従って書式設定されます。

```
m/d/yy h:mm XM
```

現在の日付と現在の時間（たとえば、8/12/01 1:55 PM）を返します。

```
<$dateCurrent()$>
```

10 日後の日付と現在の時間（たとえば、8/22/01 1:55 PM）を返します。

```
<$dateCurrent(10)$>
```

10 日前の日付と現在の時間（たとえば、8/2/01 1:55 PM）を返します。

```
<$dateCurrent(-10)$>
```

関連項目

– CURRENT_DATE（6-114 ページ）

DATEDCACHEINTERVALDAYS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Dynamic Converter（5-17 ページ）

説明

Dynamic Converter のキャッシュされたコンテンツを削除するために、スキャンするまでの日数を定義します。

- コンテンツ・サーバーは、この間隔でスキャンを行って、キャッシュされた古いコンテンツを見つけて削除します。
- デフォルトは、7 です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
DatedCacheIntervalDays=30
```

関連項目

– ConversionCacheExpirationPeriodInDays（6-105 ページ）
– MaxConversionCacheSizeInMegs（6-516 ページ）

DATEOUTPUTFORMAT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 日付と時間（5-19 ページ）
- 索引付け（5-28 ページ）

説明

インデクサの日付の出力形式を設定します。

- この変数は、Verity Integration 以外の索引付けエンジンで使用されます。Verity Integration は、Content Server とともに使用できるアドオン製品であり、購入することができます。
- デフォルト値はありません。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
DateOutputFormat=M/d/yyyy hh:mm:ss a
```

DCMaxFileSize

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Dynamic Converter（5-17 ページ）

説明

Oracle Dynamic Converter で処理されるファイルの最大サイズ（バイト数）を定義します。

- 指定された値よりもソース・ファイルの方が大きい場合、変換は発生しません。
- デフォルトは、20000000（20MB）です。
- Dynamic Converter バージョン 5.0 以上で使用されます。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
DCMaxFileSize=10000000
```

DCSHOWEXPORTLINK

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- Dynamic Converter（5-17 ページ）

説明

この関数は、コンテンツ・アイテムを変換するように Dynamic Converter が構成されているかどうかを検証します。アイテムの `dFormat` の値に基づいて、値を返します。

通常、この関数は、Dynamic Converter リンクを条件付きで表示するために検索結果ページで使用されます。

出力

- Dynamic Converter がコンテンツ・アイテムを変換するように構成されている場合は、TRUE を返します。
- Dynamic Converter が構成されていない場合は、FALSE を返します。

例

```
<$QueryText = "dDocTitle <substring> `test`"$>
<$executeService("GET_SEARCH_RESULTS")$>
<$loop SearchResults$>
    <$if dcShowExportLink()$>
        <a href="<$HttpCgiPath$>?IdcService=
        GET_DYNAMIC_CONVERSION&dID=<$dID$>">
        HTML Conversion of <$dDocTitle$></a>
    <$endif$>
<$endloop$>
```

DCTIMEOUT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Dynamic Converter（5-17 ページ）

説明

Oracle Dynamic Converter 変換処理のタイムアウト値を分単位で設定します。

- ドキュメントを HTML に動的変換するまでの待機時間（分）を定義します。
- デフォルトは、3 分です。
- Dynamic Converter バージョン 5.0 以上で使用されます。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
DCTimeOut=5
```

DCVIEWFORMAT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Dynamic Converter（5-17 ページ）

説明

Oracle Dynamic Converter によって変換されるソース・ファイルのレンディションを定義します。

- *native* に設定すると、ネイティブ・ファイルが変換されます。
- *webviewable* に設定すると、Web で表示可能なファイルが変換されます。
- *alternate* に設定すると、代替ファイルが変換されます。
- デフォルトは、*native* です。
- Dynamic Converter バージョン 5.0 以上で使用されます。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
DCViewFormat=webviewable
```

DEBUGMODE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- デバッグ（5-20 ページ）

説明

コンテンツ・サーバーをデバッグ・モードで実行するかどうかを定義します。この変数は、インストール中およびコンテンツ・サーバーの更新の際に設定されます。

デフォルトは、FALSE です。

場所

<*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

```
DebugMode=true
```

DEBUGSTDCONVERSION

タイプおよび使用方法

- 構成変数（4-3 ページ）
- デバッグ（5-20 ページ）

説明

コンテンツ・サーバーによって発行される変換 / ステップの定義を上書きする構成設定。

パラメータ

この変数は、1 つの引数（新規定義のパス）を取ります。

場所

<*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

```
DebugStdConversion=/temp/idcconversions
```

DEFAULT ACCOUNTS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 内部セキュリティ（5-36 ページ）
- ユーザー（5-40 ページ）

説明

匿名ユーザーのデフォルト・アカウントを定義します。

- これはカンマ区切りのアカウントのリストである必要があります。
- アカウント名の後に、各アカウントの権限をカッコ内に指定する必要があります。
- #none エントリの場合は、アカウントが割り当てられていないコンテンツ・アイテムに対する権限が付与されます。#all エントリの場合は、すべてのアカウントに対する権限が付与されます。
- デフォルトは、*#none(RWDA)* です。
- アカウントのリストを文字列で返します。

場所

<*install_dir*>/config/config.cfg

例

構成設定として、デフォルト・アカウント情報を定義します。

```
DefaultAccounts=BOS(R),SEA(RW),MSP/Gen(RWD)
```

スクリプトとして、アカウント情報を文字列で返します。

```
<$DefaultAccounts$>
```

関連項目

- DefaultNetworkAccounts（6-140 ページ）
- ExternalUserAccounts（6-214 ページ）
- SelfRegisteredAccounts（6-725 ページ）

DEFAULTAUTH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Web サーバー（5-42 ページ）
- 外部セキュリティ（5-37 ページ）

説明

デフォルトの認証のチャレンジ・タイプを設定します。

- これは IIS がインストールされている Windows 専用のセキュリティ・オプションです。この設定は、Netscape や Apache Web サーバーでは使用されません。
- デフォルトは、*Basic*（Stellent データベース）です。
- 保護された Stellent リソースにアクセスする場合は、これを *NTLM* に設定して、デフォルトの認証のチャレンジを Microsoft ネットワーク・ログインとして定義します。たとえば、システムにログインしたことのないユーザー宛にセキュリティで保護された URL を電子メールで送信した場合、受信者は、その URL をクリックすると、内部コンテンツ・サーバー・ユーザーではなく、外部（NTLM）ユーザーとしてログインします。
- ユーザーが一度ログインすると、ブラウザを閉じて再起動した後も、ログインでの選択がブラウザに記憶されます。これは、電子メールの URL をクリックすると、使用する認証プロトコルが記憶されることを意味します（Stellent データベースでは「Basic」、Microsoft ネットワークでは「NTLM」）。

場所

<*install_dir*>/data/users/config/filter.hda

例

次は構成エントリの使用例です。

```
DefaultAuth=NTLM
```

関連項目

- NtlmSecurityEnabled（6-555 ページ）
- IntradocRealm（6-369 ページ）

DEFAULTFIELDINCLUDE

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドの表示に使用するインクルードを指定します。

出力

なし。

例

問合せページの非表示フィールドに対するデフォルトのフィールド表示インクルードを *std_query_hidden_field* に設定します。

```
<$if isFieldHidden$>
    <$if isQuery and not (fieldType like "Date") and not (fieldType like "Int")$>
        <$defaultFieldInclude = "std_query_hidden_field"$>
    <$endif$>
```

関連項目

- fieldCaptionInclude（6-217 ページ）
- fieldEntryInclude（6-221 ページ）

DEFAULTFILTERINPUTFORMAT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ローカライゼーション（5-29 ページ）
- Inbound Refinery（5-11 ページ）
- Dynamic Converter（5-17 ページ）

説明

変換フィルタ（Dynamic Converter や Text Indexer Filter コンポーネントなど）のデフォルトの入力キャラクタ・セットを定義します。

- 変換用の製品では、システムの文字コードを使用して、変換フィルタへの入力のコードを決定します。ほとんどの場合は、これで十分です。ただし、この値の上書きが必要になる場合もあります。
- コード値の完全なリストは、*<install_dir>/shared/config/resources* ディレクトリにある *std_encoding.htm* ファイルの `lcIsoEncoding` 値を参照してください。
- デフォルトは、*Cp1252* です。

場所

<install_dir>/config/config.cfg

例

次は日本語テキストの構成エントリとしての使用例です。

```
DefaultFilterInputFormat=shift_jis
```

次は日本語テキストの構成エントリとしての使用例です。

```
DefaultFilterInputFormat=euc-kr
```

関連項目

– DefaultFilterOutputFormat（6-134 ページ）

DEFAULTFILTEROUTPUTFORMAT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ローカライゼーション（5-29 ページ）
- Inbound Refinery（5-11 ページ）
- Dynamic Converter（5-17 ページ）

説明

変換フィルタ（Dynamic Converter や Text Indexer Filter コンポーネントなど）のデフォルトの出力キャラクタ・セットを定義します。

- 変換用の製品では、システムの文字コードを使用して、変換フィルタの出力のコードを決定します。ほとんどの場合は、これで十分です。ただし、この値の上書きが必要になる場合もあります。
- コード値の完全なリストは、*<install_dir>/shared/config/resources* ディレクトリにある *std_encoding.htm* ファイルの `lcIsoEncoding` 値を参照してください。
- デフォルトは、*Cp1252* です。

場所

<install_dir>/config/config.cfg

例

次は日本語テキストの構成エントリとしての使用例です。

```
DefaultFilterOutputFormat=shift_jis
```

次は日本語テキストの構成エントリとしての使用例です。

```
DefaultFilterOutputFormat=euc-kr
```

関連項目

– DefaultFilterInputFormat（6-133 ページ）

DEFAULTGRAPHICTIMEOUT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Inbound Refinery（5-11 ページ）

説明

デフォルトのサムネイル・タイムアウトを定義します。

- サムネイルのタイムアウト値を返します。
- この変数の後に、次のいずれかのフラグが必要です。

フラグ	説明
#factor	この値とファイル・サイズを掛け合せて、サムネイル変換の許容時間を決定します。（結果の時間は、#max および #min フラグによって制限されます。）デフォルト値は、3 です。ビジーまたは処理の遅いシステムでは、この値を増加します。
#max	サムネイル変換処理が完了するまでの Inbound Refinery の最大待機時間を秒数で指定します。デフォルトは、60 秒です。
#min	サムネイル変換処理が完了するまでの Inbound Refinery の最小待機時間を秒数で指定します。デフォルトは、5 秒です。

場所

- 「Shared Inbound Refinery Configuration」画面→「Timeout Values」タブ→「Default Timeout for Graphics」
- <*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
DefaultGraphicTimeout#factor=3
```

DEFAULTHTMLCONVERSION

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Dynamic Converter（5-17 ページ）

説明

Dynamic Converter 変換のデフォルト・テンプレートを定義します。

- この値は、デフォルトとして指定するチェックイン済のテンプレート・ファイルのコンテンツ ID である必要があります。
- デフォルト値はありません。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
DefaultHtmlConversion=PLAIN
```

DEFAULTHTMLREFRESHTIMEOUTINSECONDS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

「Work In Progress」ページ、「My Checked-Out Content」ページおよび「My Workflow Assignments」ページのリフレッシュ時間（秒）を定義します。

この変数に加えて、URL またはサービスのデータバインダに、`htmlRefreshTimeout` を設定できます。これを使用して、ページによって異なるリフレッシュ時間を設定できます。これが必要な場合は、`DefaultHtmlRefreshTimeoutInSeconds` を config.cfg ファイルに設定せず、かわりに、`htmlRefreshTimeout` を URL またはデータバインダに設定します。

`htmlRefreshUrl` は、ページのリフレッシュ後にロードするページの URL に設定できます。デフォルトは、現在のページです。この変数は、ページの URL またはデータバインダに設定できます。

`DefaultHtmlRefreshTimeoutInSeconds` のデフォルト値は、60 です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
DefaultHtmlRefreshTimeoutInSeconds=120
```

関連項目

- htmlRefreshTimeout（6-297 ページ）
- htmlRefreshUrl（6-298 ページ）

DEFAULTMASTERDOMAIN

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Web サーバー（5-42 ページ）
- 外部セキュリティ（5-37 ページ）

説明

Web サーバー・フィルタのデフォルトのマスター・ドメインを定義します。

- 指定されたドメインからの Windows グループは、ロールに変換される前に、名前に DOMAINNAME¥ 接頭辞が付加されません。
- ドメインを指定せずにユーザーがログインした場合は、デフォルトのマスター・ドメインであると推測されます。
- デフォルトは、Web サーバーのホストである Windows Server マシンのドメインです。
- 指定されたドメイン名を文字列で返します。
- この変数を <none> に設定すると、ユーザー名の接頭辞として値は付加されません。値を設定する場合は、大カッコを使用する必要があります。

場所

- 「Configure NT Domain Security」ページ→「Default Master Domain」
- 「Active Directory Configuration」ページ→「Default Master Domain」
- <*install_dir*>/data/users/config/filter.hda

例

次は構成エントリの使用例です。

```
DefaultMasterDomain=CORP_DOMAIN
```

DEFAULTNATIVETIMEOUT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Inbound Refinery（5-11 ページ）

説明

デフォルトのネイティブ・ファイルのタイムアウトを定義します。

- ネイティブ・ファイルのタイムアウト値を返します。
- この変数の後に、次のいずれかのフラグが必要です。

フラグ	説明
#factor	この値とファイル・サイズを掛け合せて、ネイティブから後処理スクリプトへの変換の許容時間を決定します。（結果の時間は、#max および #min フラグによって制限されます。）デフォルト値は、3 です。ビジーまたは処理の遅いシステムでは、この値を増加します。
#max	ネイティブから後処理スクリプトへの変換処理が完了するまでの Inbound Refinery の最大待機時間を秒数で指定します。デフォルトは、10 秒です。
#min	ネイティブから後処理スクリプトへの変換処理が完了するまでの Inbound Refinery の最小待機時間を秒数で指定します。デフォルトは、2 秒です。

場所

- 「Shared Inbound Refinery Configuration」画面→「Timeout Values」タブ→「Default Native to Postscript Timeout」
- <*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
DefaultNativeTimeout#factor=3
```

DEFAULTNETWORKACCOUNTS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

外部ユーザー・ベース（NTLM、Active Directory または LDAP）に定義されたユーザーのデフォルトのアカウント権限を定義します。

- これはカンマ区切りのアカウントのリストである必要があります。

重要 : アカウントを区切るカンマの前後に空白を含めないでください。たとえば、`BOS(R),MSP(RW)` は使用できますが、`BOS(R) , MSP(RW)` は使用できません。

- 各アカウントの権限は、アカウント名の後ろのカッコ内に指定できます。権限が指定されていない場合は、デフォルトで RWDA 権限が付与されます。
- `#none` エントリの場合は、アカウントが割り当てられていないコンテンツ・アイテムに対する権限が付与されます。`#all` エントリの場合は、すべてのアカウントに対する権限が付与されます。
- デフォルトは、*#none(RWDA)* です。
- この設定は、匿名ユーザーには適用されません。6-130 ページの「Default Accounts」を参照してください。
- この設定は、最低限のアカウント権限を定義します。外部ユーザー・ベースで定義されたアカウント権限は、これらの権限に追加されます。たとえば、デフォルトが *#none(RW),Project(R)* で、ユーザーのグループが *Project(RWD)* 権限にマップされている場合、ユーザーの権限は *#none(RW),Project(RWD)* です。
- アカウントのリストを文字列で返します。

場所

NTLM または Active Directory Security

- 「Configure NT Domain Security」ページ→「Default Network Accounts」
- 「Active Directory Configuration」ページ→「Default Network Accounts」
- <*install_dir*>/data/users/config/filter.hda

LDAP セキュリティ

❖ 「Add/Edit LDAP Provider」ページ→「Default Network Accounts」

❖ <*install_dir*>/data/providers/*provider_name*/provider.hda

例

次は構成エントリの使用例です。

```
DefaultNetworkAccounts=#none,guest_account

DefaultNetworkAccounts=BOS(R),MSP/Gen(R),#none(RW)
```

関連項目

– Default Accounts（6-130 ページ）
– ExternalUserAccounts（6-214 ページ）
– SelfRegisteredAccounts（6-725 ページ）
– AcctPermDelim（6-5 ページ）

DEFAULTOPTIONLISTSCRIPT

タイプおよび使用方法

❖ その他のフィールド表示変数（4-29 ページ）

❖ フィールド表示（5-24 ページ）

説明

標準オプション・リスト・フィールドを表示する Idoc スクリプトのピースを定義します。

出力

なし。

例

optList（6-572 ページ）関数を使用してオプション・リストを生成します。

```
<$if optionsAllowPreselect and fieldValue$>
    <$defaultOptionListScript = "<$optList " & optionListName & ":fieldValue$>"$>
<$else$>
    <$defaultOptionListScript = "<$optList " & optionListName & ":noselected$>"$>
<$endif$>
```

現在のオプション・リストの ResultSet に対するループ処理によってオプション・リストを生成します。

```
<@dynamichtml compute_resultset_option_list_script@>
<$if not optionListKey$>
    <$optionListKey = fieldName$>
<$endif$>
<$defaultOptionListScript = "<$loop " & optionListResultSet & "$>" &
"<$inc('std_resultset_option_list_item')$>" & "<$endloop$>"$>
<@end@>
```

関連項目

– optionListScript（6-568 ページ）

– optList（6-572 ページ）

DEFAULTPASSWORDENCODING

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

ユーザーのパスワードを格納するときに使用するエンコーティングのタイプを定義します。

- デフォルトでは、Secure Hash Algorithm 1（SHA1）が使用されます。
- オープン・テキストのパスワードが必要な場合、推奨される値は `OpenText` です。
- 他の空でない値が割り当てられている場合、エンコーディングは実行されません。
- デフォルト値は、`SHA1-CB` です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
DefaultPasswordEncoding=OpenText
```

DEFAULTPOSTSCRIPTTIMEOUT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

デフォルトの後処理スクリプトのタイムアウトを定義します。

- デフォルトの後処理スクリプトのタイムアウト値を定義します。
- この変数の後に、次のいずれかのフラグが必要です。

フラグ	説明
#factor	この値とファイル・サイズを掛け合せて、後処理スクリプトから PDF への変換の許容時間を決定します。（結果の時間は、#max および #min フラグによって制限されます。）デフォルト値は、4 です。ビジーまたは処理の遅いシステムでは、この値を増加します。
#max	後処理スクリプトから PDF への変換処理が完了するまでの Inbound Refinery の最大待機時間を秒数で指定します。デフォルトは、2 秒です。
#min	後処理スクリプトから PDF への変換処理が完了するまでの Inbound Refinery の最小待機時間を秒数で指定します。デフォルトは、20 秒です。

場所

- 「Shared Inbound Refinery Configuration」画面→「Timeout Values」タブ→「Default Postscript to PDF Timeout」
- <*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
DefaultPostscriptTimeout#factor=4
```

DELETEACTIONITEMINTERVAL

タイプおよび使用方法

- Collaboration Manager の構成変数（4-4 ページ）
- Collaboration Manager（5-5 ページ）

説明

フォルダまたはコンテンツ・アイテムを削除するとき、ブラウザが Content Server からのレスポンスをチェックするまでの待機時間をミリ秒単位で指定します。

- この設定は、伝播でのプログレス・バーの更新間隔の設定にも使用されます。
- デフォルトは、300 です。
- Collaboration Manager のインストール中に追加されます。

場所

<*install_dir*>/<*instancename*>/data/components/CollaborationManager/config.cfg

例

```
deleteActionItemInterval=200
```

関連項目

- actionItemInterval（6-9 ページ）
- deleteWatchdogInterval（6-146 ページ）
- watchdogInterval（6-907 ページ）

DELETEWATCHDOGINTERVAL

タイプおよび使用方法

- Collaboration Manager の構成変数（4-4 ページ）
- Collaboration Manager（5-5 ページ）

説明

フォルダまたはコンテンツ・アイテムを削除するとき、ブラウザが Content Server からのレスポンスを待機する時間をミリ秒単位で指定します。指定された時間内にレスポンスを受信しなかった場合は、サーバーが応答していないことを示す警告メッセージが表示されます。

- デフォルトは、300,000 です。
- Collaboration Manager のインストール中に追加されます。

場所

<*install_dir*>/<*instancename*>/data/components/CollaborationManager/config.cfg

例

```
deleteWatchdogInterval=200000
```

関連項目

- actionItemInterval（6-9 ページ）
- deleteActionItemInterval（6-145 ページ）
- watchdogInterval（6-907 ページ）

DELIMITEDUSERROLES

タイプおよび使用方法

- ユーザー読取り専用変数（4-33 ページ）
- 内部セキュリティ（5-36 ページ）
- ユーザー（5-40 ページ）

説明

現在のユーザーが所属するロールを、カンマ区切りのコロン・デリミタ付きのロールのリストとして取得します。

出力

ユーザーのロール・リストを文字列で返します。

例

次のように書式設定されたロールのリストを返します。

```
:guest:,:PublicContributor:,:ClassifiedConsumer:
```

関連項目

– UserRoles（6-867 ページ）

DIRECTORYLOCKINGLOGPATH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- デバッグ（5-20 ページ）

説明

ディレクトリを一時的にロックする際にログを有効化し、ログ・ファイルのパスを定義します。

- ディレクトリの一時的ロック中に発生したエラーは、このログ・ファイルに報告されます。
- この変数の値が設定されていない場合、ログは無効化されます。
- デフォルト値はありません。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
DirectoryLockingLogPath=C:/temp/locking.log
```

DISABLEAMBERLAYOUTS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

トレイおよびトップ・メニューのレイアウト・ベースの外観を無効にします。

デフォルト値は、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

```
DisableAmberLayouts=true
```

DISABLECLASSICLAYOUTPUBLISHING

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

Web レイアウトまたは静的ファイルを公開する際に Classic レイアウトを使用不可にします。<*install_dir*>/shared/config/resources/std_resources.htm ファイルで使用されます。

場所

<*install_dir*>/config/config.cfg

例

```
DisableClassicLayoutPublishing=true
```

DISABLECOLLEGIATESKINPUBLISHING

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

Web レイアウトまたは静的ファイルを公開する際に Collegiate スキンを使用不可にします。<*install_dir*>/shared/config/resources/std_resources.htm ファイルで使用されます。

場所

<*install_dir*>/config/config.cfg

例

```
DisableCollegiateSkinPublishing=true
```

DISABLECONTENTPROFILES

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・プロファイル（5-7 ページ）

説明

コンテンツ・プロファイル機能を無効化するために使用します。ただし、すでに作成済のプロファイルは削除されません。プロファイルは、ユーザー・インタフェースを通じて引き続き使用できます。

この構成変数を使用して、コンテンツ・プロファイルまたは別のコンポーネントが特定の動作の原因であるかどうかを確認できます。

デバッグの完了後、コンテンツ・プロファイルを再び使用するには、この変数を config.cfg ファイルから削除し、コンテンツ・サーバーを再起動します。

場所

<*install_dir*>/config/config.cfg

例

```
DisableContentProfiles=true
```

DISABLEDBLONGINTEGERFIELD

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

MSDE ソフトウェアの旧バージョンでは、4 バイトの整数を使用できました。Content Server の新バージョンでは、8 バイトの整数およびそれ以上が使用されます。この変数により、Content Server で 4 バイトの整数を引き続き使用できます。

MSDE を使用する Content Server 6.2 を Content Server 7.0 以上のリリースにアップグレードすると、dRevRank 列が追加されるため、アップグレードされたインスタンスは起動しません。Content Server 6.2 では、Content Server の新バージョンよりも古い MSDE バージョンを使用しています。この変数を TRUE に設定して追加すると、コンテンツ・サーバーが起動します。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
DisableDBLongIntegerField=true
```

DISABLEFORCEDCONVERSIONS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Dynamic Converter（5-17 ページ）

説明

Dynamic Converter での強制変換および事前変換を無効化するかどうかを定義します。

- TRUE に設定すると、動的に変換されたレンディションをユーザーが参照する場合にのみ Dynamic Converter によってファイルが変換されます。チェックインでの変換（強制および事前）は、発生しません。
- この設定は、TRUE に設定した後にチェックインされた新しいファイルに適用されます。DisableKeepingForcedConversions（6-157 ページ）設定が有効な場合を除き、既存の強制変換および事前変換は引き続き使用されます。
- この設定は、次のような場合に役立ちます。
  - サーバーのキャッシュ領域または変換の処理容量が制限されている場合
  - Web ページとして表示されるファイルが非常に少ないため、動的に変換する必要のあるファイルが比較的少ない場合
  - オンザフライの変換での遅延時間が許容される場合
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
DisableForcedConversions=true
```

関連項目

- DisableKeepingForcedConversions（6-157 ページ）
- ForcedConversionRules（6-236 ページ）

DISABLEGZIPCOMPRESSION

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 変換（5-11 ページ）

説明

HTML ページおよびログ・ファイルの gzip 圧縮を無効化します。クライアントのブラウザが圧縮済 HTML をサポートする場合にのみページを圧縮します。圧縮によるページの平均縮小率は 80% で、HTTP 1.1 と互換性のある任意のブラウザでサポートされます。

このエントリを使用しないかぎり、FILTER_DEBUG からの出力は gzip 形式です。

- TRUE に設定すると、ページは、クライアントの Web ブラウザに送信される前に圧縮されません。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
DisableGzipCompression=true
```

関連項目

- CGI_RECEIVE_DUMP（6-65 ページ）
- CGI_SEND_DUMP（6-66 ページ）
- FILTER_DEBUG（6-234 ページ）
- HTTP_ACCEPT_ENCODING（6-300 ページ）

DISABLEHTTPUPLOADCHUNKING

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

アップロードのチャンク化機能（Upload アプレットまたは HTTP プロバイダで使用）を無効化します。

- TRUE に設定した場合、チャンク化機能は無効です。
- FALSE に設定した場合、チャンク化機能は有効です。
- デフォルトは、TRUE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
DisableHttpUploadChunking=false
```

関連項目

- AppletChunkSize（6-39 ページ）
- AppletChunkThreshold（6-40 ページ）
- ChunkedRequestTrace（6-70 ページ）

DISABLEKEEPINGFORCEDCONVERSIONS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Dynamic Converter（5-17 ページ）

説明

Dynamic Converter が、更新のために変換済ファイルを強制変換として処理するかどうかを定義します。

- TRUE に設定すると、強制または事前変換によって変換されたファイルは、Dynamic Converter によって強制変換とはみなされません。したがって、Dynamic Converter では、通常はファイルを再変換せずに既存の変換を使用している場合でも、既存の変換は強制変換として認識されず、ファイルは再変換されます。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
DisableKeepingForcedConversions=true
```

関連項目

– DisableForcedConversions（6-154 ページ）

DISABLEMEETINGCENTER

タイプおよび使用方法

- Collaboration Manager の構成変数（4-4 ページ）
- Collaboration Manager（5-5 ページ）

説明

設定すると、Collaboration Manager のオンライン会議機能が無効になります。

- この変数が設定されていないか、0 に設定した場合、オンライン会議は有効です。
- Collaboration Manager のインストール中に追加されます。

場所

<*install_dir*>/<*instancename*>/data/components/CollaborationManager/config.cfg

例

```
DisableMeetingCenter=1
```

DISABLEOPTIMIZEDLATESTRELEASEQUERYING

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Dynamic Converter（5-17 ページ）

説明

Dynamic Converter が再変換の必要なファイルを判定するとき、キャッシュされたタイムスタンプを使用するか、コンテンツ・サーバーのデータベース情報を使用するかを定義します。

- TRUE に設定すると、Dynamic Converter は、キャッシュされたタイムスタンプではなく、コンテンツ・サーバーのデータベースへの問合せによって再変換の必要なファイルを判断します。
- この設定を有効にすると、6.2 より前の Dynamic Converter の動作に戻ります。この設定は、最適化された問合せ機能（EnableOptimizedLatestReleaseQuerying（6-198 ページ））で問題が発生した場合、または関連のある問題のトラブルシューティングが必要な場合に使用できます。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
DisableOptimizedLatestReleaseQuerying=true
```

関連項目

– EnableOptimizedLatestReleaseQuerying（6-198 ページ）

DISABLESHAREDCACHECHECKING

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）
- クラスタリング（5-5 ページ）

説明

ロードされたすべてのリソースについてタイムスタンプのチェックを無効化します。

- TRUE に設定すると、コンポーネントに定義されたリソース・ファイルは、コンテンツ・サーバーの再起動の際にのみリフレッシュされます。
- この設定によって、HCST および HCSP ページでのタイムスタンプ・チェックは停止されません。
- 通常、コンテンツ・サーバーのファイル・システムがネットワーク共有にあるとき、この設定を使用してファイル・システムのアクティビティを制限できます。この設定は、特に、ファイルのチェックやタイムスタンプのオーバーヘッドによってパフォーマンスが著しく低下する可能性のある本番環境で重要です。開発環境では、この設定はオプションにすることができます。
- 詳細は、クラスタリングのドキュメントを参照してください。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
DisableSharedCacheChecking=true
```

DISABLESTELLENT05SKINPUBLISHING

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

Web レイアウトまたは静的ファイルを公開する際に Stellent05 スキンを使用不可にします。<*install_dir*>/shared/config/resources/std_resources.htm ファイルで使用されます。

場所

<*install_dir*>/config/config.cfg

例

```
DisableStellent05SkinPublishing=true
```

DISABLESTELLENTSKINPUBLISHING

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

Web レイアウトまたは静的ファイルを公開する際に Stellent スキンを使用不可にします。<*install_dir*>/shared/config/resources/std_resources.htm ファイルで使用されます。

場所

<*install_dir*>/config/config.cfg

例

```
DisableStellentSkinPublishing=true
```

DISABLETOPMENUSPUBLISHING

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

Web レイアウトまたは静的ファイルを公開する際に Top Menu レイアウトを使用不可にします。<*install_dir*>/shared/config/resources/std_resources.htm ファイルで使用されます。

場所

<*install_dir*>/config/config.cfg

例

```
DisableTopMenusPublishing=true
```

DISABLETRAYSPUBLISHING

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

Web レイアウトまたは静的ファイルを公開する際に Trays レイアウトを使用不可にします。<*install_dir*>/shared/config/resources/std_resources.htm ファイルで使用されます。

場所

<*install_dir*>/config/config.cfg

例

```
DisableTraysPublishing=true
```

DISABLEWINDOWSPUBLISHING

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

Web レイアウトまたは静的ファイルを公開する際に Trays レイアウトを使用不可にします。<*install_dir*>/shared/config/resources/std_resources.htm ファイルで使用されます。

場所

<*install_dir*>/config/config.cfg

例

```
DisableTraysPublishing=true
```

DISPLAYPSORIGIN

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）
- デバッグ（5-20 ページ）

説明

変換された PDF ファイルに、*www.stellent.com* へのハイパーリンクを設定します。デバッグのために使用されます。

- TRUE に設定すると、変換された PDF ファイルの（(0,0),(25,25)）に、*www.stellent.com* へのハイパーリンクが設定されます。
- デフォルトは、FALSE です。

場所

<*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
DisplayPSOrigin=true
```

DISTILLERNORMJOBSETTING

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

Adobe Acrobat Distiller ジョブ・オプションの標準（最適化されない）PDF 変換へのパスが設定されます。

- 文字列としてパスを返します。
- PDF Converter のインストール中に Adobe Acrobat Distiller が検出された場合、デフォルトは、*<IBR_install_dir>/shared/AdobeDistillerJobOptions/Acrobat5.joboptions* です。
- PDF Converterのインストール中にAdobe Acrobat Distillerが検出されなかった場合、デフォルトはありません。

場所

- 「Shared Inbound Refinery Configuration」画面→「Paths」タブ→「Distiller JobOptions File Path for Normal Distilling」
- *<install_dir>*/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
DistillerNormJobSetting=C:/stellent/IdcRefinery/shared/
AdobeDistillerJobOptions/IntradocPDF.joboptions
```

関連項目

- UseAdobeDistiller（6-830 ページ）
- DistillerOptJobSetting（6-168 ページ）

DISTILLEROPTJOBSETTING

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

Adobe Acrobat Distiller ジョブ・オプションの最適化された PDF 変換へのパスが設定されます。

- 文字列としてパスを返します。
- PDF Converter のインストール中に Adobe Acrobat Distiller が検出された場合、デフォルトは、*<IBR_install_dir>/shared/AdobeDistillerJobOptions/Acrobat5FastWebInternalThumbnail.joboptions* です。
- PDF Converterのインストール中にAdobe Acrobat Distillerが検出されなかった場合、デフォルトはありません。

場所

- 「Shared Inbound Refinery Configuration」画面→「Paths」タブ→「Distiller JobOptions File Path for Optimized Distilling」
- <*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
DistillerOptJobSetting=C:/stellent/IdcRefinery/shared/
AdobeDistillerJobOptions/IntradocPDFOptimized.joboptions
```

関連項目

- UseAdobeDistiller（6-830 ページ）
- DistillerNormJobSetting（6-167 ページ）

DISTILLERPATH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

ファイルを PDF に変換するために PDF Converter が使用する実行ファイルのパスを設定します。

- 文字列としてパスを返します。
- PDF Converterのインストール中にAdobe Acrobat Distillerが検出されなかった場合、デフォルトは、Jaws PDF Library のパスです。
- PDF Converter のインストール中に Adobe Acrobat Distiller が検出された場合、デフォルトは、Distiller 実行ファイルのパスです。

場所

- 「Shared Inbound Refinery Configuration」画面→「Paths」タブ→「Adobe Distiller Executable File Path」
- <*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
DistillerPath=C:/Program Files/IdcRefinery/shared/jaws_ps/ps2pdf.exe
```

関連項目

- UseAdobeDistiller（6-830 ページ）
- DistillerNormJobSetting（6-167 ページ）
- DistillerOptJobSetting（6-168 ページ）

DoAutoMaxMerge

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）

説明

Verity Integration で使用されます。Verity Integration は、Content Server とともに使用できるアドオン製品であり、購入することができます。

Verity Indexer の自動チューンアップ機能を有効にします。

- 自動チューンアップ機能は、計算された間隔で Verity Indexer の実行にクリーンアップの構築パラメータを自動的に追加します（6-11 ページの「AdditionalIndexBuildParams」パラメータを参照）。これにより、検索の効率および索引付けのパフォーマンスが向上します。
- TRUE に設定すると、自動チューンアップ機能が有効になります。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
DoAutoMaxMerge=true
```

関連項目

- AdditionalIndexBuildParams（6-11 ページ）
- AutoMaxMergeTrace（6-50 ページ）
- MaxMergeBaseCount（6-522 ページ）

DocConverterEngineDir

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Inbound Refinery（5-11 ページ）

説明

Inbound Refinery 変換エンジンの実行可能ファイルの場所を定義します。

- 文字列としてパスを返します。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
DocConverterEngineDir=c:/stellent/IdcRefinery/ConversionEngines/
```

DOCLOADRESOURCEINCLUDES

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ページ表示（5-30 ページ）
- Idoc スクリプト（5-27 ページ）

説明

指定されたコンテンツ・アイテムに、現在のページの表示に使用されるすべてのインクルードをロードします。

- 指定されたコンテンツ・アイテムのファイル拡張子は *idoc* である必要があります。
- この関数により、*StatusCode* が二次的に設定されます。指定されたファイルを正常にロードしてページを正しく表示する必要がある場合は、abortToErrorPage（6-2 ページ）関数を使用します。

パラメータ

コンテンツ・サーバーにチェックインされるコンテンツ・アイテムを指定する、CGI エンコードされたパラメータ・リストを取ります。パラメータ・オプションを次の表に示します。

注意： *.hcsp* ページで使用する場合は、CGI エンコードされたパラメータ・リスト中のアンパサンド（&）文字を `&` 文字に変更する必要があります。

オプション・パラメータ	説明
dID	*dID* が指定されていない場合は、*dDocName* および *RevisionSelectionMethod* を指定する必要があります。*RevisionSelectionMethod* パラメータが指定されていないか、値が *Specific* の場合、この ID を持つコンテンツ・アイテムのリビジョンのレンディションが返されます（存在する場合）。
dDocName	*dDocName* がわかっている場合は、コンテンツ・アイテムのすべてのリクエストで、*dDocName* を指定することをお薦めします。このパラメータは、エラー・メッセージ（およびフォームなどのその他の機能）で、指定されていることが前提となっています。 • *dDocName* が指定されていない場合は、*dID* を指定し、*RevisionSelectionMethod* は指定しないでください。 • *RevisionSelectionMethod* が指定されている場合は、この名前を持つコンテンツ・アイテムのリビジョンのレンディションが返されます（存在する場合）。 • *RevisionSelectionMethod* が指定されていない場合は、エラー・メッセージで *dDocName* が使用される場合があります。
RevisionSelectionMethod	このパラメータを指定する場合は、*dDocName* を指定する必要があります。この変数の値は、指定された *dDocName* から *dID* を計算するための方法です。値は、*Specific*、*Latest* または *LatestReleased* です。 • **Specific:** *dDocName* は無視され、*dID* が必要となり、特定のリビジョンを取得するために使用されます。 • **Latest:** コンテンツ・アイテムの最新のリビジョン（ワークフロー内のリビジョンを含む）を使用して *dID* を計算します。 • **LatestReleased:** コンテンツ・アイテムの最も最近リリースされたリビジョンを使用して *dID* を計算します。
Rendition	• 指定されていない場合、*Rendition* のデフォルトは *Primary* です。このパラメータは、コンテンツ・アイテムのレンディションを指定します。 • 値が *Primary*、*Web* または *Alternate* の場合、選択されたリビジョンのプライマリ、Web で表示可能または代替のレンディションが返されます。

出力

なし。

例

mydoc の最新リビジョンのプライマリ・ボールト・レンディションに、リソースのインクルードをロードします。

```
<$docLoadResourceIncludes("dDocName=mydoc&RevisionSelectionMethod=Latest")$>
```

関連項目

– abortToErrorPage（6-2 ページ）

DOCROOTFILENAME

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

拡張子またはディレクトリ・パスを付けずに、ファイルのファイル名を取得します。

- 通常、この関数は、コンテンツ・サーバーが制御する静的 URL のコンテンツ ID（*dDocName*）部分を抽出するために使用されます。

パラメータ

1 つのパラメータを取ります。

- パラメータは、パスおよびファイル名のみです。

出力

ファイル名を文字列で返します。

例

値 *mydoc* を返す例を次に示します。

```
docRootFilename("/groups/public/documents/adacct/mydoc.pdf")
```

DOCTYPESELECTED

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

現在のコンテンツ・アイテムのタイプが、アクティブな ResultSet のタイプと一致するかどうかを評価します。

出力

- コンテンツ・アイテムのタイプが一致する場合は、TRUE を返します。
- コンテンツ・アイテムのタイプが一致しない場合は、FALSE を返します。

例

コンテンツ・アイテムのタイプが ResultSet のタイプと一致するかどうかに基づいて値を返します。

```
<$DocTypeSelected$>
```

DOCURL

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

weblayout ディレクトリにあるファイルの URL を取得します。

出力

ファイルの URL を文字列として返します。

例

コンテンツ・アイテムへの URL リンクの作成に使用されます。

```
<$if HasUrl$>
    <a href="<$DocUrl$>"><$dDocName$></a>
<$else$>
    <$dDocName$>
<$endif$>
```

DOCURLALLOWDISCLOSURE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 内部セキュリティ（5-36 ページ）

説明

現在のユーザーに対して URL を公開できるかどうかを評価します。

パラメータ

1 つのパラメータを取ります。

- パラメータは、絶対パス（*http://mycomputer/stellent/groups/.../documents/mydoc.pdf* など）または完全相対パス（*/stellent/groups/.../documents/mydoc.pdf* など）のみです。

出力

ブール値を返します。

- URL を公開できる場合は、TRUE を返します。
- URL が制限されている場合は、FALSE を返します。

例

ユーザーが *mydoc.pdf* ドキュメントの URL を表示できるかどうかを判断します。

```
<$docUrlAllowDisclosure("/stellent/groups/documents/mydoc.pdf")$>
```

DoDocNameOrder

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

Repository Manager 内のコンテンツ・アイテムを *dDocName* でソートします。Repository Manager が起動すると、デフォルトの問合せがデータベースに対して実行されます。デフォルトでは、問合せ結果は、アイテムの dDocName でソートされます。dDocName でソートすると、予測どおりの順序になりますが、問合せ結果が長くなる可能性があります。

- TRUE に設定すると、コンテンツ・アイテムは *dDocName* でソートされます。
- FALSE に設定すると、コンテンツ・アイテムはソートされません。
- デフォルトは、TRUE です。

JDBC Query Trace を有効にしてコンソール・ログに情報をログし、order by 句が問合せに含まれていないことを確認できます。JDBC Query Trace を有効にするには、「System Audit Information」ページの「Edit Active Console Output Tracing」セクションで、`systemdatabase` をオプションとして選択します。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
DoDocNameOrder=false
```

DOMAINCONTROLLERNAME

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Web サーバー（5-42 ページ）
- 外部セキュリティ（5-37 ページ）

説明

NTLM コールに使用するドメイン・コントローラを定義します。

- 値が設定されていない場合は、API コールを行って通信するドメイン・コントローラを見つけます。
- デフォルトは、空の文字列です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
DomainControllerName=PDC servername
```

DOWNLOADAPPLET

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）
- クライアント（5-4 ページ）

説明

複数ファイルの Download アプレットを有効にします。

- TRUE に設定すると、Download アプレットが有効化され、検索結果ページから複数のファイルをダウンロードできるようになります。
- FALSE に設定した場合、Download アプレットは無効です。
- デフォルトは、FALSE です。

場所

- 「System Properties」→「Options」タブ→「Enable download applet」
- 「Admin Server」→「General Configuration」→「Enable download applet」
- <*install_dir*>/config/config.cfg

例

構成設定として、Download アプレットを有効にします。

```
DownloadApplet=true
```

スクリプトとして、Download アプレットの状態を評価します。

```
<$DownloadApplet$>
```

関連項目

- MultiUpload（6-539 ページ）
- UploadApplet（6-826 ページ）

DOWNLOADSUGGESTEDNAME

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・アイテム（5-6 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

ダウンロードするファイルのデフォルト・パスと推奨される名前を取得します。

- 推奨される名前は、コンテンツ・アイテムの元のファイル名がベースとなります。

出力

ダウンロードされるファイルのパスと推奨される名前を文字列として返します。

例

ダウンロードされるファイルのパスと推奨される名前を返します。

```
<$DownloadSuggestedName$>
```

DPGET

タイプおよび使用方法

❖ グローバル関数（4-21 ページ）

❖ コンテンツ・プロファイル（5-7 ページ）

説明

dpSet 関数によって設定された値を返します。

dpGet および dpSet は、表示されるコンテンツ・プロファイルのルールのロジックを制御するために使用されます。通常、ユーザーはアクティブ化条件を作成して、指定されたキーの値を設定します。条件が false として評価された場合でも（つまり、ルールが起動されない場合でも）、キーは設定されます。これにより、別のルールがキーにアクセスし、アクティブ化条件を評価するときに値を取得できます。ユーザーはこれらの 2 つの関数を使用して、ルール間の依存関係を作成できます。たとえば、前のルールが評価済である場合にのみ、ルールを評価することもできます。

パラメータ

この関数は、1 つのパラメータを取ります。

❖ Key: 返される値を指定します。

例

```
<$myValue = dpGet("myKey")$>
```

関連項目

– dpSet（6-186 ページ）

DPPROMOTE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- コンテンツ・プロファイル（5-7 ページ）

説明

この関数は、値をルール評価コンテキストからリクエスト・コンテキストに昇格します。すべてのルールを評価した後、この関数によって指定されたキーおよび値は、ローカル・データにプッシュされます。

パラメータ

この関数は、2 つのパラメータを取ります。

- 最初のパラメータは、返される値を指定します。
- 2 番目のパラメータは、返される値です。

例

次の例では、1 つのリクエストの AutoNumberPrefix の値を、dDocType の値に変更します。

```
<$dpPromote("AutoNumberPrefix", dDocType)$>
```

次の例では、チェックイン・ページで Title フィールドを非表示にします。

```
<$dpPromote("dDocTitle:isHidden", "1")$>
```

関連項目

– dpPromoteRs（6-185 ページ）

dpPromoteRs

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- コンテンツ・プロファイル（5-7 ページ）

説明

この関数は、dpPromote と同様に、プロファイル・ルールで生成される ResultSet をリクエスト・コンテキストに昇格できます。すべてのルールを評価した後、ResultSet はデータバインダにプッシュされます。値は、ルールの起動または操作中に評価されませんが、ページの表示に利用できます。

パラメータ

この関数は、2 つのパラメータを取ることができます。

- 最初のパラメータは、データにプッシュされる ResultSet の名前です。
- オプション。2 番目のパラメータは、ResultSet の代替名です。

関連項目

– dpPromote（6-184 ページ）

DPSET

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- コンテンツ・プロファイル（5-7 ページ）

説明

この関数は、Idoc スクリプトが評価される任意の場所で、アクティブ化条件またはルール値を評価する際に一時的な値を設定します。

値は、作業領域内に設定され、dpGet 関数コールによって取得できます。設定されるキーおよび値は保存されませんが、評価されているプロファイルでグローバルに使用できます。すべてのキー / 値のペアは、プロファイルのルールによってアクセスできます。

dpGet および dpSet は、表示されるコンテンツ・プロファイルのルールのロジックを制御するために使用されます。通常、ユーザーはアクティブ化条件を作成して、指定されたキーの値を設定します。条件が false として評価された場合でも（つまり、ルールが起動されない場合でも）、キーは設定されます。これにより、別のルールがキーにアクセスし、アクティブ化条件を評価するときに値を取得できます。ユーザーはこれらの 2 つの関数を使用して、ルール間の依存関係を作成できます。たとえば、前のルールが評価済である場合にのみ、ルールを評価することもできます。

パラメータ

この関数は、2 つのパラメータを取ります。

- Key: 返される値を指定します。
- Value: キーに関連付けられる値。

例

```
<$dpSet("myKey", "1")$>
```

関連項目

– dpGet（6-183 ページ）

DTMPORT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- HTML プレビュー（5-26 ページ）

説明

DtmServer（HTML プレビューで必要）が、Oracle Content Publisher クライアントからの接続をリスニングするポート番号を定義します。

- デフォルトは、4441 です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
DtmPort=4442
```

dWfName

タイプおよび使用方法

- ワークフロー変数（4-39 ページ）
- ワークフロー（5-44 ページ）

説明

ワークフローの名前を取得します。

出力

ワークフローの名前を文字列として返します。

例

<*install_dir*>/shared/config/templates/workflow_info.htm テンプレート・ページにある次のコードを使用して、「Workflow Step Information」ページにワークフロー名を表示します。

```
<td align=right><span class=infoLabel><$lc("wwLabelWorkflow")$>
</span></td>
<td><span class=tableEntry><$dWfName$></span></td>
```

dWfStepName

タイプおよび使用方法

- ワークフロー変数（4-39 ページ）
- ワークフロー（5-44 ページ）

説明

ワークフロー内の現在のステップの名前を取得します。

出力

現在のステップの名前を文字列として返します。

例

<*install_dir*>/shared/config/templates/workflow_info.htm テンプレート・ページにある次のコードを使用して、「Workflow Step Information」ページに現在のステップ名を表示します。

```
<td align=right><span class=infoLabel><$lc("wwCurrentStep")$>
</span></td>
<td><span class=tableEntry>
    <$loop WorkflowStep$>
    <$dWfStepName$>
    <$exec RequiredUsers = dWfStepWeight$>
    <$if isTrue(dWfStepIsAll)$><$exec RequiredUsers = 0$><$endif$>
    <$endloop$></span>
</td>
```

DYNAMICCONVERTERHTMLFLAVOR

タイプおよび使用方法

❖ 構成変数（4-3 ページ）

❖ Dynamic Converter（5-17 ページ）

説明

Dynamic Converter が使用する HTML スタイルを定義します。

❖ 次の値を設定できます。

- Internet Explorer 4.0
- Netscape 4.0
- HTML 3.0
- HTML 4.0

❖ デフォルトは、*null* です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
DynamicConverterHtmlFlavor=HTML 4.0
```

EMPTYACCOUNTCHECKINALLOWED

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- 内部セキュリティ（5-36 ページ）

説明

チェックイン・ページでアカウントを指定する必要があるかどうかを確認します。

- 「Standard Page Resources」ページで、アカウントが指定されていない場合にエラー・メッセージを表示するために使用されます。

出力

- アカウント値が必要な場合は、TRUE を返します。
- アカウント値が必要でない場合は、FALSE を返します。

例

アカウント番号が必要かどうかを評価し、エラー・メッセージを表示します。

```
<$if not EmptyAccountCheckinAllowed$>
    <$isRequired = 1, requiredMsg = "Please specify an account."$>
<$endif$>
```

ENABLECOPYCONVERTCOPY

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Inbound Refinery（5-11 ページ）

説明

変換の前に、ネイティブ・ファイルをローカル・ファイル・システムにコピーできます。

- TRUE に設定すると、Inbound Refinery は、ネイティブ・ファイルをローカル・ファイル・システムにコピーし、適用可能な変換ステップを実行し、次に、結果を weblayout ディレクトリにコピーします。
- FALSE に設定すると、ネイティブ・ファイルは、ローカル・ファイル・システムにコピーされません。
- デフォルトは、FALSE です。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
EnableCopyConvertCopy=true
```

ENABLEDOCUMENTHIGHLIGHT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）

説明

PDF、テキストおよび HTML ファイルでのフルテキスト検索用語の強調表示を有効にします。

- TRUE に設定すると、検索用語の強調表示が有効になります。
- FALSE に設定すると、検索用語の強調表示が無効になります。
- Content Server のインストール後のデフォルトは、FALSE です。
- PDF Converter のインストール後のデフォルトは、TRUE です。

場所

- 「System Properties」→「Options」タブ→「Enable search keyword highlighting」
- 「Admin Server」→「General Configuration」→「Enable search keyword highlighting」
- <*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
EnableDocumentHighlight=false
```

スクリプトとして、構成設定の値を返します。

```
<$EnableDocumentHighlight$>
```

関連項目

– UseHtmlOrTextHighlightInfo（6-843 ページ）

ENABLEENVIRONMENTPACKAGER

タイプおよび使用方法

- 構成変数（4-3 ページ）
- デバッグ（5-20 ページ）

説明

Environment Packager デバッグ機能を有効にします。

- この機能により、管理者は、environment.txt というテキスト・ファイルとともに、admin、config および bin ディレクトリにあるすべてのファイルを ZIP アーカイブにパッケージ化できます。このテキスト・ファイルには、Java 仮想マシンおよびサーバーのオペレーティング・システムに関する追加の情報が含まれます。ZIP ファイル *server_environment*.zip は、コンテンツ・サーバーのルート・ディレクトリに置かれます。
- TRUE に設定すると、Environment Packager 機能が有効になります。
- FALSE に設定すると、Environment Packager 機能が無効になります。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
EnableEnvironmentPackager=true
```

ENABLEERRORFILE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- バッチ・ローダー（5-3 ページ）
- デバッグ（5-20 ページ）

説明

バッチ・ローダーのエラー・ファイルを生成します。

- バッチ・ロード中にエラーが検出された場合、問題のあるレコードが追加されたエラー・ファイルが生成されます。
- TRUE に設定すると、エラー・ファイルが生成されます。
- デフォルト値はありません。

場所

<*install_dir*>/bin/intradoc.cfg

例

次は構成エントリの使用例です。

```
EnableErrorFile=true
```

ENABLEEXPIRATIONNOTIFIER

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

コンテンツ・アイテムの有効期限の通知（NOE）を有効にします。これにより、近いうちに期限切れになるドキュメントについて、作成者および管理者に電子メールで通知されます。

- NOE を有効にし、再起動すると、ScheduledEvents <*install_dir*>/*config/activestate.hda* ファイルにエントリが含まれます。
- NOE の実行を制御するインクルードは `set_doevent_for_notification_of_expiration` です。このインクルードでは、後述の NotifyTime パラメータが使用されます。
- 後述のパラメータの NotificationQuery を 30 に設定し（デフォルトは 7）、NotificationIntervalInDays を 7 に設定すると、1 か月以内に期限切れになる対象について週に 1 度の電子メールで通知できます。
- スケジュールされたイベントを定義する表は、<*install_dir*>/*shared/config/resources/std_resources.htm* ファイルにあります。表の名前は IdcScheduledSystemEvents で、行の名前は NotificationOfExpiration です。
- NOE のデバッグ情報を取得するには、次の変数を *config.cfg* ファイルに設定してコンテンツ・サーバーを再起動します。ログはサーバーから出力されます。

```
TraceSectionsList=system,scheduledevents
TraceIsVerbose=true
```

場所

<*install_dir*>/bin/config.cfg

パラメータ

次のオプションのパラメータを取ります。

パラメータ	説明
NotificationQuery	通知の対象となる期限切れドキュメントを見つけるために実行する問合せを指定します。デフォルトは次のとおりです。 `dOutDate > <$dateCurrent()$> <AND> dOutDate < <$dateCurrent(7)$>`
NotificationMaximum	問合せから返される最大ドキュメント数を指定します。デフォルトは、1000 です。
NotifyExtras	電子メール通知を受信する管理ユーザーを指定します。このエントリを使用しない場合、デフォルトは `sysadmin` です。空白のままの場合、追加の通知は送信されません。
NotifyTime	通知を送信する時間。デフォルトは、00:01 です。24 時間表記で指定します。
NotificationFieldNames	通知に含めるフィールド名を指定します。デフォルトは、dDocAuthor、dOutDate、dDocTitle、dDocName、dDocType、dID、dInDate、dRevLabel、dSecurityGroup です。
NOEUsePlainTextEmail	通知にプレーン・テキストの電子メールを使用できます。デフォルトは、false です。電子メール・テンプレートの名前は QUERY_NOTIFICATION です。`NOEUsePlainTextEmail=true` の場合は、`emailFormat=text` です。
NotificationIntervalInDays	間隔を日数で指定します。デフォルトは、1 です。

例

次は構成エントリの使用例です。

```
EnableExpirationNotifier=true
```

ENABLEOPTIMIZEDLATESTRELEASEQUERYING

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Dynamic Converter（5-17 ページ）

説明

コンテンツ・アイテムのメタデータを更新するときに、コンテンツ・アイテムのタイムスタンプを更新するかどうかを定義します。

- タイムスタンプはキャッシュされ、再変換の必要なファイルを判断するために Dynamic Converter によって使用されます。
- TRUE に設定すると、ソース・コンテンツ・アイテムのメタデータを更新するときに、動的に変換される Web で表示可能なすべてのレンディションが、1 秒だけ更新されます。これにより、Dynamic Converter によって変更が認識され、ファイルが再変換されます。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
EnableOptimizedLatestReleaseQuerying=true
```

関連項目

- DisableOptimizedLatestReleaseQuerying（6-159 ページ）
- LatestReleasedTimestampTimeout（6-471 ページ）

ENABLEREBUILDFORCESRECONVERSION

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Dynamic Converter（5-17 ページ）

説明

検索索引を再構築する際に、Dynamic Converter の事前変換および強制変換を再評価するかどうかを定義します。

- TRUE に設定すると、索引の再構築の際に索引付けされるコンテンツ・アイテムは、Dynamic Converter により、強制変換および事前変換の変換ルールに基づいて変換を再評価されます。
- この設定を使用すると、新しい変換ルールを作成した後、索引の再構築の際にファイルの動的な変換が可能になります。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
EnableRebuildForcesReconversion=true
```

関連項目

– ForcedConversionRules（6-236 ページ）

EnableSchemaPublishing

タイプおよび使用方法

- 構成変数（4-3 ページ）
- スキーマ（5-33 ページ）

説明

スキーマの公開を有効または無効にします。

- TRUE に設定すると、公開は有効になります。FALSE に設定すると、公開は無効になります。
- デフォルトは、TRUE です。

警告 : FALSE に設定すると、Configuration Manager Admin Applet で「**Republish Schema**」オプションが使用されている場合でも、公開できません。選択されたノードでのみ Schema Publisher を有効にする必要のあるクラスタでは、通常、この値を FALSE に設定します。

場所

<*install_dir*>/config/config.cfg

例

```
EnableSchemaPublishing=false
```

ENCODEHTML

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- セキュリティ（5-36 ページ）

説明

この Idoc 関数を使用して、無効または破損した HTML 構成のデータ入力をフィルタします。この関数は、Threaded Discussions でデフォルトで使用されます。Threaded Discussions は、以前のコンポーネントであり、現在では、Content Server 8.0 のインストールの際に利用可能です。

`rule` パラメータで `unsafe` 値を使用している場合、Content Server によって受信されるすべての入力データは、既知の安全でないスクリプト・タグのみに適用されます。この機能を変更するには、`HtmlDataInputFilterLevel` 構成変数を使用して実行するフィルタ処理を変更します。

パラメータ

2 つの必須パラメータと 1 つのオプションのパラメータを取ります。

- 最初のパラメータは、エンコードする文字列です。
- 2 番目のパラメータは、HTML 構成をエンコードする際に適用するルールです。次の値を使用できます。
  - `none`: HTML 構成は変換されません。
  - `unsafe`: 既知の安全でないスクリプト・タグのみがエンコードされます。
  - `exceptsafe`: 既知の安全なスクリプト・タグはエンコードされません。
  - `lfexceptsafe`: ユーザーによって拡張コメントが入力され、元のテキストの改行を保持する必要がある場合にお薦めします。

  安全および安全でないスクリプト・タグのコメント処理、属性値、カッコの処理および使用可能な値の詳細は、『セキュリティおよびユーザー・アクセスの管理』を参照してください。
- 空白文字のない長い文字列を改行するかどうか、および適用する最大ワード・サイズを指定するオプションのパラメータもあります。wordbreak または nowordbreak のいずれかを指定します。

出力

エンコードされた文字列を返します。

関連項目

– 使用方法およびすべてのパラメータ値の詳細は、『セキュリティおよびユーザー・アクセスの管理』を参照してください。
– HtmlDataInputFilterLevel（6-295 ページ）

ENTERPRISESEARCHASDEFAULT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）
- Enterprise Search（5-24 ページ）

説明

標準検索ページでの Oracle Enterprise Search を有効にします。

- TRUE に設定すると、標準検索ページに Enterprise Search オプションが表示されます。
- デフォルトは、TRUE です。

場所

- 「System Properties」→「Options」タブ→「Enable Enterprise Search on standard query pages」
- 「Admin Server」→「General Configuration」→「Enable Enterprise Search on standard query pages」
- <*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
EnterpriseSearchAsDefault=true
```

ENTRYCOUNT

タイプおよび使用方法

- ワークフロー変数（4-39 ページ）
- ワークフロー（5-44 ページ）

説明

現在のリビジョンが現在のワークフロー・ステップへのエントリ回数を取得します。

- この変数は、条件文の作成に使用できますが、ハードコードまたは変更することはできません。
- この変数はコンパニオン・ファイルでローカライズされ、キーで維持されます。
  `<step_name>@<workflow_name>.entryCount`

出力

エントリ回数を整数で返します。

例

次のコードは、*MaxEntry* というジャンプを定義し、ステップへの最終エントリから 1 週間を超過している場合、親ワークフローに移動してレビューアに通知します。

```
<$if wfCurrentGet("entryCount")==2$>
    <$wfSet("WfJumpName","MaxEntry")$>
    <$wfSet("WfJumpTargetStep",wfExit(0,0))$>
    <$wfSet("WfJumpEntryNotifyOff","0")$>
<$endif$>
```

EVAL

タイプおよび使用方法

❖ グローバル関数（4-21 ページ）

❖ Idoc スクリプト（5-27 ページ）

説明

変数の定義を Idoc スクリプトとして評価します。

❖ リテラル文字列を再帰的に評価できます。

パラメータ

1 つのパラメータを取ります。

❖ パラメータは、Idoc スクリプトとして評価される文字列を定義する変数のみです。

出力

変数定義の Idoc スクリプトが解決されたパラメータ文字列を返します。

例

変数 *one* に文字列 CompanyName を割り当て、変数 *two* には、Idoc スクリプトのデリミタで囲まれた変数 *one* を含む文字列を割り当てます。ページで、変数 *one* は文字列 CompanyName を表し、変数 *two* は文字列 Welcome to <one> を表し、*eval(two)* は文字列 Welcome to CompanyName を表します。

```
<$one="CompanyName"$>
<$two="Welcome to <$one$>"$>
<$one$><br>
<$two$><br>
<$eval(two)$>
```

フィールド名に接頭辞 *xPrefix* を持つすべてのメタデータ・フィールドを除外するとします。この場合、*eval* 関数を使用して、この目的を果す Idoc スクリプトを動的に作成して評価できます。

```
<$loop DocMetaDefinition$>
<$if strIndexOf("xPrefix", dName) >= 0$>
    <$myScript = "<$" & dName & ":isExcluded=1$>"$>
    <$eval(myScript)$>
<$endif$>
<$endloop$>
```

関連項目

– キーワードと関数の対比（3-19 ページ）
– exec（6-210 ページ）
– setResourceInclude（6-737 ページ）

EXCLUDECONVERSIONFILTER

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Inbound Refinery（5-11 ページ）

説明

Inbound Refinery によって処理されない変換形式のリストを指定します。

- これはカンマ区切りのリストです。
- 形式は、Configuration Manager の「File Formats」タブの「Conversion」列にリストされます。
- デフォルトは、空の文字列です。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
ExcludeConversionFilter=FrameMaker,MSPub
```

EXCLUSIVECHECKOUT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 内部セキュリティ（5-36 ページ）
- ユーザー（5-40 ページ）

説明

別のユーザーによって作成されたコンテンツをチェックアウトできるかどうかを判断します。

- TRUE に設定すると、作成者またはセキュリティ・グループに対する管理権限を持つユーザーのみがコンテンツ・アイテムをチェックアウトできます。
- FALSE に設定すると、セキュリティ・グループに対する書込み権限を持つユーザーは、別のユーザーによって作成されたコンテンツをチェックアウトできます。
- デフォルトは、FALSE です。

場所

- 「System Properties」→「Content Security」タブ→「Allow only original contributor to check out」
- 「Admin Server」→「Content Security」→「Allow only original contributor to check out」
- <*install_dir*>/config/config.cfg

例

次の構成設定では、オリジナル・コントリビュータのみがコンテンツ・アイテムをチェックアウトできます。

```
ExclusiveCheckout=true
```

スクリプトとして、構成設定の値を返します。

```
<$ExclusiveCheckout$>
```

関連項目

– isUserOverrideSet（6-450 ページ）

EXCLUSIVEINCLUDECONVERSIONFILTER

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Inbound Refinery（5-11 ページ）

説明

Inbound Refinery によって処理される変換形式のリストを指定します。

- リストにない変換形式は、Inbound Refinery によって処理されません。
- これはカンマ区切りのリストです。
- 形式は、Configuration Manager の「File Formats」タブの「Conversion」列にリストされます。
- デフォルトは、空の文字列です。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
ExclusiveIncludeConversionFilter=FrameMaker,MSPub
```

EXEC

タイプおよび使用方法

- 特殊キーワード（3-18 ページ）
- Idoc スクリプト（5-27 ページ）

説明

Idoc スクリプト式を実行し、出力を抑制します（ページに式を表示しません）。

- Idoc スクリプトの旧バージョンでは、*exec* キーワードは、出力ファイルでの変数値の表示を抑制するために必要でした。現行バージョンでは、*exec* キーワードは、出力で式の表示を抑制するためにのみ必要です。

  たとえば、次の最初の 1 行は、最後の 2 行と等価です。

  ```
  <$varA="stringA", varB ="stringB"$>
  <$exec varA="stringA"$>
  <$exec varB="stringB"$>
  ```

- 通常、*exec* キーワードは、「背後にある」コードを評価するために使用されます（たとえば、後からページで使用されるインクルードの指定など）。

パラメータ

1 つのパラメータを取ります。

- パラメータは、Idoc スクリプト式のみです。

出力

式の値を返しますが、ページに式は表示されません。

例

inc グローバル関数を評価して、ページにコードを表示せずに、指定されたインクルードを組み込みます。

```
<@dynamichtml std_definitions@>
    <$exec inc("std_page_variable_definitions")$>
    <$exec inc("define_image_files")$>
<@end@>
```

関連項目

– キーワードと関数の対比（3-19 ページ）
– eval（6-205 ページ）

EXECUTESERVICE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- Idoc スクリプト（5-27 ページ）

説明

コンテンツ・サーバー・サービスを実行します。

- この関数では、ページを構築している間に指定されたサービスを実行できます。通常、サービスは、ブラウザで IdcCommand または CGI URL などのツールを使用して実行されます。
- 動的サーバー・ページで使用されます。
- `executeService` 関数でコールできるサービスは、パラメータ入力が必要でないという意味で、「スクリプト可能」である必要があります。スクリプト可能なサービスのアクセス・レベルは 32 以上です。

パラメータ

1 つのパラメータを取ります。

- パラメータは、実行するサービスの名前のみです。

注意 : DataBinder のライブ・データは、指定されたサービスのパラメータとして使用されます。

出力

- サービスが正常に実行された場合、`0` の StatusCode（6-756 ページ）を返します。
- サービスの実行中にエラーが発生した場合は、`-1` の StatusCode（6-756 ページ）を返します。

注意 : 実行されたサービスのすべての出力は抑制されますが、ResultSet およびロードされた値は使用できます。

例

サービス名を指定すると、サービスを実行します。

```
<$executeService("servicename")$>
```

関連項目

– StatusCode（6-756 ページ）

EXPIREREVISIONONLY

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

指定された有効期限に、コンテンツ・アイテムのすべてのリビジョンが期限切れになるのか、特定のリビジョンのみかを決定します。

注意：この変数は、Content Server 6.2 以上に適用されます。6.2 より前のバージョンでは、特定のリビジョンのみが指定された有効期限で期限切れになりました。バージョン 6.2 以降では、デフォルトの動作は、現在のリビジョンが期限切れになると、そのコンテンツ・アイテムのすべてのリビジョンが期限切れになります。この変数を TRUE に設定すると、コンテンツ・サーバーは、6.2 以前の動作に戻ります。

- TRUE に設定すると、現在のリビジョンのみが期限切れになり、最新の期限切れでないリビジョンが現在のリビジョンになります。
- FALSE に設定すると、現在のリビジョンが期限切れになると、すべてのリビジョンが期限切れになります。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
ExpireRevisionOnly=true
```

EXTERNALUSERACCOUNTS

タイプおよび使用方法

- ユーザー読取り専用変数（4-33 ページ）
- 外部セキュリティ（5-37 ページ）
- ユーザー（5-40 ページ）

説明

外部ユーザー・ベース（NTLM、Active Directory または LDAP）で定義されたユーザーのデフォルト・ロールを取得します。

出力

- 各アカウントの権限がアカウント名の後ろのカッコ内に指定された、カンマ区切りのアカウントのリストを返します。
- `#none` エントリの場合は、アカウントが割り当てられていないコンテンツ・アイテムに対する権限が付与されます。`#all` エントリの場合は、すべてのアカウントに対する権限が付与されます。

例

スクリプトとして、デフォルトのアカウント情報を文字列で返します。

```
<$ExternalUserAccounts$>
```

関連項目

- ExternalUserRoles（6-215 ページ）
- UserAccounts（6-849 ページ）
- Default Accounts（6-130 ページ）
- DefaultNetworkAccounts（6-140 ページ）

EXTERNALUSERROLES

タイプおよび使用方法

- ユーザー読取り専用変数（4-33 ページ）
- 外部セキュリティ（5-37 ページ）
- ユーザー（5-40 ページ）

説明

外部ユーザー・ベース（NTLM、Active Directory または LDAP）で定義されたユーザーのデフォルト・ロールを取得します。

出力

ロールのリストを文字列で返します。

例

スクリプトとして、デフォルトのロール情報を文字列で返します。

```
<$ExternalUserRoles$>
```

関連項目

- ExternalUserAccounts（6-214 ページ）
- UserRoles（6-867 ページ）

FIELDCAPTION

タイプおよび使用方法

- フィールド情報変数（4-26 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドのキャプション・ラベルを指定します。

出力

メタデータ・フィールドのキャプションを文字列として返します。

例

dDocTitle メタデータ・フィールドのキャプションをローカライズされた文字列 *wwTitle* として定義します。

```
<$fieldName = "dDocTitle", fieldCaption = lc("wwTitle"), isRequired = 1,
fieldType = "BigText", requiredMsg = lc("wwSpecifyTitle")$>
```

注意：詳細は、『Customizing Content Server』のローカライズされた文字列の解決に関する項を参照してください。

FIELDCAPTIONINCLUDE

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドのキャプションの表示に使用するインクルードを指定します。

出力

なし。

例

キャプションの表示のインクルードを *std_field_caption* に設定します。

```
<$fieldCaptionInclude = "std_field_caption"$>
```

関連項目

- defaultFieldInclude（6-132 ページ）
- fieldEntryInclude（6-221 ページ）
- fieldCaptionStyle（6-218 ページ）

FIELDCAPTIONSTYLE

タイプおよび使用方法

❖ その他のフィールド表示変数（4-29 ページ）

❖ フィールド表示（5-24 ページ）

説明

メタデータ・フィールドのキャプションのスタイルを指定します。

❖ キャプションで一般的に使用される SPAN スタイルを次に示します。これらは *<install_dir>*/shared/config/resources/std_page.htm リソース・ファイルの *std_style_declaration* インクルードに定義されています。

- searchLabel
- infoLabel
- tableEntry
- requiredField

出力

キャプションのスタイル名を返します。

例

キャプションのスタイルを *requiredField* に設定します。

```
<$if isRequired and not suppressCaptionHighlighting$>
    <$fieldCaptionStyle = "requiredField"$>
<$endif$>
```

標準のフィールド・キャプションを生成します。

```
<@dynamichtml std_field_caption@>
    <span class=<$fieldCaptionStyle$>><$fieldCaption$>
    <$if not isFormSubmit$><$":"$><$endif$></span>
<@end@>
```

関連項目

– fieldCaption（6-216 ページ）

– fieldCaptionInclude（6-217 ページ）

– fieldValueStyle（6-230 ページ）

– NotationForRequiredFields（6-548 ページ）

– StyleForRequiredFields（6-783 ページ）

FIELDDEFAULT

タイプおよび使用方法

- フィールド情報変数（4-26 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドのデフォルト値を指定します。

出力

メタデータ・フィールドのデフォルト値を文字列として返します。

例

dDocTitle メタデータ・フィールドのデフォルト値を定義します。

```
<$fieldName = "dDocTitle", fieldCaption = lc("wwTitle"), isRequired = 1,
fieldType = "BigText", requiredMsg = lc("wwSpecifyTitle"),
fieldDefault="Enter a Title"$>
```

FIELDEDITWIDTH

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

HTML ページのメタデータ入力フィールドの文字幅を指定します。この変数は、リソース `compute_std_field_includes` に含まれる `compute_namevalue_edit_widths` に設定されます。

出力

なし。

FIELDENTRYINCLUDE

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドの値の表示に使用するインクルードを指定します。

出力

なし。

例

値の表示のインクルードを *std_checkbox_entry* に設定します。

```
<$fieldEntryInclude = "std_checkbox_entry"$>
```

関連項目

- defaultFieldInclude（6-132 ページ）
- fieldCaptionInclude（6-217 ページ）
- fieldValueStyle（6-230 ページ）

FIELDEXTRASCRIPTINCLUDE

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

特定のメタデータ・フィールドに対する追加の JavaScript 検証を含むインクルード・ファイルの名前を指定します。これはリソース `compute_std_field_includes` に設定されます。

出力

なし。

FIELDINCLUDE

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドを表示するインクルードの名前を指定します。通常、インクルード・ファイルは `std_namevalue_field` ですが、別のインクルードを設定できます。

通常は `xFieldName:include` を使用してページの先頭で設定しますが、リソース `compute_std_field_includes` を上書きして設定することもできます。

出力

なし。

FIELDISOPTIONLIST

タイプおよび使用方法

- フィールド情報変数（4-26 ページ）
- フィールド表示（5-24 ページ）

説明

オプション・リストを持つメタデータ・フィールドかどうかを指定します。

- TRUE に設定されている場合、フィールドにはオプション・リストがあります。
- FALSE に設定されている場合、フィールドにはオプション・リストがありません。
- デフォルトは、FALSE です。

出力

- フィールドのオプション・リストがある場合は、TRUE を返します。
- フィールドのオプション・リストがない場合は、FALSE を返します。

例

事前定義済のアカウントが存在する場合、*dDocAccount* メタデータ・フィールドをオプション・リストとして定義します。

```
<$if HasPredefinedAccounts$>
    <$fieldIsOptionList = 1, optionListName = "docAccounts",
    fieldOptionListType = "combo"$>
<$endif$>
```

関連項目

- hasOptionList（6-285 ページ）
- fieldOptionListType（6-227 ページ）
- optionListName（6-565 ページ）

FIELDMAXLENGTH

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドの最大長を指定します。これはフィールドのサイズではなく、フィールドに含まれる最大文字数です。通常は `xFieldName:include` を使用してページの先頭で設定しますが、リソース `compute_std_field_includes` を上書きして設定することもできます。

出力

なし。

FIELDNAME

タイプおよび使用方法

- フィールド情報変数（4-26 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドの内部名を指定します。

- 事前定義済のメタデータ・フィールドは、小文字の d で始まります（たとえば、*dDocName*）。
- カスタム・メタデータ・フィールドは、小文字の x で始まります（たとえば、*xComments*）。

出力

メタデータ・フィールドの名前を文字列として返します。

例

dDocTitle メタデータ・フィールドを定義します。

```
<$fieldName = "dDocTitle", fieldCaption = lc("wwTitle"), isRequired = 1,
fieldType = "BigText", requiredMsg = lc("wwSpecifyTitle")$>
```

FIELDOPTIONLISTTYPE

タイプおよび使用方法

❖ フィールド情報変数（4-26 ページ）

❖ フィールド表示（5-24 ページ）

説明

メタデータ・フィールドのオプション・リストのタイプを指定します。

❖ この変数に指定できる値を次に示します。

fieldOptionListType の値	対応する Configuration Manager のオプション
strict	Select List Validated または Select List Not Validated
combo	Edit および Select List
multi	Edit および Multiselect List
access	なし（Collaboration Server のメンバー・アクセス・リストで使用）

出力

オプション・リストのタイプを文字列で返します。

例

dDocAccount オプション・リストのタイプを *combo*（Edit および Select List）として定義します。

```
<$if HasPredefinedAccounts$>
    <$fieldIsOptionList = 1, optionListName = "docAccounts",
    fieldOptionListType = "combo"$>
<$endif$>
```

関連項目

– fieldIsOptionList（6-224 ページ）

– optionListName（6-565 ページ）

FIELDTYPE

タイプおよび使用方法

❖ フィールド情報変数（4-26 ページ）

❖ フィールド表示（5-24 ページ）

説明

メタデータ・フィールドのタイプを指定します。

❖ この変数に指定できる値を次に示します。

fieldType の値	対応する Configuration Manager のオプション
Text	Text
BigText	Long Text
Int	Integer
Date	Date
Memo	Memo

出力

メタデータ・フィールドのタイプを文字列で返します。

例

dDocTitle メタデータ・フィールドを *BigText*（Long Text）フィールドとして定義します。

```
<$fieldName = "dDocTitle", fieldCaption = lc("wwTitle"), isRequired = 1,
fieldType = "BigText", requiredMsg = lc("wwSpecifyTitle")$>
```

FIELDVALUE

タイプおよび使用方法

- フィールド情報変数（4-26 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドの値を指定します。

出力

現在のメタデータ・フィールドの値を返します。

例

標準フィールドの値を生成します。

```
<@dynamichtml std_value_label@>
    <span class="<$fieldValueStyle$>"><$fieldValue$></span><!--""-->
<@end@>
```

関連項目

- fieldEntryInclude（6-221 ページ）
- fieldCaptionStyle（6-218 ページ）
- fieldValueStyle（6-230 ページ）
- optionListName（6-565 ページ）

FIELDVALUESTYLE

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドの値のスタイルを指定します。

- 値で一般的に使用される SPAN スタイルを次に示します。これらは <*install_dir*>/shared/config/resources/std_page.htm リソース・ファイルの *std_style_declaration* インクルードに定義されています。
  - tableEntry
  - xxsmall
  - strongHighlight

出力

値のスタイル名を返します。

例

値のスタイルを設定します。

```
<$if isFieldInfoOnly$>
    <$if valueStyle$>
        <$fieldValueStyle = valueStyle$>
    <$else$>
        <$fieldValueStyle = "tableEntry"$>
    <$endif$>
<$endif$>
```

標準フィールドの値を生成します。

```
<@dynamichtml std_value_label@>
    <span class="<$fieldValueStyle$>"><$fieldValue$></span><!--'"-->
<@end@>
```

関連項目

- fieldEntryInclude（6-221 ページ）
- fieldCaptionStyle（6-218 ページ）
- valueStyle（6-897 ページ）

FIELDWIDTH

タイプおよび使用方法

- フィールド情報変数（4-26 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドの幅を文字数で指定します。

出力

現在のメタデータ・フィールドの幅を返します。

例

フィールド幅が 10 文字の *Extension* フィールドを生成します。

```
<$fieldName = "dExtension", fieldCaption = lc("wwNativeExtension"),
fieldWidth = 10$>
```

関連項目

– fieldCaptionStyle（6-218 ページ）

FILEENCODING

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ローカライゼーション（5-29 ページ）
- コンテンツ・サーバー（5-8 ページ）
- Inbound Refinery（5-11 ページ）

説明

使用するファイル・エンコーディングを指定します。

- コンテンツ・サーバーが、オペレーティング・システムのネイティブ言語以外の言語を使用するように構成されている場合（たとえば、英語のシステムで、英語と日本語のロケールを使用する場合）、Content Server および Inbound Refinery の両方で、この変数を **UTF8** に設定する必要があります。

場所

- Content Server: <*install_dir*>/bin/intradoc.cfg
- Inbound Refinery: <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
FileEncoding=UTF8
```

FILEURL

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

現在の動的サーバー・ページ（HCSP または HCST）の相対 URL を取得します。

- 通常、この変数は、自己参照ページ（自己にポスト・バックするフォームなど）で使用されます。

出力

URL を文字列で返します。

例

現在の動的サーバー・ページの相対 URL を返します。

```
<$fileUrl$>
```

FILTER_DEBUG

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Web サーバー（5-42 ページ）
- デバッグ（5-20 ページ）

説明

Web サーバー・フィルタ内で発生するイベントのログを有効にします。

- TRUE に設定すると、Web サーバー・フィルタの認証および認可エラーは、次のログ・ファイルに記録されます。
  - **IIS:** <*install_dir*>/idcplg/idc_cgi_isapi-<*instance*>.dll.log
  - **Apache または NES:** <*install_dir*>/data/users/authfilt.log
- デフォルトは、FALSE です。

場所

<*install_dir*>/data/users/config/filter.hda

例

次は構成エントリの使用例です。

```
FILTER_DEBUG=true
```

関連項目

- CGI_DEBUG（6-64 ページ）
- CGI_RECEIVE_DUMP（6-65 ページ）
- CGI_SEND_DUMP（6-66 ページ）
- PLUGIN_DEBUG（6-584 ページ）
- DisableGzipCompression（6-155 ページ）

FIRSTREV

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

一連の現在のリビジョン・ラベルで、最初のリビジョン・ラベルを返します。

出力

最初のリビジョン・ラベルを文字列で返します。

例

最初のリビジョン・ラベルを返します（デフォルトは 1 です）。

```
<$FIRSTREV$>
```

FORCEDCONVERSIONRULES

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Dynamic Converter（5-17 ページ）

説明

ファイルのチェックイン時に強制変換を適用する Dynamic Converter のルールを定義します。

- これはカンマ区切りのリストです。
- チェックイン時にコンテンツ・アイテムが指定された変換ルールと一致した場合、ファイルはそのルールに従って変換されます。各ファイルは、複数のレンディションに変換できます。
- デフォルトはありません。

注意：強制変換処理では、同じ Dynamic Converter テンプレートが 2 回適用されることはありません。たとえば、`ForcedConversionRules=RuleA,RuleB` を設定し、*RuleA* および *RuleB* が異なるレイアウトを持つ同じテンプレートを指定する場合、*RuleB* による変換は発生しません。（この場合のベスト・プラクティスとしては、レイアウトをマージし、Idoc スクリプトを使用して適切なレイアウト要素を動的に選択します。）

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
ForcedConversionRules=Rule1,Rule2,Rule3
```

関連項目

- DisableForcedConversions（6-154 ページ）
- incDynamicConversionByRule（6-345 ページ）
- rule:IsFragmentOnly（6-404 ページ）

FORCEDISTINCTREVLABELS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

同じコンテンツ・アイテムへの追加リビジョンに、同じリビジョン・ラベルを使用できるかどうかを定義します。

- この設定の変更は、新しいコンテンツのみに適用されます。
- TRUE に設定すると、同じコンテンツの 2 つのリビジョンに同じリビジョン・ラベルを使用できません。
- デフォルトの設定は、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
ForceDistinctRevLabels=true
```

FORCEDOCTYPECHOICE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- フィールド表示（5-24 ページ）

説明

チェックイン・ページの「Type」オプション・リストに空白のオプションをデフォルトとして追加します。これにより、ユーザーはいずれかのタイプを選択する必要があります。

- 表示オプションとしてページで使用されます。
- TRUE に設定すると、チェックイン・ページの「Type」オプション・リストに空白のオプションが表示されます。
- デフォルトの設定は、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
ForceDocTypeChoice=true
```

FORCEEXPIRE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- Idoc スクリプト（5-27 ページ）

説明

この関数を使用して、特定のインクルードのキャッシュを期限切れにすることができます。これは、新規メタデータ・フィールドの追加、新規の問合せの保存、または表示可能なリンクの変更など、ページの表示に変更を加える場合に役立ちます。

includeName の値が NULL または空の文字列の場合は、キャッシュ全体が削除されます。これは、特定のページのすべてのインクルードが同じキャッシュに保存され、同時に失効させる必要がある場合に役立ちます。

詳細は、3-19 ページの「キーワードと関数の対比」を参照してください。

パラメータ

2 つの必須パラメータと 2 つのオプションのパラメータを取ります。次に例を示します。

```
forceExpire(includeName, scope [, cacheName, key])
```

パラメータ	説明
includeName	評価する動的 html 要素の名前。
scope	ユーザーごとに異なる html を表示する場合は session、すべてのユーザーに同じものを表示する場合は application に設定します。
cacheName	このデータをデフォルトのキャッシュではなく、名前付きキャッシュに入れる場合のオプションのパラメータ。空の文字列を渡すと、インクルードは、セッションのデフォルト・キャッシュに入れられます。
key	キャッシュの自動名前スコープによる競合を回避するには、一意のキーをオプションで指定できます。

例

次の例は、std_page_begin インクルードのかわりに使用できます。ユーザーがいずれかの標準サービス・コールを実行して、標準の左ナビゲーション・バーのリンクが変更されたかどうかを確認します（問合せの保存、ポータル・デザインの変更)。変更されている場合、std_page_begin でキャッシュされた html は無効になります。次に、cacheInclude 関数によってインクルードが再評価され、キャッシュに再び保存されます。

```
<$if strEquals(IdcService, "PNE_SAVE_QUERY") or strEquals(IdcService,
"PNE_UPDATE_PORTAL_INFO")$>
<$forceExpire("std_main_page_begin", "session", "", "std_main_page_begin")$>
<$endif$>
<$cacheInclude("std_main_page_begin", "session", 600, "",
"std_main_page_begin")$>
```

関連項目

- inc（6-343 ページ）
- cacheInclude（6-60 ページ）
- setExpires（6-733 ページ）
- setHttpHeader（6-734 ページ）
- setMaxAge（6-735 ページ）

FORCEJDBCSYNC

タイプおよび使用方法

- 構成変数（4-3 ページ）
- データベース（5-18 ページ）
- デバッグ（5-20 ページ）

説明

データベースへのアクセスを一度に 1 つの問合せに制限します。

- 通常、この設定は、デバッグや問題のある JDBC ドライバに対して使用されます。
- コンテンツ・サーバーが、読取り専用タイプのアクティビティで多く使用されている場合、これを FALSE に設定すると、データベースのスループットが改善されます。
- TRUE に設定すると、一度に 1 つの問合せがデータベースにアクセスできます。
- FALSE に設定すると、一度に複数の問合せがデータベースにアクセスできます。
- デフォルトの設定は、FALSE です。（バージョン 4.5 以前のデフォルトは TRUE です。）

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
ForceJdbcSync=true
```

FORCESECURITYGROUPCHOICE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- フィールド表示（5-24 ページ）

説明

チェックイン・ページの「Security Group」オプション・リストに空白のオプションをデフォルトとして追加します。これにより、ユーザーはいずれかのセキュリティ・グループを選択する必要があります。

- 表示オプションとしてページで使用されます。
- TRUE に設定すると、チェックイン・ページの「Security Group」オプション・リストに空白のオプションが表示されます。
- デフォルトの設定は、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
ForceSecurityGroupChoice=true
```

FORMATDATE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 日付と時間（5-19 ページ）

説明

日付 / 時間をデフォルトの日付 / 時間の形式に再フォーマットします。

- データベース書式の日付は評価できません（2001-06-15 など）。
- 長い形式の日付は評価できません（June 15, 2001 など）。

パラメータ

1 つのパラメータを取ります。

- パラメータは、日付 / 時間を指定する文字列のみです。

出力

- dateCurrent（6-121 ページ）で使用される形式の日付 / 時間を返します（6/15/01 1:55 PM など）。
- パラメータを評価できない場合は NULL を返します。
- 時間が指定されていない場合、デフォルトの時間 12:00 AM を返します。

例

次の例では、日付と時間を書式設定し、12/14/99 2:00 PM として表示します。

```
<$formatDate("12/14/1999 02:00 PM")$>
```

次の例では、日付を書式設定し、デフォルトの時間を割り当て、9/15/03 12:00 AM として表示します。

```
<$formatDate("09/15/2003")$>
```

次のスクリプトは、指定された日付と時間を書式設定して表示します。1 行目は、代替の日付と時間の書式を評価し、カスタム変数に割り当てます。2 行目は、この日付をユーザーに表示します。

```
<$my_customDateTime = formatDate("06/14/2004 15:05:34")$>
Final Approval: <$my_customDateTime$>
```

出力 :

Final Approval: 6/14/04 3:05 PM

FORMATDATEDATABASE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 日付と時間（5-19 ページ）
- データベース（5-18 ページ）

説明

SQL 問合せに備えて日付と時間を書式設定します。

- 長い形式の日付は評価できません（May 22, 2000 など）。

パラメータ

1 つのパラメータを取ります。

- パラメータは、日付 / 時間を指定する文字列または変数のみです。

出力

- ODBC 書式の日付 / 時間を返します。
  `[ts 'yyyy-mm-dd hh:mm:ss']`
- パラメータを評価できない場合は NULL を返します。
- 時間が指定されていない場合、デフォルトの時間 00:00:00 を返します。

例

SQL 問合せの現在の日付と時間の書式を設定します。

```
<$formatDateDatabase(dateCurrent())$>
```

日付と時間の書式を設定し、2001-03-19 15:32:00 として表示します。

```
<$formatDateDatabase("03/19/2001 3:32 PM")$>
```

日付と時間の書式を設定し、1999-04-03 00:00:00 として表示します。

```
<$formatDateDatabase("4/3/99")$>
```

FORMATDATEDISPLAY

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 日付と時間（5-19 ページ）

説明

日付 / 時間をユーザーに表示するための日付 / 時間の形式に再フォーマットします。「System Properties」の「Display Date Format」を使用して日付をフォーマットします。

パラメータ

1 つのパラメータを取ります。

- パラメータは、日付 / 時間を指定する文字列のみです。

出力

- 使用する書式で日付 / 時間を返します。
- パラメータを評価できない場合は NULL を返します。

例

```
<$formatDateDisplay$>
```

関連項目

– formatDateOnlyDisplay（6-249 ページ）

FORMATDATEONLY

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 日付と時間（5-19 ページ）

説明

日付 / 時間をデフォルトの日付の形式に再フォーマットし、時間を除外します。

- データベース書式の日付は評価できません（2000-02-02 など）。
- 長い形式の日付は評価できません（June 12, 2001 など）。

パラメータ

1 つのパラメータを取ります。

- パラメータは、日付 / 時間を指定する文字列のみです。

出力

- dateCurrent（6-121 ページ）で使用される形式の日付のみを返します（7/12/00 など）。
- パラメータを評価できない場合は NULL を返します。

例

現在の日付のみを返します（9/12/01 など）。

```
<$formatDateOnly(dateCurrent())$>
```

10 日後の日付を返します（9/22/01 など）。

```
<$formatDateOnly(dateCurrent(10))$>
```

日付と時間の書式を設定し、日付のみを 1/17/00 として表示します。

```
<$formatDateOnly("01/17/2000 2:00 PM")$>
```

次のスクリプトは、現在の日付と 100 日後の日付を表示します。1 行目は、現在の日付のみをカスタム変数に割り当てます。2 行目は、100 日後の日付を 2 番目のカスタム変数に割り当てます。3 行目は、これらの日付をユーザーに表示します（たとえば、Start Date: 10/12/01 and End Date: 1/20/02）。

```
<$my_startDate = formatDateOnly(dateCurrent())$>
<$my_endDate = formatDateOnly(dateCurrent(100))$>
Start Date: <$my_startDate$> and End Date: <$my_endDate$>
```

FORMATDATEONLYDISPLAY

タイプおよび使用方法

❖ グローバル関数（4-21 ページ）

❖ 日付と時間（5-19 ページ）

説明

日付をユーザーに表示するための日付の形式に再フォーマットします。「System Properties」の「Display Date Format」を使用して日付をフォーマットします。

❖ formatDateDisplay に類似していますが、日付のみをフォーマットします。

パラメータ

1 つのパラメータを取ります。

❖ パラメータは、日付を指定する文字列のみです。

出力

❖ 使用する書式で日付を返します。

❖ パラメータを評価できない場合は NULL を返します。

例

```
<$formatDateOnlyDisplay$>
```

関連項目

– formatDateDisplay（6-246 ページ）

FORMATDATEONLYFULL

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 日付と時間（5-19 ページ）

説明

重要：この設定は、バージョン 5.1 以降では非推奨です。formatDateWithPattern（6-251 ページ）を使用します。

日付 / 時間を長い日付の形式に再フォーマットし、時間を除外します。

- データベース書式の日付は評価できません（2000-02-02 など）。

パラメータ

1 つのパラメータを取ります。

- パラメータは、日付 / 時間を指定する文字列のみです。

出力

- 長い形式の日付を返します。
  `month d, yyyy`
- パラメータを評価できない場合は NULL を返します。

例

現在の日付を長い形式で返します。

```
<$formatDateOnlyFull(dateCurrent())$>
```

365 日後の日付を長い形式で返します（たとえば、September 12, 2002）。

```
<$formatDateOnlyFull(dateCurrent(365))$>
```

日付のみの書式を設定し、June 12, 2001 として表示します。

```
<$formatDateOnlyFull("6/12/01 3:00 PM")$>
```

FORMATDATEWITHPATTERN

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 日付と時間（5-19 ページ）

説明

日付 / 時間を指定された日付 / 時間パターンに再フォーマットします。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、コンテンツ・サーバーによって使用される日付文字列、または parseDate（6-578 ページ）または dateCurrent（6-121 ページ）関数で作成された日付オブジェクトです。
- 2 番目のパラメータは、MM/dd/yyyy などの日付 / 時間パターンです。

  大文字の Z は、エントリに UTC タイムゾーンが使用されることを示します。小文字の zzzz は、UTC 時間からの時間オフセット（HHMM）を表し、先頭にプラス（+）またはマイナス（-）を付けてオフセットを示します（Content Server バージョン 8.0 以上で使用可能）。

出力

- パターン・パラメータで指定された形式で日付 / 時間を返します。
- パラメータを評価できない場合は NULL を返します。

例

Wed, 5 Jul 2006 12:08:56 -0700 が表示されます。

```
<$formatDateWithPattern(dateCurrent(),"EEE, d MMM yyyy HH:mm:ss zzzz")$>
```

2006-07-05 14:30:33Z が表示されます。

```
<$formatDateWithPattern(dateCurrent(),"yyyy-MM-dd HH:mm:ssZ")$>
```

関連項目

– 日付の書式設定の詳細は、『Using Content Server in International Environments』を参照してください。

FORMATTIMEONLY

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 日付と時間（5-19 ページ）

説明

日付 / 時間をデフォルトの時間の形式に再フォーマットし、日付を除外します。

パラメータ

1 つのパラメータを取ります。

- パラメータは、日付 / 時間を指定する文字列または変数のみです。

出力

- dateCurrent（6-121 ページ）で使用される形式の時間のみを返します（1:15 PM など）。
- パラメータを評価できない場合は NULL を返します。

例

現在の時間のみを返します。

```
<$formatTimeOnly(dateCurrent())$>
```

時間のみの書式を設定し、5:00 PM として表示します。

```
<$formatTimeOnly("2/2/99 5:00 PM")$>
```

時間のみの書式を設定し、6:14 PM として表示します。

```
<$formatTimeOnly("04/21/2001 18:14:00")$>
```

FORMATTIMEONLYDISPLAY

タイプおよび使用方法

❖ グローバル関数（4-21 ページ）

❖ 日付と時間（5-19 ページ）

説明

時間をユーザーに表示するための時間の形式に再フォーマットします。「System Properties」の「Display Date Format」を使用して時間をフォーマットします。

❖ formatDateDisplay に類似していますが、時間のみを書式設定します。

パラメータ

1 つのパラメータを取ります。

❖ パラメータは、時間を指定する文字列のみです。

出力

❖ 使用する書式で時間を返します。

❖ パラメータを評価できない場合は NULL を返します。

例

```
<$formatTimeOnlyDisplay$>
```

関連項目

– formatDateDisplay（6-246 ページ）

FORMINHERITMETADATAFIELDLIST

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

PDF フォームの後続のリビジョンによって継承されるカスタム・メタデータ・フィールドを定義します。

- これはカンマ区切りのリストです。
- 標準メタデータ・フィールド（「Type」、「Security Group」および「Account」など）は、デフォルトで継承されます。
- カスタム・メタデータ・フィールドは、小文字の x で始まります（たとえば、*xComments*）。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
FormInheritMetaDataFieldList=xDepartment,xRegion
```

FRAMEMAKERCYCLESALLOWED

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

FrameMaker が「応答していない」とみなされるまでに、Inbound Refinery で許容される 30 秒サイクルの最大回数。

- Inbound Refinery では、*consfile.txt* ファイルで FrameMaker のアクティビティを監視します。
- デフォルトは、1 です。

場所

<*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
FrameMakerCyclesAllowed=1
```

FRAMEMAKEREXEPATH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

Adobe FrameMaker 実行可能ファイルのパスを定義します。

- 文字列としてパスを返します。
- デフォルトは、空の文字列です。

場所

- 「Shared Inbound Refinery Configuration」画面→「Paths」タブ→「FrameMaker Executable File Path」
- <*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
FrameMakerexePath=c:/Program Files/Adobe/FrameMaker6.0/FrameMaker.exe
```

FRAMEMAKERTIMEOUT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

FrameMaker 変換のタイムアウトを定義します。

- FrameMaker のタイムアウト値を返します。
- この変数の後に、次のいずれかのフラグが必要です。

フラグ	説明
#factor	この値とファイル・サイズを掛け合せて、FrameMaker 変換の許容時間を決定します。（結果の時間は、#max および #min フラグによって制限されます。）デフォルト値は、1 です。ビジーまたは処理の遅いシステムでは、この値を増加します。
#max	FrameMaker 変換処理が完了するまでの Inbound Refinery の最大待機時間を秒数で指定します。デフォルトは、10 秒です。
#min	FrameMaker 変換処理が完了するまでの Inbound Refinery の最小待機時間を秒数で指定します。デフォルトは、1 秒です。

場所

- 「Shared Inbound Refinery Configuration」画面→「Timeout Values」タブ→「FrameMaker to PostScript Timeout」
- <*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
FrameMakerTimeout#factor=2
```

GATEWAY_INTERFACE

タイプおよび使用方法

- Web サーバー変数（4-36 ページ）
- Web サーバー（5-42 ページ）

説明

このサーバーが準拠する CGI 仕様のリビジョン・レベルを取得します。

- この変数は、特定のリクエストではなく、すべてのリクエストに設定されます。

出力

CGI/revision 形式で、リビジョンのレベルを文字列として返します。

例

次はページまたはログに出力される情報の例です。

```
GATEWAY_INTERFACE=CGI/1.1
```

次は Idoc スクリプト文または評価の一部としての使用例です。

```
<$if GATEWAY_INTERFACE$>
<!--statement-->
```

GENERATEUNIQUEID

タイプおよび使用方法

- ページ変数（4-25 ページ）
- ページ表示（5-30 ページ）

説明

この関数は、HTML ページの一意の ID を返します。HTML 要素の `id` 属性とともに使用して、一意の ID を保証します。

パラメータ

1 つのパラメータを取ります。

- パラメータは、ID が生成されるフィールドの名前のみです。

出力

HTML ページの ID を返します。

GETCOOKIE

タイプおよび使用方法

- 設定可能変数（4-34 ページ）
- Web サーバー（5-42 ページ）

説明

ブラウザから Cookie を取得します。

- 情報をパーソナライズに保存する必要がない場合にユーザーのセッションの追跡に役立ちます。
- 最後にナビゲートしたページ、または最後に実行した検索を追跡するのに使用できます。

パラメータ

この関数は、1 つのパラメータ（Cookie の名前）を取ります。

例

```
<$myCounter=getCookie("myCounter")$>
<$if not myCounter$>
    <$myCounter=1$>
<$endif$>
Num times loaded: <$myCounter$>
<$setCookie("myCounter", myCounter+1, dateCurrent(1))$>
```

関連項目

– setCookie（6-732 ページ）

GETCOPYACCESS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 内部セキュリティ（5-36 ページ）

説明

コンテンツ・アイテムのコピーを取得するためにユーザーに必要な権限を決定します。

- TRUE に設定すると、読取り権限のみを持つユーザーが、コンテンツ・アイテムのコピーを取得できます。
- FALSE に設定すると、コンテンツ・アイテムのコピーを取得するには、コンテンツ・アイテムへの書込み権限が必要です。
- デフォルトは、FALSE です。

場所

- 「System Properties」→「Content Security」タブ→「Allow get copy for user with read privilege」
- 「Admin Server」→「Content Security」→「Allow get copy for user with read privilege」
- <*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
GetCopyAccess=true
```

スクリプトとして、構成設定の値を返します。

```
<$GetCopyAccess$>
```

GETDEBUGTRACE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- デバッグ（5-20 ページ）

説明

デバッグ・トレースの出力を取得します。

パラメータ

なし。

出力

- 構築中のページに蓄積されたデバッグ・トレースの出力を返します。
- ScriptDebugTrace（6-709 ページ）が未設定の場合は、空の文字列を返します。

例

デバッグ・トレースの出力を取得し、ページに情報を出力します。

```
<$getDebugTrace()$>
```

関連項目

– ScriptDebugTrace（6-709 ページ）

GETERRORTRACE

タイプおよび使用方法

❖ グローバル関数（4-21 ページ）

❖ デバッグ（5-20 ページ）

説明

エラー・トレースの出力を取得します。

❖ エラー・トレースの出力は、エンコードされて HTML ページに表示されます。たとえば、< および > デリミタは HTML でエスケープされ、キャリッジ・リターンは <*br*> タグに変換されます。

パラメータ

なし。

出力

❖ 構築中のページに蓄積されたエラー・トレースの出力を返します。

❖ ScriptErrorTrace（6-710 ページ）が未設定の場合は、空の文字列を返します。

例

エラー・トレースの出力を取得し、ページに情報を出力します。

```
<$getErrorTrace()$>
```

関連項目

– ScriptErrorTrace（6-710 ページ）

GETFIELDCONFIGVALUE

タイプおよび使用方法

- 動的変数（4-20 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

この関数は、特定フィールドの構成フラグを返します。フィールドが存在しない場合は、かわりにデフォルトを返します。

パラメータ

パラメータを 3 つ取ります。

- メタデータ・フィールドの名前
- フィールドが見つかった場合に返されるフラグ
- フィールドが存在しない場合に返されるデフォルト値

出力

指定された構成フラグを返します。

例

```
<$caption = getFieldConfigValue("dDocTitle", "fieldCaption", lc("wwTitle"))$>
```

GETFIELDVIEWDISPLAYVALUE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- スキーマ（5-33 ページ）

説明

「Schema」オプション・リストにある項目の表示値を返します。

たとえば、Customer というカスタム・メタデータ・フィールドのオプション・リストが存在するとします。データベースには、CustomerName および一意の CustomerID を持つスキーマ表があります。チェックイン時に、ユーザーに CustomerName が表示されますが、データベースには CustomerID が保存されています。getFieldViewDisplayValue 関数は、CustomerID に基づいて解読可能な CustomerName を抽出します。

パラメータ

パラメータを 3 つ取ります。

- 最初のパラメータは、オプション・リストで使用されるフィールドの名前です。
- 2 番目のパラメータは、現在のフィールドに使用されるスキーマ・ビューの名前です。
- 3 番目のパラメータは、フィールドに割り当てる値です。

出力

表示値を返します。

例

```
<$customerName = getFieldViewDisplayValue("xCustomer", "Customer_View",
"1234")$>
```

関連項目

– Get View Display Value

GETFIELDVIEWVALUE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- コンテンツ・プロファイル（5-7 ページ）

説明

この関数は、フィールドのコンテンツ・ルールを定義するときに使用されます。この関数を使用して、コンテンツ・プロファイルのフィールドの値をビューに依存させることができます。フィールドは、デフォルト値と導出値の両方を持つことが可能であるため、ビューを使用してフィールド間の相互依存性を作成できます。

パラメータ

パラメータを 3 つ取ります。

- フィールド。メタデータ・フィールドの名前。このフィールドに関連付けられたビューが必要です。
- 値。値の参照キー。
- 列名。ビューの表の列。

出力

フィールド値を参照として使用して、指定された列に値を返します。

例

次の例では、xEmployee フィールドのビュー表に、type という名前の列があります。xEmployee メタデータ・フィールドに指定された値を使用して、この関数は、この表への参照を提供し、type 列の値を返します。

```
getFieldViewValue("xEmployee", #active.xEmployee, "type")
```

GETFREEMEMORY

タイプおよび使用方法

❖ グローバル関数（4-21 ページ）

❖ デバッグ（5-20 ページ）

説明

この関数は、Java 仮想マシン（JVM）の空きメモリーの容量を返します。これは「System Audit Information」ページのパフォーマンス監査機能です。

パラメータ

なし。

関連項目

– getTotalMemory（6-273 ページ）

GETHELPPAGE

タイプおよび使用方法

- 動的変数（4-20 ページ）
- ページ表示（5-30 ページ）

説明

この関数は、ページ名に基づいてヘルプ・ページの相対 URL を返します。

パラメータ

1 つのパラメータを取ります。

- 使用するページの名前

例

```
<input type=Button value="<$lc("wwQuickHelp")$>"
    onClick="QuickHelp('<$getHelpPage("QH_AdvancedSearch")$>', 'Search')">
```

GETOPTIONLISTSIZE

タイプおよび使用方法

- 動的変数（4-20 ページ）
- ページ表示（5-30 ページ）

説明

この関数は、オプション・リストのサイズを返します。これを使用して、「User Profile」ページにカスタム・ユーザー・メタデータ・フィールドを表示するかどうかを決定します。

パラメータ

1 つのパラメータを取ります。

- パラメータは、サイズを求めるオプション・リストのみです。

出力

指定されたオプション・リストのサイズを返します。

GETPARENTVALUE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- スキーマ（5-33 ページ）

説明

この関数は、スキーマ・ビューから親値を返します。これは、現在のフィールドの依存オプション・リストを生成する前に必要となります。

ほとんどのオプション・リストは、親フィールドの一意キーに対する依存性をトリガーするため、ほとんどの場合、この関数は内部親値パラメータと同じ値を返します。ただし、これは複合依存選択リストの要件ではありません。

パラメータ

パラメータを 4 つ取ります。

- 現在のフィールドに使用されるスキーマ・ビューの名前。
- このフィールドと親フィールドのスキーマ関係の名前。
- 親フィールドの名前。
- 親フィールドの内部値。これはページに表示される値とは異なる場合があります。多くの場合は、内部に保存された一意の数値です。

例

```
<$parentValue=getParentValue("xState", "Country_State", "xCountry", "123")$>
```

GETREQUIREDMSG

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- コンテンツ・プロファイル（5-7 ページ）

説明

この関数は、必要なエラー・メッセージを Java スクリプトに移入するためにプレゼンテーション・ページで使用されます。エラー・メッセージは、必要であると指定されたフィールドに、コンテンツ・ルールのユーザー・インタフェースから設定されます。次のアルゴリズムに基づいて、必要なメッセージを返します。

1. 関数は、*fieldname*:requiredMsg を探します。
2. 値が存在する場合は、それをローカライズします。
3. 値が未定義であるか、メッセージが空の場合、関数はデフォルトのメッセージを返し、ローカライズしません。メッセージをローカライズするには、この関数（`getRequiredMsg`）にメッセージを渡す前に、関数のコール元がデフォルト・メッセージに `lc` 関数を使用する必要があります。

パラメータ

この関数は、2 つのパラメータを取ります。

- フィールド名 : エラー・メッセージに関連付けられたフィールド
- メッセージ : 使用するデフォルトのメッセージ

関連項目

lc（6-472 ページ）

GETTEXTFILE

タイプおよび使用方法

❖ グローバル関数（4-21 ページ）

❖ Idoc スクリプト（5-27 ページ）

説明

テキスト・ファイルの Web バージョンを取得し、その内容を文字列に返します。

❖ これは、テキスト /html、テキスト / プレーンまたはテキスト /xml など、テキストが含まれる形式を持つファイルでのみ使用できます。

❖ コンテンツ・アイテムの Web URL を決定するための十分な情報が必要です。次の情報が存在する必要があります。

- dDocName、dWebExtension、dSecurityGroup、dRevLabel および dDocType。
- アカウントが有効な場合は、dDocAccount も指定する必要があります。
- この情報は、GET_SEARCH_RESULTS コールまたは DOC_INFO コールの後に自動的に使用可能になります。

❖ パフォーマンスの向上のため、この関数を `cacheInclude` コールでラップします。ただし、すべてのドキュメントが同じキャッシュに保存されると、セキュリティ上の問題が生じる可能性があります。

パラメータ

この関数はパラメータを取らず、ページ上の変数を使用してドキュメントの URL を決定します。

例

```
<$loop SearchResults$>
    <$fullDocument=getTextFile()$>
    <h2><$dDocTitle$></h2>
    <div><$fullDocument$></div>
<$endloop$>
```

関連項目

– cacheInclude（6-60 ページ）

GETTOTALMEMORY

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- デバッグ（5-20 ページ）

説明

この関数は、Java 仮想マシン（JVM）の合計メモリーを返します。これはサーバーのメモリー使用量を確認するために、「System Audit Information」ページで使用されるパフォーマンス監査機能です。

パラメータ

なし。

関連項目

– getFreeMemory（6-267 ページ）

GETUSERVALUE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- Idoc スクリプト（5-27 ページ）
- ユーザー（5-40 ページ）

説明

現在のユーザーのユーザー・メタデータ・フィールド値を取得します。

- パラメータは、Users データベース表の列を参照する必要があります。グローバル参照をサポートしないユーザー・パーソナライズ機能とは異なり、Users 表のユーザーに割り当てられた情報は、コンテンツ・サーバーで使用できます。

パラメータ

1 つのパラメータを取ります。

- パラメータは、ユーザーに関連する変数のみです。

出力

- 現在のユーザーのメタデータ・フィールドの値を返します。
- ユーザーの値を取得した場合は、TRUE になります。
- 取得でエラーが発生した場合は、FALSE になります。
- 値が未指定または不明の場合は、`-1` の StatusCode（6-756 ページ）を返します。

例

現在ログオンしているユーザーのユーザー・タイプを返します。

```
<$getUserValue('dUserType')$>
```

GETVALUE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- Idoc スクリプト（5-27 ページ）
- ResultSet（5-32 ページ）

説明

この関数は、様々な使用方法があります。

- ローカル、アクティブまたは環境データから、特定のメタデータ・フィールドの値を取得します。
- 特定の ResultSet から、特定の列の値を取得します。
- ResultSet の行に関する情報を取得します。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、データ型または ResultSet 名のいずれかです。
- 2 番目のパラメータは、メタデータ・フィールドの名前、列名または ResultSet 行のパラメータです。

多様な形式

省略形を使用して、`getValue(arg1,arg2)` 関数と同じ結果を返すこともできます。これは `<$arg1.arg2$>` の形式をとり、*arg1* および *arg2* は、*getValue* 関数のリテラル文字列引数 arg1 および arg2 です。

注意 : データ型（local、active および env）および ResultSet パラメータは、先頭に # 記号を付加する必要があります。

次のパラメータの組合せを使用できます。

getValue および省略形	説明
getValue("#local",*fieldName*) <$#local.*fieldName*$>	ローカル・データから、指定されたメタデータ・フィールドの値を取得します。
getValue("#active",*fieldName*) <$#active.*fieldName*$>	次の順序で、データから指定されたメタデータ・フィールドの値を取得します。 1. ローカル・データ 2. アクティブな ResultSet 3. 他のすべての ResultSet 4. 環境設定
getValue("#env",*fieldName*) <$#env.*fieldName*$>	環境設定から、指定されたメタデータ・フィールドの値を取得します。
getValue(*ResultSetName*,*fieldName*) <$*ResultSetName*.*fieldName*$>	指定された ResultSet の現在行から、指定されたメタデータ・フィールドの値を取得します。
getValue(*ResultSetName*,*columnName*) <$*ResultSetName*.*columnName*$>	指定された ResultSet の現在行から、指定された列の値を取得します。
getValue(*columnName*) <$*columnName*$>	現在の ResultSet の現在行から、指定された列の値を取得します。
getValue(*ResultSetName*,"#row") <$*ResultSetName*.#row$>	指定された ResultSet の現在行の番号を取得します。 最初の行は 0 です。
getValue(*ResultSetName*,"#numRows") <$*ResultSetName*.#numRows$>	指定された ResultSet の行の総数を取得します。
getValue(*ResultSetName*,"#isRowPresent") <$*ResultSetName*.#isRowPresent$>	指定された ResultSet に、行が 1 つ以上存在するかどうかを確認します。 これは rsNext（6-685 ページ）を使用して手動でループ処理する場合に役立ちます。
getValue(*ResultSetName*,"#isEmpty") <$*ResultSetName*.#isEmpty$>	指定された ResultSet に、行が含まれていないことを確認します。

出力

❖ 出力される値は、6-275 ページの「多様な形式」を参照してください。

❖ 値が見つからない場合は、空の文字列を返します。

例

❖ *DOC_INFO* という ResultSet からコンテンツ ID を取得します。

```
<$name = getValue("DOC_INFO","dDocName")$>
```

または

```
<$name = <$DOC_INFO.dDocName$>
```

❖ 渡されたパラメータ *dDocType*（ローカル・データ）が、アクティブな ResultSet の値と同じかどうかを確認します。

```
<$loop DocTypes$>
<$if strEquals(#active.dDocType, getValue("#local","dDocType"))$>
<!--do special HTML for selected document type-->
<$endif$>
<!-- additional statement-->
<$endloop$>
```

関連項目

– 特殊キーワード（3-18 ページ）

GETVALUEFORSPECIFIEDUSER

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ユーザー（5-40 ページ）
- ワークフロー（5-44 ページ）

説明

特定のユーザーのユーザー属性の値を取得します。

- この関数は、拡張ワークフロー機能の定義に役立ちます。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、ユーザー名です。
- 2 番目のパラメータは、ユーザー属性（*dFullName*、*dUserType* または *dEmail*）を指定する Users データベース表の列名です。

出力

- 指定されたユーザーのユーザー属性の値を返します。
- 値が見つからない場合は、空の文字列を返します。

例

mjones のフルネームを取得します。

```
<$getValueForSpecifiedUser('mjones', 'dFullName')$>
```

GETVIEWVALUE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- スキーマ（5-33 ページ）

説明

「Schema」オプション・リストにある項目の表示値を返します。

たとえば、Customer というカスタム・メタデータ・フィールドのオプション・リストが存在するとします。データベースには、CustomerName および一意の CustomerID を持つスキーマ表があります。チェックイン時に、ユーザーに CustomerName が表示されますが、データベースには CustomerID が保存されています。getFieldViewDisplayValue 関数は、CustomerID に基づいて解読可能な CustomerName を抽出します。

パラメータ

パラメータを 3 つ取ります。

- 現在のフィールドに使用されるスキーマ・ビューの名前
- データベースに格納されている ID の値
- ビューに関連付けられた列の名前

出力

表示値を返します。

例

```
<$custName = getViewValue("Customer_View", "1234", "CustomerName")$>
<$custRegion = getViewValue("Customer_View", "1234", "CustomerRegion")$>
```

関連項目

– getFieldViewDisplayValue（6-265 ページ）

GETVIEWVALUERESULTSET

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- スキーマ（5-33 ページ）

説明

この関数は、スキーマ表をロードし、SchemaData という名前の ResultSet としてページに置きます。ResultSet の列名は、データベース表での名前と同じです。

この関数は、親値に基づいて依存選択肢のリストを取得する場合に最も役立ちます。

パラメータ

パラメータを 3 つ取ります。

- 使用するスキーマ・ビューの名前
- スキーマおよびビューの関係
- スキーマの親の値

例

国名リストのスキーマ表およびビューがあるとします。ビューの名前は Country_view です。

この表の内容をページに出力するには、次のコードを使用します。

```
<$getViewValuesResultSet("Country_View", "", "")$>
<$loop SchemaData$>
<$count = 0, num = rsNumFields("SchemaData")$>
   <$loopwhile count < num$>
      <$fieldName=rsFieldByIndex("SchemaData", count)$>
      <$fieldName$> = <$getValue("SchemaData", fieldName)$>
      <$count = count + 1$>
   <$endloop$>
<$endloop$>
```

これにより、列名が不明な場合でも表が出力されます。

次の例では、Country フィールドおよび State フィールドの DCL があるとします。States のリストは、選択した Country に依存します。Country が USA である場合、States のリストを取得するには、次のコードを使用します。

```
<$getViewValuesResultSet("State_View", "Country_State", "USA")$>
```

HASAPPRIGHTS

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 内部セキュリティ（5-36 ページ）

説明

現在のユーザーが、管理アプリケーションに対する権限を持つかどうかを確認します。

パラメータ

1 つのパラメータを取ります。

- パラメータは、次のいずれかの管理アプリケーションです。
  - UserAdmin
  - WebLayout
  - RepoMan
  - Workflow
  - ConfigMan
  - Archiver

出力

- 指定されたアプリケーションに対する権限がある場合は、TRUE を返します。
- 指定されたアプリケーションに対する権限がない場合は、FALSE を返します。
- 値が未指定または不明の場合は、-1 の StatusCode（6-756 ページ）を返します。

例

現在のユーザーが、指定されたアプリケーションに対する権限のあるユーザーかどうかを評価します。

```
<$hasAppRights('RepoMan')$>
```

HASEXTERNALUSERS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）
- 外部セキュリティ（5-37 ページ）

説明

外部データベースが存在することをコンテンツ・サーバーに示します。

- TRUE に設定すると、コンテンツ・サーバーは外部ユーザーを認識します。外部ユーザーの統合（LDAP など）をサポートするカスタム・コンポーネントが記述されている場合は、TRUE に設定する必要があります。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
HasExternalUsers=true
```

スクリプトとして、構成設定の値を返します。

```
<$HasExternalUsers$>
```

関連項目

– NtlmSecurityEnabled（6-555 ページ）

HASGLOBALUSERS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）
- 外部セキュリティ（5-37 ページ）

説明

コンテンツ・サーバーのグローバル・データベースに、別のコンテンツ・サーバーからのユーザーが存在することを指定します。

- この変数は、複数のマスター・コンテンツ・サーバーが、NTLM、LDAP または Active Directory 以外のシステムからのユーザー・ベースを共有している場合に設定します。
- マスター・コンテンツ・サーバーでは、デフォルトは FALSE です。
- プロキシ・コンテンツ・サーバーでは、デフォルトは TRUE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
HasGlobalUsers=true
```

HASLOCALCOPY

タイプおよび使用方法

- 設定可能変数（4-34 ページ）
- コンテンツ・アイテム（5-6 ページ）
- クライアント（5-4 ページ）

説明

クライアント・マシンのダウンロード先のディレクトリに、リクエストされたコンテンツ・アイテムのコピーが存在するかどうかを確認します。

- 通常、ダウンロードの際に上書きするかどうかをユーザーに問い合せるために使用します。たとえば、この変数は、Oracle ODMA Client で使用されます。

出力

- ローカル・コピーが検出された場合は、TRUE を返します。
- ローカル・コピーが検出されなかった場合は、FALSE を返します。

例

コンテンツ・アイテムのローカル・コピーがあるかどうかを確認します。

```
<$HasLocalCopy$>
```

関連項目

– ClientControlled（6-74 ページ）
– IsNotLatestRev（6-423 ページ）

HASOPTIONLIST

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドにオプション・リストが存在することを指定します。

出力

- フィールドのオプション・リストがある場合は、TRUE を返します。
- フィールドのオプション・リストがない場合は、FALSE を返します。

例

フィールドにオプション・リストが存在することを指定します。

```
<$hasOptionList=1$>
```

フィールドにオプション・リストがある場合は、オプション・リストの値を生成します。

```
<$if hasOptionList$>
    <$if isQuery$></td><td><$endif$>
    <$include std_option_list_entry$>
<$endif$>
```

関連項目

– fieldIsOptionList（6-224 ページ）

HASORIGINAL

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

リビジョンの元のファイルが存在するかどうかを確認します。

- この変数は、*dOriginalName* 値を確認します。
- コンテンツ・アイテムがチェックイン済であっても、元のファイルが存在しない可能性があります（たとえば、基本ワークフローが開始されたが、初期の投稿ステップでファイルがチェックインされていない場合）。

出力

- 元のファイルが存在する場合は、TRUE を返します。
- 元のファイルが存在しない場合は、FALSE を返します。

例

元のファイルがあるかどうかを確認します。

```
<$if HasOriginal and not isNew$>
```

HASPREDEFINEDACCOUNTS

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- 内部セキュリティ（5-36 ページ）
- ユーザー（5-40 ページ）

説明

現在のユーザーが、事前定義済のアカウントに対する権限を持つユーザーかどうかを確認します。

- 事前定義済のアカウントは、ユーザー管理ユーティリティで作成されるアカウントです。

出力

- ユーザーが事前定義済のアカウントに割り当てられている場合は、TRUE を返します。
- ユーザーが事前定義済のアカウントに割り当てられていない場合は、FALSE を返します。

例

ユーザーが事前定義済のアカウントに割り当てられている場合、「Accounts」オプション・リストを表示します。

```
<$if HasPredefinedAccounts$>
   <$fieldIsOptionList = 1, optionListName = "docAccounts",
   fieldOptionListType = "combo"$>
<$endif$>
```

HASSITESTUDIO

タイプおよび使用方法

- Link Manager の構成変数（4-19 ページ）
- 検索（5-34 ページ）

説明

Link Manager 機能とともに使用します。

この関数を使用して Site Studio に固有のリンク・パターンを有効にして、パターン・エンジンのフレンドリ URL を解析します。TRUE に設定すると、この変数により、Site Studio に固有のリンク・パターンが有効化されます。

デフォルトの値は FALSE で、Site Studio に固有のリンク・パターンは無効になります。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
HasSiteStudio=TRUE
```

関連項目

– Link Manager の使用の詳細は、『リポジトリ・コンテンツの管理』を参照してください。

HASURL

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

weblayout ディレクトリに、現在のコンテンツ・アイテムのファイルが存在するかどうかを確認します。

出力

- Web レイアウト・ファイルが存在する場合は、TRUE を返します。
- Web レイアウト・ファイルが存在しない場合は、FALSE を返します。

例

Web レイアウト・ファイルがあるかどうかを確認します。

```
<$if HasUrl$>
    <$include doc_url_field$>
<$endif$>
```

HEAVYCLIENT

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- クライアント（5-4 ページ）

説明

Stellent ODMA クライアントまたは Upload アプレットを使用してチェックインされたかどうかを確認します。

出力

- ODMAクライアントまたはUploadアプレットがチェックインに使用されている場合は、TRUE を返します。
- ODMAクライアントまたはUploadアプレットがチェックインに使用されていない場合は、FALSE を返します。

例

チェックイン方法を確認します。

```
<$HeavyClient$>
```

HELPDIR

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

オンライン・ヘルプ・ファイルが含まれているディレクトリのパスを定義します。

- コンテンツ・サーバーと Inbound Refinery で、このパスを別々に設定できます。
- 文字列としてパスを返します。
- コンテンツ・サーバーのデフォルトは、*<install_dir>*/weblayout/help です。
- Inbound Refinery のデフォルトは、*<SharedDir>*/help です。

場所

- コンテンツ・サーバー : <*install_dir*>/config/config.cfg
- Inbound Refinery: <*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は Content Server の構成エントリとしての使用例です。

```
HelpDir=c:/stellent/weblayout/help/
```

次は Inbound Refinery の構成エントリとしての使用例です。

```
HelpDir=c:/stellent/IdcRefinery/shared/help/
```

HIDEVITALREVIEW

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Record Manager および Retention Manager（5-32 ページ）

説明

TRUE に設定すると、Rentention Manager および Records Manager の検索およびチェックイン・ページで、「Subject to Review」の関連フィールドが非表示になります。FALSE に設定すると、フィールドが表示されます。デフォルトは、TRUE です。

場所

- <*install_dir*>/custom/RecordsManagement/records_management_environments.cfg
- <*install_dir*>/custom/RetentionManager/retentionmanager_environments.cfg

例

次は構成エントリの使用例です。

```
HideVitalReview=false
```

HIGHLIGHTBEGIN

タイプおよび使用方法

❖ 構成変数（4-3 ページ）

❖ Dynamic Converter（5-17 ページ）

説明

Dynamic Converter で変換されるファイルで、テキストの強調表示スタイルの開始 HTML タグを設定します。

❖ デフォルトは、`<strong>` です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
HighlightBegin=<b><font color=blue>
```

関連項目

– HighlightEnd（6-294 ページ）

HIGHLIGHTEND

タイプおよび使用方法

❖ 構成変数（4-3 ページ）

❖ Dynamic Converter（5-17 ページ）

説明

Dynamic Converter で変換されるファイルで、テキストの強調表示スタイルの終了 HTML タグを設定します。

❖ デフォルトは、`</strong>` です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
HighlightEnd=</font></b>
```

関連項目

– HighlightBegin（6-293 ページ）

HTMLDATAINPUTFILTERLEVEL

タイプおよび使用方法

- 構成変数（4-3 ページ）
- セキュリティ（5-36 ページ）

説明

encodeHtml（6-201 ページ）とともに使用する構成エントリです。このエントリは、あるレベルのエンコーディングを適用して、Content Server へのすべての入力で不正な HTML 構成をフィルタ処理します。

`encodeHtml rule` パラメータで `unsafe` 値を使用している場合、Content Server によって受信されるすべての入力データは、既知の安全でないスクリプト・タグのみに適用されます。この機能を変更するには、`HtmlDataInputFilterLevel` 構成変数を使用して実行するフィルタ処理を変更します。

パラメータ

このエントリは、1 つのパラメータ（フィルタ・レベル）を取ります。次の値に設定できます。

- none: フィルタリングは実行されません。
- unsafe: 不正な HTML 構成から保護されます。
- exceptsafe: フィルタを通じて既知の安全な構成のみが許可されます。

場所

<*install_dir*>/config/config.cfg

関連項目

– encodeHtml（6-201 ページ）
–『セキュリティおよびユーザー・アクセスの管理』

HTMLEDITORPATH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

コンポーネント・ウィザードからコンポーネント・ファイルの編集に使用される実行可能ファイルのパスを定義します。

- このパスは、コンポーネント・ウィザードで定義されます。
- グラフィカル HTML エディタ（FrontPage など）ではなく、テキスト・エディタ（WordPad など）を指定します。グラフィカル・エディタを使用すると、HTML タグが挿入または変更されて、Idoc スクリプトのタグがコンテンツ・サーバーで認識されない文字列に変換される可能性があります。
- デフォルト値はありません。

場所

- <*install_dir*>/config/config.cfg
- <*install_dir*>/bin/intradoc.cfg

例

次は構成エントリの使用例です。

```
HTMLEditorPath=C:/Program Files/Windows NT/Accessories/wordpad.exe
```

HTMLREFRESHTIMEOUT

タイプおよび使用方法

- 値変数（4-34 ページ）
- ページ表示（5-30 ページ）

説明

DefaultHtmlRefreshTimeoutInSeconds に類似しています。「Work In Progress」ページ、「My Checked-Out Content」ページまたは「My Workflow Assignments」ページのリフレッシュ時間（秒）を定義します。

htmlRefreshTimeout は、URL またはサービスのデータバインダに設定できます。これを使用して、ページによって異なるリフレッシュ時間を設定できます。これが必要な場合は、DefaultHtmlRefreshTimeoutInSeconds を config.cfg ファイルに設定せず、かわりに、htmlRefreshTimeout を URL またはデータバインダに設定します。

例

ページの URL での設定の例を次に示します。

```
&htmlRefreshTimeout=90
```

関連項目

- DefaultHtmlRefreshTimeoutInSeconds（6-137 ページ）
- htmlRefreshUrl（6-298 ページ）

HTMLREFRESHURL

タイプおよび使用方法

- 値変数（4-34 ページ）
- ページ表示（5-30 ページ）

説明

「Work In Progress」ページ、「My Checked-Out Content」ページまたは「My Workflow Assignments」ページのリフレッシュ後にロードするページの URL の設定に使用されます。デフォルトは、現在のページです。この変数は、ページの URL またはサービスのデータバインダに設定できます。

例

ページの URL での設定の例を次に示します。

```
&htmlRefreshUrl=http://www.home.com
```

関連項目

- DefaultHtmlRefreshTimeoutInSeconds（6-137 ページ）
- htmlRefreshTimeout（6-297 ページ）

HTTP_ACCEPT

タイプおよび使用方法

- Web サーバー変数（4-36 ページ）
- Web サーバー（5-42 ページ）

説明

ブラウザが受信できるコンテンツ・タイプのリストを取得します。

出力

コンテンツ・タイプのリストをカンマ区切りの文字列として返します。

例

次はページまたはログに出力される情報の例です。

```
HTTP_ACCEPT=text/xml,application/xml,application/xhtml+xml,
text/html;q=0.9,text/plain;q=0.8,video/x-mng,image/png,image/jpeg,
image/gif;q=0.2,text/css,*/*;q=0.1
```

次は Idoc スクリプト文または評価の一部としての使用例です。

```
<$if HTTP_ACCEPT$>
<!--statement-->
```

HTTP_ACCEPT_ENCODING

タイプおよび使用方法

- Web サーバー変数（4-36 ページ）
- Web サーバー（5-42 ページ）

説明

ブラウザがサポートする圧縮のエンコーディングのリストを取得します。

- バージョン 7.0 では、HTML ファイルが配信のために圧縮されます。デフォルトは、GZIP 圧縮です。

出力

圧縮のエンコーディングのリストをカンマ区切りの文字列として返します。

例

次はページまたはログに出力される情報の例です。

```
HTTP_ACCEPT_ENCODING=gzip, deflate, compress;q=0.9
```

次は Idoc スクリプト文または評価の一部としての使用例です。

```
<$if HTTP_ACCEPT_ENCODING$>
<!--statement-->
```

HTTP_ACCEPT_LANGUAGE

タイプおよび使用方法

- Web サーバー変数（4-36 ページ）
- Web サーバー（5-42 ページ）

説明

ブラウザに設定された ISO 言語のリストを取得します。

- この変数は、新規またはゲスト・ユーザーに設定するユーザー・ロケールの予測に役立ちます。

出力

言語のリストをカンマ区切りの文字列として返します。

例

次はページまたはログに出力される情報の例です。

```
HTTP_ACCEPT_LANGUAGE=en-us,en;q=0.50
```

次は Idoc スクリプト文または評価の一部としての使用例です。

```
<$if HTTP_ACCEPT_LANGUAGE$>
<!--statement-->
```

HTTP_COOKIE

タイプおよび使用方法

- Web サーバー変数（4-36 ページ）
- Web サーバー（5-42 ページ）

説明

HTTP リクエスト・ヘッダーで Cookie の名前 / 値のペアを取得します。

出力

`name1=string1; name2=string2` の形式で文字列を返します。

例

次はページまたはログに出力される情報の例です。

```
HTTP_COOKIE=IntradocAuth=Basic; IntrdocLoginState=1
```

次は Idoc スクリプト文または評価の一部としての使用例です。

```
<$if HTTP_COOKIE$>
<!--statement-->
```

HTTP_HOST

タイプおよび使用方法

- Web サーバー変数（4-36 ページ）
- Web サーバー（5-42 ページ）

説明

Web サーバーの名前を取得します。

出力

Web サーバー名を文字列で返します。

例

次はページまたはログに出力される情報の例です。

```
HTTP_HOST=centralserver
```

次は Idoc スクリプト文または評価の一部としての使用例です。

```
<$if HTTP_HOST$>
<!--statement-->
```

HTTP_INTERNETUSER

タイプおよび使用方法

- Web サーバー変数（4-36 ページ）
- Web サーバー（5-42 ページ）

説明

コンテンツ・サーバーがユーザーを設定できるように、Web サーバー・セキュリティ・フィルタがコンテンツ・サーバーに渡す CGI パラメータを取得します。

- この変数が設定されていない場合は、REMOTE_USER 変数の値が使用されます。
- HTTP_INTERNETUSER および REMOTE_USER が設定されていない場合、ユーザーは匿名です。

出力

ユーザー名を文字列で返します。

例

次はページまたはログに出力される情報の例です。

```
HTTP_INTERNETUSER=admin
```

次は Idoc スクリプト文または評価の一部としての使用例です。

```
<$if HTTP_INTERNETUSER$>
<!--statement-->
```

HTTP_REFERER

タイプおよび使用方法

- Web サーバー変数（4-36 ページ）
- Web サーバー（5-42 ページ）

説明

ローカル・サーバー上の参照先ディレクトリの完全な URL を取得します。

出力

完全な URL を文字列で返します。

例

次はページまたはログに出力される情報の例です。

```
HTTP_REFERER=http://centralserver/stellent/
```

次は Idoc スクリプト文または評価の一部としての使用例です。

```
<$if HTTP_REFERER$>
<!--statement-->
```

HTTP_USER_AGENT

タイプおよび使用方法

- Web サーバー変数（4-36 ページ）
- Web サーバー（5-42 ページ）

説明

クライアント・ブラウザのタイプ、バージョン番号、ライブラリおよびブラウザが構成されているプラットフォームを取得します。

出力

`software/version (library) (platform)` の形式で文字列を返します。

例

次はページまたはログに出力される情報の例です。

```
HTTP_USER_AGENT=Mozilla/4.7 [en] (WinNT; U)
```

次は Idoc スクリプト文または評価の一部としての使用例です。

```
<$if HTTP_USER_AGENT$>
<!--statement-->
```

HTTPABSOLUTECGIPATH

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・サーバー（5-8 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

コンテンツ・サーバーの CGI パスを完全な URL として取得します。

- この変数は、直接変更できません。パスを変更するには、HttpAbsoluteCgiRoot（6-308 ページ）構成設定を使用します。

出力

コンテンツ・サーバーの CGI パスを文字列で返します。

例

マスター・コンテンツ・サーバー（stellent）の例を次に示します。

```
HttpAbsoluteCgiPath=http://localhost/stellent/idcplg/
```

プロキシ・コンテンツ・サーバー（stellent_2）の例を次に示します。

```
HttpAbsoluteCgiPath=http://localhost/stellent/idcplg/stellent_2/pxs
```

関連項目

- HttpCgiPath（6-312 ページ）
- HttpAbsoluteCgiRoot（6-308 ページ）
- HttpBrowserFullCgiPath（6-311 ページ）

HTTPABSOLUTECGIROOT

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・サーバー（5-8 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

コンテンツ・サーバーの CGI パスを完全な URL として定義します。

場所

<*install_dir*>/config/config.cfg

例

マスター・コンテンツ・サーバーの例を次に示します。

```
HttpAbsoluteCgiRoot=http://my_computer/idcplg/idc_cgi_isapi-<instance>.dll
```

プロキシ・コンテンツ・サーバー（stellent_2）の例を次に示します。

```
HttpAbsoluteCgiRoot=http://my_computer/idcplg/
idc_cgi_isapi-<instance>.dll/stellent_2/pxs
```

関連項目

– HttpAbsoluteCgiPath（6-307 ページ）

HTTPABSOLUTEWEBROOT

タイプおよび使用方法

- 値変数（4-34 ページ）
- ディレクトリおよびパス（5-22 ページ）
- Web サーバー（5-42 ページ）

説明

Web サーバーのルート・ディレクトリを完全な URL として定義します。

場所

<*install_dir*>/config/config.cfg

例

マスター・コンテンツ・サーバーの例を次に示します。

```
HttpAbsoluteWebRoot=http://my_computer/stellent/
```

プロキシ・コンテンツ・サーバー（stellent_2）の例を次に示します。

```
HttpAbsoluteWebRoot=http://my_computer/stellent_2/
```

関連項目

– HttpRelativeWebRoot（6-321 ページ）

HTTPADMINCGIPATH

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・サーバー（5-8 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

管理サーバーの CGI パスを相対 URL として取得します。

出力

管理サーバーの CGI パスを文字列で返します。

例

マスターまたはプロキシ・コンテンツ・サーバーの例を次に示します。

```
HttpAdminCgiPath=/idcplg/idc_cgi_isapi-<instance>.dll/cs-admin/pxs
```

HTTPBROWSERFULLCGIPATH

タイプおよび使用方法

- 値変数（4-34 ページ）
- ディレクトリおよびパス（5-22 ページ）
- クライアント（5-4 ページ）

説明

この変数を使用して、アプレットおよび管理サーバー用に作成された CGI パス構成に対する明示的な制御を設定します。

- Content Server は、ブラウザのアドレス・バーで HTTP アドレスを評価します。`HttpIgnoreWebServerInternalPortNumber` 構成エントリでは評価を十分に支援できない場合は、明示的な回答を持つ `HttpBrowserFullCgiPath` を設定できます。
- コンテンツ・サーバーの相対パスを指定します（後述の例の `idc1`）。

例

```
HttpBrowserFullCgiPath=http://localhost/idc1/idcplg
```

関連項目

– HttpIgnoreWebServerInternalPortNumber（6-316 ページ）

HTTPCGIPATH

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・サーバー（5-8 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

コンテンツ・サーバーの CGI パスを取得します。

- 次の構成設定を使用して、この変数の値を決定します。

変数	説明
UseSSL（6-881 ページ）	TRUE に設定されている場合、Secure Sockets Layer（SSL）が使用されます（*http* ではなく、*https* を使用）。
isAbsoluteCgi	相対パスではなく、完全な URL を使用するかどうかを定義します。これはコンテンツ・サーバーによって設定される内部フラグであり、ユーザーが構成するものではありません。

出力

コンテンツ・サーバーの CGI パスを文字列で返します。

例

マスター・コンテンツ・サーバーの例を次に示します。

```
HttpCgiPath=/stellent/idcplg
```

プロキシ・コンテンツ・サーバー（stellent_2）の例を次に示します。

```
HttpCgiPath=/stellent/idcplg/stellent_2/pxs
```

関連項目

- HttpAbsoluteCgiPath（6-307 ページ）
- HttpEnterpriseCgiPath（6-314 ページ）
- UseSSL（6-881 ページ）

HTTPCOMMONROOT

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・サーバー（5-8 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

common ディレクトリの URL を取得します。

- 1 つのコンテンツ・サーバーのインストールからのリソースを複数のコンテンツ・サーバーが共有できます。この変数は、リソースが共有されるコンテンツ・サーバーの <*install_dir*>/common/ ディレクトリの URL パスを定義します。たとえば、*HttpCommonRoot* は、Web アプレットがある共有の共通ディレクトリへのアクセスに使用する接頭辞を定義します。

出力

相対 URL を文字列で返します。外部の URL である場合は、相対パスのかわりに完全な URL を返します。

例

マスターまたはプロキシ・コンテンツ・サーバーの例を次に示します。

```
HttpCommonRoot=/stellent/common/
```

関連項目

- HttpHelpRoot（6-315 ページ）
- HttpImagesRoot（6-317 ページ）
- HttpSharedRoot（6-323 ページ）
- HttpWebRoot（6-325 ページ）

HTTPENTERPRISECGIPATH

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・サーバー（5-8 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

マスター・コンテンツ・サーバーの CGI パスを相対 URL として取得します。

- 複数のコンテンツ・サーバーが同一の Web ログインを共有する場合は、いずれかのコンテンツ・サーバーがマスターまたはエンタープライズ・サーバーとして指定されます。

出力

マスター・コンテンツ・サーバーの CGI パスを文字列で返します。

例

マスターまたはプロキシ・コンテンツ・サーバーの例を次に示します。

```
HttpEnterpriseCgiPath=/idcplg/idc_cgi_isapi-<instance>.dll
```

関連項目

– HttpCgiPath（6-312 ページ）

HTTPHELPROOT

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・サーバー（5-8 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

help ディレクトリの URL を取得します。

- 1 つのコンテンツ・サーバーのインストールからのオンライン・ヘルプ・ファイルを複数のコンテンツ・サーバーが共有できます。この変数は、ヘルプ・ファイルが共有されるコンテンツ・サーバーの <*install_dir*>/weblayout/help ディレクトリの URL パスを定義します。

出力

相対 URL を文字列で返します。外部の URL である場合は、相対パスのかわりに完全な URL を返します。

例

マスターまたはプロキシ・コンテンツ・サーバーの例を次に示します。

```
HttpHelpRoot=/stellent/help/
```

関連項目

- HttpCommonRoot（6-313 ページ）
- HttpImagesRoot（6-317 ページ）
- HttpSharedRoot（6-323 ページ）
- HttpWebRoot（6-325 ページ）

HTTPIGNOREWEBSERVERINTERNALPORTNUMBER

タイプおよび使用方法

- 動的変数（4-20 ページ）
- Web サーバー（5-42 ページ）

説明

TRUE に設定すると、管理サーバーおよびアプレット用に計算された HTTP アドレスは、Web サーバーが提供するサーバー・ポート値を使用しません。

- この変数を設定すると、ロード・バランシング構成で、ロード・バランシング・サーバーが使用するポート番号が、リクエストへのレスポンスを実際に配信する内部 Web サーバーのポート番号と異なるという問題を回避できます。

場所

<*install_dir*>/config/config.cfg

例

```
HttpIgnoreWebServerInternalPortNumber=true
```

関連項目

– HttpBrowserFullCgiPath（6-311 ページ）

HTTPIMAGESROOT

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・サーバー（5-8 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

images ディレクトリの URL を取得します。

- 1 つのコンテンツ・サーバーのインストールからユーザー・インタフェース・イメージを複数のコンテンツ・サーバーが共有できます。この変数は、イメージ・ファイルが共有されるコンテンツ・サーバーの <*install_dir*>/weblayout/images ディレクトリの URL パスを定義します。

出力

相対 URL を文字列で返します。外部の URL である場合は、相対パスのかわりに完全な URL を返します。

例

マスターまたはプロキシ・コンテンツ・サーバーの例を次に示します。

```
HttpImagesRoot=/stellent/images/
```

関連項目

- HttpCommonRoot（6-313 ページ）
- HttpHelpRoot（6-315 ページ）
- HttpSharedRoot（6-323 ページ）
- HttpWebRoot（6-325 ページ）

HTTPLAYOUTROOT

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・サーバー（5-8 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

現在のレイアウト・ファイルが含まれているフォルダの URL を取得します。現在のレイアウトに固有の追加の JavaScript リソースがある場合に役立ちます。

HTTPRELATIVEADMINROOT

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・サーバー（5-8 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

管理サーバーの相対 URL を取得します。

- インストールの際のデフォルトは、`/cs-admin/` です。

出力

相対 URL を文字列で返します。

例

マスターまたはプロキシ・コンテンツ・サーバーの例を次に示します。

```
HttpRelativeAdminRoot=/cs-admin/
```

HTTPRELATIVECGIROOT

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・サーバー（5-8 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

コンテンツ・サーバーの CGI パスを相対 URL として定義します。

場所

<*install_dir*>/config/config.cfg

例

マスター・コンテンツ・サーバーの例を次に示します。

```
HttpRelativeCgiRoot=/idcplg/idc_cgi_isapi-<instance>.dll
```

プロキシ・コンテンツ・サーバー（stellent_2）の例を次に示します。

```
HttpRelativeCgiRoot=/idcplg/idc_cgi_isapi-<instance>.dll/stellent_2/pxs
```

関連項目

– HttpAbsoluteCgiRoot（6-308 ページ）

HTTPRELATIVEWEBROOT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ディレクトリおよびパス（5-22 ページ）
- Web サーバー（5-42 ページ）

説明

Web サーバーのルート・ディレクトリを相対 URL として定義します。

- *http://www.mycomputer.com/stellent/* などの完全なルートではなく、*/stellent/* などの相対ルートを使用します。
- 相対 Web ルート・ディレクトリを文字列で返します。
- デフォルト値はありません。

場所

- 「System Properties」→「Internet」タブ→「Http Relative Web Root」
- コンテンツ・サーバー : <*install_dir*>/config/config.cfg
- 管理サーバー : <*install_dir*>/admin/config/config.cfg

例

構成設定として、相対 Web ルートを定義します。

```
HttpRelativeWebRoot=/stellent/
```

スクリプトとして、相対 Web ルートを文字列で返します。

```
<$HttpRelativeWebRoot$>
```

関連項目

– HttpAbsoluteWebRoot（6-309 ページ）

HTTPSERVERADDRESS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ディレクトリおよびパス（5-22 ページ）
- Web サーバー（5-42 ページ）

説明

Web サーバー・アドレスを部分 URL として定義します。

- *http://www.mycomputer.com/* などの完全なアドレスではなく、*mycomputer* などの部分 URL を使用します。
- Web サーバー・アドレスを文字列で返します。
- デフォルト値はありません。

場所

- 「System Properties」→「Internet」タブ→「HTTP Server Address」
- コンテンツ・サーバー : <*install_dir*>/config/config.cfg
- 管理サーバー : <*install_dir*>/admin/config/config.cfg

例

構成設定として、Web サーバー・アドレスを定義します。

```
HttpServerAddress=mycomputer
```

スクリプトとして、Web サーバー・アドレスを文字列で返します。

```
<$HttpServerAddress$>
```

関連項目

- IdcAdminServerHostName（6-327 ページ）
- IdcAdminServerHostName（6-328 ページ）
- IdcCommandServerHost（6-334 ページ）
- IntradocServerHostName（6-370 ページ）

HTTPSHAREDROOT

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・サーバー（5-8 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

shared ディレクトリの URL パスを取得します。

- 1 つのコンテンツ・サーバーのインストールからのリソースを複数のコンテンツ・サーバーが共有できます。この変数は、リソース・ファイルが共有されるコンテンツ・サーバーの <*install_dir*>/shared/ ディレクトリの URL パスを定義します。

出力

相対 URL を文字列で返します。外部の URL である場合は、相対パスのかわりに完全な URL を返します。

例

マスターまたはプロキシ・コンテンツ・サーバーの例を次に示します。

```
HttpSharedRoot=/stellent/
```

関連項目

- HttpCommonRoot（6-313 ページ）
- HttpHelpRoot（6-315 ページ）
- HttpImagesRoot（6-317 ページ）
- HttpWebRoot（6-325 ページ）
- SharedDir（6-739 ページ）

HTTPSYSTEMHELPROOT

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・サーバー（5-8 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

この変数は、デフォルトのシステム言語のヘルプ・ファイルへのパスを返すことを除き、HttpHelpRoot 変数と同じです。

出力

ヘルプ・ファイルのパスを返します。

例

```
HttpSystemHelpRoot=/stellent/help/
```

関連項目

- HttpCommonRoot（6-313 ページ）
- HttpHelpRoot（6-315 ページ）
- HttpImagesRoot（6-317 ページ）
- HttpWebRoot（6-325 ページ）

HTTPWEBROOT

タイプおよび使用方法

- 値変数（4-34 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

weblayout ディレクトリの URL パスを取得します。

- 1 つのコンテンツ・サーバーのインストールからのリソースを複数のコンテンツ・サーバーが共有できます。この変数は、リソース・ファイルが共有されるコンテンツ・サーバーの <*install_dir*>/weblayout/ ディレクトリの URL パスを定義します。
- 次の構成設定により、この変数の値が決まります。

変数	説明
UseSSL（6-881 ページ）	TRUE に設定されている場合、Secure Sockets Layer（SSL）が使用されます（*http* ではなく、*https* を使用）。
isAbsoluteWeb	相対パスではなく、完全な URL を使用するかどうかを定義します。これはコンテンツ・サーバーによって設定される内部フラグであり、ユーザーが構成するものではありません。

出力

相対 URL を文字列で返します。外部の URL である場合は、相対パスのかわりに完全な URL を返します。

例

マスターまたはプロキシ・コンテンツ・サーバーの例を次に示します。

```
HttpWebRoot=/stellent/
```

関連項目

- UseSSL（6-881 ページ）
- HttpCommonRoot（6-313 ページ）
- HttpHelpRoot（6-315 ページ）
- HttpImagesRoot（6-317 ページ）
- HttpSharedRoot（6-323 ページ）

IDC_NAME

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

コンテンツ・サーバー・インスタンスの一意名を定義します。

警告 : 重複した *IDC_Name* を使用すると、データの破損の原因になります。アーカイバでは、同じ *IDC_Name* を共有する 2 つのインスタンス間でのデータの移動またはコピーはできません。このような操作を行うと、ターゲット・システムのデータが破損します。各コンテンツ・サーバー・インスタンスに一意の *IDC_Name* を付けます。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IDC_Name=StellentMSP
```

関連項目

- InstanceDescription（6-366 ページ）
- InstanceMenuLabel（6-367 ページ）

IDCADMINSERVERHOSTNAME

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

管理サーバーが使用するホスト名を定義します。通常は、小規模のエンタープライズ・デプロイでの HttpServerAddress（6-322 ページ）に類似しています。

場所

<*install_dir*>/admin/config/config.cfg

例

次は構成エントリの使用例です。

```
IdcAdminServerHostName=stellent.company.com
```

関連項目

– HttpServerAddress（6-322 ページ）

IDCADMINSERVERHOSTNAME

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

完全修飾サーバー名を定義します。

- `HttpServerAddress` が外部名に設定されている場合、ベスト・プラクティスは、`IntradocServerHostName` および `IdcAdminServerHostName` をサーバー名に定義することです。

場所

<*install_dir*>/config/config.cfg

例

SampleCompany での *Main* サーバーの構成エントリとしての使用例を次に示します。

```
IdcAdminServerHostName=Main.SampleCompany.de
```

関連項目

- HttpServerAddress（6-322 ページ）
- IntradocServerHostName（6-370 ページ）

IDCADMINSERVERPORT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

管理サーバーが使用するポートを定義します。

- デフォルトは、4440 です。

場所

<*install_dir*>/admin/config/config.cfg

例

次は構成エントリの使用例です。

```
IdcAdminServerPort=5550
```

IDCANALYZELOGDIR

タイプおよび使用方法

- 構成変数（4-3 ページ）
- デバッグ（5-20 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

Content Server Analyzer ユーティリティの出力ファイルの書込み先、およびファイル・システムの分析処理中に検出された追加ファイルの転送先を指定します。

- 部分ディレクトリ・パスまたは完全なディレクトリ・パスを指定できます。
  - 部分ディレクトリ・パス（*analyzer_logs/* など）を指定した場合は、そのサブディレクトリが <*install_dir*>/bin/ ディレクトリに作成されます。
  - 完全なディレクトリ・パス（*c:/analyzer_logs* など）を指定した場合は、それがログ・ディレクトリになります。
- ログ・ファイル名は、LogName（6-509 ページ）の設定で指定します。
- デフォルトは、<*install_dir*>/bin/*logs/* です。

場所

<*install_dir*>/bin/intradoc.cfg

例

次は構成設定の例です。

```
IdcAnalyzeLogDir=f:/stellent/analyzer_logs/
```

関連項目

- LogName（6-509 ページ）
- SafeDir（6-695 ページ）
- TempDir（6-795 ページ）

IDCAUTHEXTRAREQUESTPARAMS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Web サーバー（5-42 ページ）

説明

Web サーバー・プラグインがローカル・データから取得し、リクエストのヘッダーで送信する値を指定します。

- たとえば、Web サーバー・フィルタのプラグインで、ユーザーの属性を取得してそれをローカルな値に設定する場合は、*IdcAuthExtraRequestParams* に属性名を指定して、コンテンツ・サーバーにデータを渡すことができます。
- また、この設定により、Idoc スクリプトを使用して追加の HTTP ヘッダー値にアクセスすることもできます。これらの値は、データ・バインダの環境変数設定の一部となります。
  - *IdcAuthExtraRequestParams* リストの名前は、大 / 小文字には関係なくヘッダーの名前と一致する必要があります（大 / 小文字の区別なしの一致）。
  - Idoc スクリプト変数として使用する場合、*IdcAuthExtraRequestParams* の名前は、正確に一致する必要があります（大 / 小文字を区別する一致）。
- デフォルト値はありません。

場所

<*install_dir*>/config/config.cfg

例

次の設定は、IIS Web サーバー・プラグインが、ヘッダーで HTTP_REFERER のローカル値を送信するように指定します（IIS では *HTTP_* 接頭辞が自動的に追加されます）。

```
IdcAuthExtraRequestParams=referer
```

IDCCOMMANDFILE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

IdcCommand を使用して実行されるコマンドが含まれるファイルを指定します。

- 詳細は、『IdcCommand—Java Command Utility Reference Guide』を参照してください。
- デフォルトの設定はありません。

場所

<*install_dir*>/bin/intradoc.cfg

例

次は構成エントリの使用例です。

```
IdcCommandFile=c:/idc/add_user.txt
```

IDCCOMMANDLOG

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

IdcCommand 実行のログ・ファイルのパスを指定します。

- ログ・ファイルには、コマンドの実行時間およびコマンドの成功または失敗のステータスが含まれます。
- 詳細は、『IdcCommand—Java Command Utility Reference Guide』を参照してください。
- デフォルトの設定はありません。

場所

<*install_dir*>/bin/intradoc.cfg

例

次は構成エントリの使用例です。

```
IdcCommandLog=c:/idc/idccommand.log
```

IDCCOMMANDSERVERHOST

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

IdcCommand の接続先のコンテンツ・サーバーの Web サーバー・アドレスを指定します。

- デフォルトは、IdcCommand の接続先のコンテンツ・サーバーの HttpServerAddress（6-322 ページ）です。
- 詳細は、『IdcCommand—Java Command Utility Reference Guide』を参照してください。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IdcCommandServerHost=instance_name.company.com
```

関連項目

– HttpServerAddress（6-322 ページ）

IDCCOMMANDUSER

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

IdcCommand を使用したコマンドの実行を許可されるユーザーを指定します。

- 詳細は、『IdcCommand—Java Command Utility Reference Guide』を参照してください。
- デフォルトの設定はありません。

場所

<*install_dir*>/bin/intradoc.cfg

例

次は構成エントリの使用例です。

```
IdcCommandUser=sysadmin
```

IDCHTTPHEADERVARIABLES

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Web サーバー（5-42 ページ）

説明

HTTP ヘッダー・レスポンスに含めるデータ・パラメータを定義します。

- これはカンマ区切りのリストです。
- 名前と値のペアのエンコーディング形式は、コンテンツ・サーバーの HTTP ヘッダー・エンコーディングの適用後、一般的な hda 形式です。
- 値は UTF-8 でエンコードされ、特殊文字は %*xx* でエンコードされます。
- この設定は、特に、クライアントによるリクエストを監査する必要がある Web サーバー・プラグイン向けに設計されています。プラグインでは、コンテンツ・サーバー・レスポンスの HTTP ヘッダーは確認できますが、コンテンツの本文を確認できません。リクエストの一部のパラメータを HTTP ヘッダー・レスポンス内にプッシュすることで、プラグインは、アクセスされたドキュメントおよび実行されたアクションを監査できます。
- HTTP ベースのクライアントにレスポンスを送信するときに、コンテンツ・サーバーによって生成される HTTP ヘッダーの名前は、*IdcVariables* です。

場所

<*install_dir*>/config/config.cfg

例

IdcHttpHeaderVariables=IdcService,dDocName の場合、UTF-8 エンコーディングを適用する前のヘッダー値は次のようになります。

```
@Properties LocalData
IdcService=GET_DOC_PAGE
dDocName=TestDoc
blDateFormat=M/d{/yy}{ h:mm[:ss]{ a}}!mAM,PM!tAmerica/New_York
blFieldTypes=
@end
```

UTF-8 エンコーディングでは、改行文字が *%0A* 文字に変更される以外は、すべての文字がそのままになります。blDateFormat は、コンテンツ・サーバーのシステム日付の書式です。

IDCLOCALE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ローカライゼーション（5-29 ページ）

説明

現在のユーザーのロケールを追跡するためにクライアントの HTTP Cookie の値を設定します。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IdcLocale=Deutsch
```

IDCREGISTRATIONPATH

タイプおよび使用方法

- 値変数（4-34 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

Oracle ライセンス・サーバーのパスを指定します。「Registration」ページからライセンス・キーを取得し検証するために使用されます。

- デフォルトは、*http://license.stellent.com/idcplg/idc_cgi_isapi-idcm1.dll* です。

出力

文字列としてライセンス・サーバー・パスを返します。

例

ライセンス・サーバー・パスを返します。

```
<$IdcRegistrationPath$>
```

IDCSERVERBINDADDRESS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- クラスタリング（5-5 ページ）

説明

Content Server を特定の IP アドレスにバインドするためにクラスタ環境で使用されます。

場所

リクエストをリスニングするコンテンツ・サーバーの config.cfg ファイルに設定します。

- <*install_dir*>/config/config.cfg

例

次は構成変数としての使用例です。

```
IdcServerBindAddress=IPaddress
```

IDCSERVERSOCKETQUEUEDEPTH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）
- クラスタリング（5-5 ページ）

説明

TCP/IP ソケット・キューの深さを指定します。

- この設定を使用して、特にクラスタ環境で、コンテンツ・サーバーのパフォーマンスをチューニングできます。この値を減らすと、コンテンツ・サーバーのスループットが制限される場合があります。
- デフォルトは、200 です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IdcServerSocketQueueDepth=256
```

IMAGEALCHEMYEXEPATH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

PDF ファイルのサムネイルを作成するために Thumbnails プログラムで使用される Image Alchemy 実行可能ファイルのパスを設定します。

- 文字列としてパスを返します。
- デフォルトは、*<SharedDir>*/alchemy/ALCHLONG.EXE です。

場所

<*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
ImageAlchemyExePath=c:/Program Files/Alchemy/alchlong.exe
```

関連項目

- CreatePDFThumbnails（6-111 ページ）
- IsThumbnailPresent（6-443 ページ）

INC

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- Idoc スクリプト（5-27 ページ）

説明

インクルード・リソースからコードをページに追加します。

- この関数は、Idoc スクリプト変数をパラメータとして取ることができる以外は、*include* キーワードと同じ操作を実行します。詳細は、3-19 ページの「キーワードと関数の対比」を参照してください。

パラメータ

1 つのパラメータを取ります。

- パラメータは、インクルードの名前を定義する変数のみです。

出力

指定されたインクルード・リソースに定義されたコードを表示します。

例

たとえば、すべてのカスタム・メタデータ・フィールドではなく、一部のフィールドのみで Idoc スクリプトを実行する場合があります。その場合は、次の Idoc スクリプトを実行して、フィールド名に基づいてインクルードを動的に作成できます（*specific_include_xComments* など）。

```
<$loop DocMetaDefinition$>
    <$myInclude = "specific_include_" & dName$>
    <$exec inc(myInclude)$>
<$endloop$>
```

exec キーワードの使用により、*inc* 関数によって指定されたインクルードの出力が抑制される点に注意してください。*inc* の前に *exec* を使用しないと、指定されたインクルード内の HTML がページに表示されます。

specific_include_xComments が存在しない場合でも、出力が表示されないため、このコードはエラーをスローしません。

関連項目

- キーワードと関数の対比（3-19 ページ）
- include（6-347 ページ）
- setResourceInclude（6-737 ページ）

INCDYNAMICCONVERSIONBYRULE

タイプおよび使用方法

❖ グローバル関数（4-21 ページ）

❖ Dynamic Converter（5-17 ページ）

説明

特定の Dynamic Converter 変換ルールに関連付けられたテンプレートおよびレイアウトを使用して、ドキュメントの LatestReleased リビジョンの変換結果を返します。

❖ 強制変換によって作成されたグラグメントは、この関数を使用して直接参照できます。

パラメータ

パラメータを 2 つ取ります。

❖ 最初のパラメータは、ドキュメントのコンテンツ ID です。

❖ 2 番目のパラメータは、変換ルールの名前です。

出力

指定された変換ルールに定義されたテンプレートおよびレイアウトに従って変換された指定のドキュメントを取得します。

例

変換ルール *RuleA* に従って変換された、コンテンツ ID が *PhoneList* であるコンテンツ・アイテムの変換後のレンディションを取得します。

```
<$incDynamicConversionByRule("PhoneList","RuleA")$>
```

関連項目

– ForcedConversionRules（6-236 ページ）

INCGLOBAL

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- Idoc スクリプト（5-27 ページ）

説明

テキスト・ファイルの全内容を現在のページの表示に含めます。

- この関数は、デフォルトのポータル・ページの生成に使用されます。
- グローバル・インクルード・ファイルは、HTML または Idoc スクリプト（あるいはその両方）を含むテキスト・ファイルです。ファイルはサーバーの起動時にロードされます。
- グローバル・インクルード・ファイルは、*.inc* 拡張子が付けられ、<install_dir>/data/pages/ ディレクトリに保存される必要があります。

パラメータ

1 つのパラメータを取ります。

- パラメータは、*.inc* ファイル拡張子なしのグローバル・インクルード・ファイル名のみです。

出力

指定されたグローバル・インクルード・ファイルに定義されたコードを表示します。

例

Web レイアウト・エディタのポータル更新機能を使用してポータル・ページを変更すると、変更されたテキストを含む *portal_message.inc* という名前のグローバル・インクルード・ファイルが作成されます。このファイルは、次のように、*pne_home_page.htm* テンプレート・ファイルで参照されます。

```
<$incGlobal("portal_message")$>
```

INCLUDE

タイプおよび使用方法

- 特殊キーワード（3-18 ページ）
- Idoc スクリプト（5-27 ページ）

説明

インクルード・リソース（<@dynamichtml include_name@> で定義）からコードをページに追加します。

- このキーワードにより、複数のテンプレート・ページで小さいコード・チャンクの再利用が可能となるため、これは Idoc スクリプトで最もよく使用されるコマンドです。
- 詳細は、3-4 ページの「インクルード」を参照してください。

パラメータ

1 つのパラメータを取ります。

- パラメータは、インクルードの名前のみです。

出力

指定されたインクルード・リソースに定義されたコードを表示します。

例

次のインクルードは、ほとんどのコンテンツ・サーバー Web ページの最初と最後を作成するために使用されます。これらのインクルードは、<*install_dir*>/shared/config/resources/standard_page.htm ファイルに定義されます。

```
<$include std_html_head_declarations$>
<$include body_def$>
<$include std_page_begin$>
  Hello World!
<$include std_page_end$>
```

関連項目

- キーワードと関数の対比（3-19 ページ）
- インクルード（3-4 ページ）
- setResourceInclude（6-737 ページ）
- inc（6-343 ページ）

INCLUDEXMLFLEXIONDOCTYPE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- XML Converter（5-16 ページ）

説明

FlexionDoc スキーマを使用する場合、Document Type Definition（DTD）を参照する DOCTYPE タグが XML 出力に作成されます。

- この設定は、UseXmlSchema（6-889 ページ）を TRUE に設定した場合にのみ有効です。
- TRUE に設定すると、*flexiondoc.dtd* を参照する DOCTYPE タグが XML 出力に含まれます。
- FALSE に設定すると、XML 出力に DOCTYPE タグは含まれません。
- デフォルトは、TRUE です。

場所

- 「Local Inbound Refinery Configuration」画面→「XML Options」タブ→「Include a DOCTYPE tag in FlexionDoc xml output」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
IncludeXmlFlexionDocType=true
```

関連項目

– UseXmlSchema（6-889 ページ）

INCTEMPLATE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- Idoc スクリプト（5-27 ページ）

説明

Idoc スクリプトを評価した後、コンテンツ・サーバー・テンプレートの内容をページに追加します。

- この関数を使用して、テンプレート全体の内容を含めることができます。ただし、通常では、リソース・インクルードに十分な柔軟性があり、複数ページでの Idoc スクリプトの共有に必要なすべてがサポートされるため、この使用方法は推奨されません。

パラメータ

1 つのパラメータを取ります。

- パラメータは、*.htm* ファイル拡張子のないコンテンツ・サーバー・テンプレートのファイル名のみです。

出力

指定されたテンプレートに定義されたコードを表示します。

例

new_look テンプレート・ファイルを取得します。

```
<$incTemplate("new_look")$>
```

INDEXABLERENDITIONS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）

説明

プライマリ、代替または Web で表示可能なレンディションを除く、索引付けされるレンディションの単一文字コードのカンマ区切りのリストです。索引付けの可能なレンディションは 1 つのみとなるため（XML レンディションなど）、通常は 1 つの文字のみが使用されます。

これは `IsAllowIndexRenditions` とともに使用されます。

場所

<*install_dir*>/config/config.cfg

関連項目

– IsAllowIndexRenditions（6-373 ページ）

INDEXERAUTOWORKINTERVAL

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）

説明

インデクサの自動更新サイクルが変更をチェックする間隔を秒単位で定義します。

- デフォルトは、300（5 分）です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IndexerAutoWorkInterval=600
```

INDEXERLARGEFILESIZE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）

説明

インデクサのバルクロード（バッチ）ファイルに入れるファイルのサイズ（MB）を決定します。

- この設定を超えるファイルは、別のバルクロードで索引付けされます。この設定より小さいファイルは、他のファイルとともにバッチで索引付けされます。
- デフォルトは、1MB です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IndexerLargeFileSize=2
```

IndexerPath

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）

説明

インデクサ・プログラムのパスを定義します。

場所

- バージョン 10.0 より前の Content Server: <*install_dir*>/config/config.cfg
- バージョン 10.0 以上 : <*install_dir*>/bin/intradoc.cfg

例

次は構成エントリの使用例です。

```
IndexerPath=c:/stellent/shared/search/vdk/_nti40/bin/mkvdk.exe
```

関連項目

– IndexerStyleDir（6-358 ページ）

INDEXERREBUILDSTAGINGMANUALCOPY

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）

説明

別の場所にコレクションを作成するために索引の再構築を有効にします。

- 通常、この変数は、非常に大きいコレクションを再構築する場合に使用します。再構築が終了したら、結果を手動で元のコンテンツ・サーバー・コレクションにマージできます。
- TRUE に設定すると、再構築された索引は、IndexerRebuildStagingPath（6-356 ページ）値によって設定された場所に作成されます。
- FALSE に設定すると、再構築された索引は、次にアクティブな索引ディレクトリに作成されます。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IndexerRebuildStagingManualCopy=true
```

関連項目

– IndexerRebuildStagingPath（6-356 ページ）

INDEXERREBUILDSTAGINGPATH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）

説明

インデクサの再構築が作成されるパスを定義します。

- 通常、この変数は、非常に大きいコレクションを再構築する場合に使用します。再構築が終了したら、結果を手動で元のコンテンツ・サーバー・コレクションにマージできます。
- デフォルト値はありません。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IndexerRebuildStagingPath=c:/buildX/
```

関連項目

– IndexerRebuildStagingManualCopy（6-355 ページ）

INDEXERSETCOLLECTIONVALUE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 検索（5-34 ページ）

説明

Verity Integration で使用されます。Verity Integration は、Content Server とともに使用できるアドオン製品であり、購入することができます。

この関数は、Verity 検索インデクサのロジックを変更します。この関数は、`verity_indexer_params_override` インクルード・ファイルでのみ使用されます。索引の収集に関する累積統計を計算できるように設計されています。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、使用するフラグの名前です。
- 2 番目のパラメータは、フラグに起因する値です。

任意の名前および値を使用できます。これらの値は、索引の収集のための次回のリソース・インクルードの実行で使用可能になります。

IndexerStyleDir

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）

説明

Verity Integration で使用されます。Verity Integration は、Content Server とともに使用できるアドオン製品であり、購入することができます。

Content Server のプロキシ・インスタンス上の Verity インデクサのスタイル・ファイルのパスを定義します。

場所

- バージョン 8.0 より前の Content Server: <*install_dir*>/config/config.cfg
- バージョン 8.0 以上 : <*install_dir*>/bin/intradoc.cfg

例

次は構成エントリの使用例です。

```
IndexerStyleDir=c:/newlocation/search/style/basic
```

関連項目

– IndexerPath（6-354 ページ）

IndexerTouchMonitorInterval

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）

説明

コンテンツ・サーバーが、タイムスタンプの変更されていないインデクサ・ロック・ファイルを失効として処理するまでの待機時間を定義します。また、この期間中にコンテンツ・サーバーは、維持しているロックに対する touch を 2 回実行します。

- インデクサ・ロックは、適切なディレクトリにロック・ファイルを配置することで作成されます。ロックを持続するには、ファイルのタイムスタンプを定期的に変更する必要があります。そうでない場合、コンテンツ・サーバーは、ロックが期限切れになったとみなします。
- この設定値が小さいほど、索引付けサイクル間でのロックの交代が速くなります。値が大きいほど、負荷の高いアクティビティによって touch リクエストが何秒間も遅延し、コンテンツ・サーバーが偽ってロックの失効を報告する可能性が高くなります。虚偽のロックの失効は、処理の重複を生じる可能性があります（コンテンツ・サーバーの状態への長期的な影響はありません）。
- デフォルトは、12000 ミリ秒（12 秒）です。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
IndexerTouchMonitorInterval=15000
```

関連項目

- IndexerTraceCycleListChanges（6-360 ページ）
- IndexerTraceLockFileChanges（6-361 ページ）

INDEXERTRACECYCLELISTCHANGES

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）
- デバッグ（5-20 ページ）

説明

コンテンツ・サーバーによって維持されるアクティブな索引付けサイクルのリストへの変更について、コンテンツ・サーバーのコンソールにトレース情報をダンプします。

- 索引付けサイクルの特定の時点で排他ロックが必要となるため、索引付けサイクルは、他のサイクルが非アクティブになるのを待機する場合があります。この設定を有効にすると、インデクサのパフォーマンスの監査に役立ちます。
- TRUE に設定すると、索引付けサイクル・リストへの変更がコンテンツ・サーバー・コンソールに出力されます。
- FALSE に設定すると、索引付けサイクル・リストへの変更は出力されません。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IndexerTraceCycleListChanges=true
```

関連項目

– IndexerTouchMonitorInterval（6-359 ページ）
– IndexerTraceLockFileChanges（6-361 ページ）

注意：この変数は、Content Server 7.0 リリース以降では不要になりました。この機能は、「Admin Applets」の「System Audit Information」ページで利用できるようになりました。

INDEXERTRACELOCKFILECHANGES

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）
- デバッグ（5-20 ページ）

説明

コンテンツ・サーバーがインデクサのロック・ファイルおよび一時停止ファイルに対して行った変更について、コンテンツ・サーバーのコンソールにトレース情報をダンプします。

- TRUE に設定すると、ファイルへの変更がコンテンツ・サーバー・コンソールに出力されます。
- FALSE に設定すると、ファイルへの変更は出力されません。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IndexerTraceLockFileChanges=true
```

関連項目

– IndexerTouchMonitorInterval（6-359 ページ）
– IndexerTraceCycleListChanges（6-360 ページ）

注意：この変数は、Content Server 7.0 リリース以降では不要になりました。この機能は、「Admin Applets」の「System Audit Information」ページで利用できるようになりました。

INDEXERTRACESTARTSTOP

タイプおよび使用方法

- 構成変数（4-3 ページ）
- デバッグ（5-20 ページ）
- 索引付け（5-28 ページ）

説明

- TRUE に設定すると、索引付けサイクルの開始と停止を追跡します。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IndexerTraceStartStop=true
```

関連項目

- IndexerTouchMonitorInterval（6-359 ページ）
- IndexerTraceCycleListChanges（6-360 ページ）

注意：この変数は、Content Server 7.0 リリース以降では不要になりました。この機能は、「Admin Applets」の「System Audit Information」ページで利用できるようになりました。

INDEXVAULTFILE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）

説明

Inbound Refinery によって変換されない大きいファイルが存在する場合、リポジトリ内に必要なファイルは 1 つのみです。この変数は、ボールト内のファイルを指す weblayout 領域にファイルを作成するために使用されます。

- PASSTHRU として指定されたファイル（Inbound Refinery で処理されないファイル）とともに使用します。
- `IndexVaultFile` を TRUE に設定すると、.hcst ファイルは、ファイルが PASSTHRU に設定されている場合、対応するボールト・ファイルを指す weblayout に配置されます。
- この構成変数の変更には、サーバーの再起動が必要です。
- *.hcst* ファイルの内容は、*redirectionfile_template.htm* の内容によって制御されます。
- デフォルトは、FALSE です。
- この実装には、次のような制限があります。
  - `GET_FILE` サービスを使用してファイルが配信されるため、PDF の強調表示またはバイト・サービング機能は使用できません。これは、Web サーバーの再構成またはテンプレートの上書きによって解決できます。
  - 単純なテンプレートが使用されるため、ブラウザの「戻る」ボタンが機能せず、レイアウトの変更が生じることがあります。
  - .hcst ファイルが weblayout に含まれているため、ファイル数は減少しません。ボールト・ファイルが大きい場合、ディスク領域をいくらか節約できます。

場所

<*install_dir*>/config/config.cfg

例

```
IndexVaultFile=true
```

関連項目

– IndexableRenditions（6-351 ページ）
– IsAllowIndexRenditions（6-373 ページ）
– IsAllowIndexVault（6-374 ページ）

INITIALCOLID

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Folders および WebDAV（5-26 ページ）

説明

最初のコレクション ID を初期化するカウンタを定義します。

- これは、サーバーの初期のフォルダ識別子を定義します（100 万の単位）。
- インストールの際に WebDAV 機能が有効な場合、この変数は Folders インストール・プログラムによって自動的に設定されます。
- デフォルトは、0 です。

場所

<*install_dir*>/custom/folders/folders_environment.cfg

例

```
InitialColID=0
```

INSTANCEDESCRIPTION

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

インスタンスの説明を定義します。

- インスタンスの説明は、コンテンツ・サーバー・インタフェースで使用されます。

場所

- 「System Properties」→「Server」タブ→「Instance Description」
- <*install_dir*>/config/config.cfg

例

次は構成エントリの例です。

```
InstanceDescription=Master_on_Server1
```

スクリプトとして、サーバー・インスタンスの説明を文字列で返します。

```
<$InstanceDescription$>
```

関連項目

– IDC_Name（6-326 ページ）
– InstanceMenuLabel（6-367 ページ）

INSTANCEMENULABEL

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

Windows の「スタート」メニューで使用されるコンテンツ・サーバー・インスタンスのラベルを定義します。

- デフォルト値はありません。

場所

- 「System Properties」→「Server」タブ→「Instance Menu Label」
- <*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
InstanceMenuLabel=Master_on_Server1
```

関連項目

- IDC_Name（6-326 ページ）
- InstanceDescription（6-366 ページ）

IntradocDir

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

コンテンツ・サーバーのインストール・ディレクトリのパスを定義します。

- デフォルト値はありません。

場所

- Content Server: <*install_dir*>/bin/intradoc.cfg
- 管理サーバー : <*install_dir*>/admin/bin/intradoc.cfg

例

次は構成エントリの使用例です。

```
IntradocDir=c:/stellent/
```

IntradocRealm

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Web サーバー（5-42 ページ）
- 外部セキュリティ（5-37 ページ）

説明

コンテンツ・サーバーまたは Web サーバー・プラグインが Basic 認証にチャレンジするときに使用するレルムを定義します。

- ログイン・ダイアログで表示される場合、ブラウザでは通常、レルムのことをリソースと呼びます。
- NTLM は、レルムを使用しません。そのため、認証は常に Microsoft ネットワーク全体に対して行われます。
- デフォルトは、*IDC Security* です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IntradocRealm=main
```

関連項目

- NtlmSecurityEnabled（6-555 ページ）
- DefaultAuth（6-131 ページ）

IntradocServerHostName

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

コンテンツ・サーバーへのソケット接続をオープンするときに使用するホスト名を定義します。

- この設定は、Web サーバーからのコールをサーバーへのリクエストに変換する CGI コードによって使用されます。
- デフォルト値は、*localhost* です。

注意：サービスをリモートでコールするには、`IntradocServerHostName` を IP または DNS に設定する必要があります。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IntradocServerHostName=IP
```

関連項目

- HttpServerAddress（6-322 ページ）
- IdcAdminServerHostName（6-328 ページ）

IntradocServerPort

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

Web サーバーのフィルタまたは他のアプリケーションがコンテンツ・サーバーとの対話に使用するポートを定義します。

- この設定は、Web サーバーへのコールをコンテンツ・サーバーへのリクエストに変換する CGI コードに対して、このポートを使用してコンテンツ・サーバーと対話するように指定します。コンテンツ・サーバーが停止および開始した後に、Web サーバーが停止および開始された場合、CGI コードによってこの値が自動的に選択されます。
- IIS では、CGI コードは、*iis_idc_cgi.dll* ISAPI 拡張機能によって実装されます。
- デフォルトは、4444 です。

注意：サービスをリモートでコールするには、サーバー・ポートを定義する必要があります。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IntradocServerPort=5550
```

ISACTIVETRACE

タイプおよび使用方法

❖ グローバル関数（4-21 ページ）

❖ デバッグ（5-20 ページ）

説明

コアでセクションがトレースされているかどうかをチェックします。

パラメータ

なし。

出力

アクティブなトレース結果を返します。

例

```
<$isActiveTrace()$>
```

関連項目

– isVerboseTrace（6-452 ページ）

IsAllowIndexRenditions

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）

説明

TRUE に設定すると、ドキュメントの追加のレンディションを許可されます。これは、索引付け可能なボールト・レンディションまたは Web で表示可能なレンディションがない場合にのみ使用されます。

許可されるレンディション・タイプは、`IndexableRenditions` にリストされます。

場所

<*install_dir*>/config/config.cfg

関連項目

– IndexableRenditions（6-351 ページ）

IsAllowIndexVault

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）

説明

このエントリは、IndexVaultFile が有効でない（FALSE に設定されている）場合にのみ有効です。

IsAllowIndexVault を TRUE に設定すると、Web で表示可能なレンディションが適切なフォーマット・タイプでない場合、ボールト・ファイルを索引付けできます。選択されたボールト・ファイルは、常にプライマリ・レンディションです。

場所

<*install_dir*>/config/config.cfg

関連項目

– IndexVaultFile（6-363 ページ）

ISAUTOARCHIVER

タイプおよび使用方法

- 構成変数（4-3 ページ）
- アーカイバ（5-3 ページ）

説明

コンテンツ・アイテムの自動インポートまたは転送を有効にします。

- TRUE に設定すると、コンテンツ・サーバーは、アーカイブを自動的にインポートおよび転送します。
- FALSE に設定すると、アーカイブはコンテンツ・サーバーによって自動的にインポートまたは転送されません。
- この設定は、NoAutomation（6-542 ページ）設定によって上書きされます。
- デフォルトは、TRUE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IsAutoArchiver=false
```

関連項目

– NoAutomation（6-542 ページ）

IsAutoNumber

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

コンテンツ ID の自動採番を有効にします。

- TRUE に設定すると、チェックイン時に各新規コンテンツ・アイテムにコンテンツ ID（*dDocName*）が自動的に割り当てられます。
- FALSE に設定した場合は、ユーザーが各新規コンテンツ・アイテムのコンテンツ ID を指定する必要があります。
- デフォルトは、FALSE です。

場所

- 「System Properties」→「Options」タブ→「Automatically assign a Content ID on check in」
- 「Admin Server」→「General Configuration」→「Automatically assign a content ID on check in」
- <*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IsAutoNumber=true
```

関連項目

– AutoNumberPrefix（6-51 ページ）

IsAutoQueue

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Inbound Refinery（5-11 ページ）

説明

Inbound Refinery による変換後のコンテンツ・アイテムの処理を有効にします。

- TRUE に設定すると、Inbound Refinery によって変換された後、コンテンツ・アイテムはコンテンツ・サーバーによって処理されます。
- FALSE に設定すると、Inbound Refinery によって変換された後、コンテンツ・アイテムはコンテンツ・サーバーによって処理されません。
- この設定は、NoAutomation（6-542 ページ）設定によって上書きされます。
- デフォルトは、TRUE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IsAutoQueue=false
```

関連項目

– NoAutomation（6-542 ページ）

IsAutoSearch

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）

説明

索引付けおよびアーカイブのエクスポートの自動化を有効にします。

- TRUE に設定すると、コンテンツ・アイテムの索引付けまたはアーカイブへのドキュメントのエクスポートが、コンテンツ・サーバーによって自動的に実行されます。
- FALSE に設定すると、コンテンツ・アイテムの索引付けまたはアーカイブへのドキュメントのエクスポートが、コンテンツ・サーバーによって自動的に実行されません。
- この設定は、NoAutomation（6-542 ページ）設定によって上書きされます。
- デフォルトは、TRUE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IsAutoSearch=true
```

関連項目

– NoAutomation（6-542 ページ）

isCHECKIN

タイプおよび使用方法

- ページ表示変数（4-25 ページ）
- ページ表示（5-30 ページ）

説明

現在のページがチェックイン・ページかどうかを指定します。

- TRUE に設定されている場合、現在のページはチェックイン・ページとして識別されます。
- FALSE に設定されている場合、現在のページはチェックイン・ページとして識別されません。

出力

- 現在のページがチェックイン・ページの場合は、TRUE を返します。
- 現在のページがチェックイン・ページでない場合は、FALSE を返します。

例

ページの先頭で次のように設定します。

```
isCheckin=1
```

スクリプトとしての使用例を次に示します。

```
<$if isCheckin or isUpdate or isQuery$>
    <$isFormSubmit = 1$>
    <$if not isQuery$>
        <$isEditMode = 1$>
        <$formName = "Checkin"$>
    <$endif$>
    <$else$>
        <$isFormSubmit = ""$>
        <$isEditMode = ""$>
<$endif$>
```

ISCHECKINPREAUTHED

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- 内部セキュリティ（5-36 ページ）
- ユーザー（5-40 ページ）

説明

セキュリティ・トークンの取得により、現在のチェックインがチェックイン・アプリケーションによって事前認可済かどうかをチェックします。

出力

- チェックインが事前認可済の場合は、TRUE を返します。
- チェックインが事前認可済でない場合は、FALSE を返します。

例

チェックインが事前認可済かどうかをチェックする例を次に示します。

```
<$IsCheckinPreAuthed$>
```

ISCOMPONENTENABLED

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

定義されたコンポーネントが有効かどうかをチェックします。

パラメータ

1 つのパラメータを取ります。

- パラメータは、コンポーネント名のみです。

出力

- 定義されたコンポーネントが有効な場合は、TRUE を返します。
- 有効でない場合は、FALSE を返します。

例

```
<$isComponentEnabled("Folders")$>
```

ISCONTENTREFINERYPRESENT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Inbound Refinery（5-11 ページ）

説明

Inbound Refinery がインストールされているかどうかを指定します。

- 初期のコンテンツ・サーバーのインストール中に、この値は FALSE に設定されます。
- Inbound Refinery のインストール中に、この値は TRUE に設定されます。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IsContentRefineryPresent=true
```

ISCONTRIBUTOR

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

ユーザーがコントリビュータかどうかを評価するために使用します。通常、ページに特殊リンクを表示するかどうかを決定するために使用します（たとえば、std_page では、Content Manager リンクを表示するかどうかを決定します）。

出力

- ユーザーがコントリビュータである場合は、TRUE を返します。
- ユーザーがコントリビュータでない場合は、FALSE を返します。

例

ユーザーがコントリビュータかどうかの評価に使用します。

```
<$if IsContributor$>
```

ISCRITERIASUBSCRIPTION

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

コンテンツ・アイテムのサブスクリプションが、コンテンツ ID（*dDocName*）ベースではなく、基準ベースかどうかを評価します。

出力

- サブスクリプションが基準ベースの場合は、TRUE を返します。
- 特定のコンテンツ・アイテムのサブスクリプションである場合は、FALSE を返します。

例

サブスクリプションが基準ベースかどうかを評価します。

```
<$IsCriteriaSubscription$>
```

ISCURRENTNAV

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- ページ表示（5-30 ページ）
- 検索（5-34 ページ）

説明

現在表示されているページが、検索結果ページで「次へ」/「前」ナビゲーションの構築中にループ処理されるページと同じかどうかをチェックします。

出力

- ナビゲーション・ループのページが、現在のページと同じである場合は、TRUE を返します。
- ナビゲーション・ループのページが、現在のページと同じでない場合は、FALSE を返します。

例

現在のページでのナビゲーション・ループの場合、ページ番号がプレーン・テキストとして表示されます。現在のページでのナビゲーション・ループでない場合、ページ番号はハイパーテキスト・リンクとして表示されます。

```
<$loop NavigationPages$>
<$if IsCurrentNav$>
    <$HeaderPageNumber$>
<$else$>
    <a href="<$strRemoveWs(inc("searchapi_navigation_specific_page"))$>">
    <$HeaderPageNumber$></a>
<$endif$>
<$endloop$>
```

ISDOCPAGE

タイプおよび使用方法

- ページ表示変数（4-25 ページ）
- ページ表示（5-30 ページ）

説明

現在のページが、メタデータを表示するページ（検索、チェックインおよびコンテンツ情報ページなど）かどうかを指定します。

- TRUE に設定されている場合、現在のページはドキュメント・ページとして識別されます。
- FALSE に設定されている場合、現在のページはドキュメント・ページとして識別されません。

出力

- ページがドキュメント・ページの場合は、TRUE を返します。
- ページがドキュメント・ページでない場合は、FALSE を返します。

例

ページの先頭で次のように設定します。

```
isDocPage=1
```

スクリプトとしての使用例を次に示します。

```
<$if isDocPage$>
   <!--statement-->
<$endif$>
```

IsDpSubmitErrorFatal

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・プロファイル（5-7 ページ）

説明

コンテンツ・プロファイルとともに使用します。TRUE に設定すると、グローバル・ルール違反が見つかった場合、Content Server は致命的エラーを発行できます。

場所

<*install_dir*>/config/config.cfg

例

```
IsDpSubmitErrorFatal=true
```

関連項目

–『リポジトリ・コンテンツの管理』

ISDYNAMIC

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- ページ表示（5-30 ページ）

説明

ユーザーに動的に表示されるページかどうかをチェックします。

- ユーザーに表示されるほとんどのページは動的です。ただし、一部の静的ページは、変更なしにユーザーに配信されるように設計されています。例として、ゲストのポータル・ページや一部の自動で生成される電子メールの内容などがあります。

出力

- ページがユーザーに動的に表示される場合は、TRUE を返します。
- ページが静的であるか、表示できない場合は、FALSE を返します。

例

ページが動的に表示されるかどうかを評価します。

```
<$if IsDynamic$>
    <a href="<$redirect$>">
<$endif$>
```

IsDynamicConverterEnabled

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Dynamic Converter（5-17 ページ）

説明

OracleDynamic Converter を有効にします。

- 初期のコンテンツ・サーバーのインストール中に、この値は FALSE に設定されます。
- Dynamic Converter のインストール中に、この値は TRUE に設定されます。
- デフォルトは、FALSE です。

場所

- 「System Properties」→「Options」タブ→「Enable Dynamic Converter」
- 「Admin Server」→「General Configuration」→「Enable Dynamic Converter」
- <*install_dir*>/config/config.cfg

例

構成設定として、Dynamic Converter 機能を有効にします。

```
IsDynamicConverterEnabled=true
```

スクリプトとして、Dynamic Converter 機能が有効かどうかを評価します。

```
<$if IsDynamicConverterEnabled and isTrue(IsDynamicConverterEnabled)$>
   <tr>
       <td align="center" width="75">
       <a href="<$HttpCgiPath$>?IdcService=GET_TEMPLATE_CONVERSIONS">
       <img src="<$HttpImagesRoot$><$admin_conversion_image$>" border="0"
       width="<$adminImageWidth$>" height="<$adminImageHeight$>"></a>
       </td>
       <td width="10"> </td>
       <td width="245"><span class=largeTableEntry>
       <a href="<$HttpCgiPath$>?IdcService=GET_TEMPLATE_CONVERSIONS">
       <$lc("wwTemplateConversions")$></span></a></td>
   </tr>
<$endif$>
```

ISEDITMODE

タイプおよび使用方法

- ページ表示変数（4-25 ページ）
- ページ表示（5-30 ページ）

説明

現在のページのメタデータ・フィールドが編集可能かどうかを指定します。

- この変数は、チェックイン・ページおよびコンテンツ情報の更新ページで設定されます。
- TRUE に設定されている場合、現在のページのメタデータ・フィールドを編集できます。
- FALSE に設定されている場合、現在のページのメタデータ・フィールドは編集できません。

出力

- 現在のページのメタデータ・フィールドを編集できる場合は、TRUE を返します。
- 現在のページのメタデータ・フィールドを編集できない場合は、FALSE を返します。

例

ページの先頭で次のように設定します。

```
isEditMode=1
```

スクリプトとしての使用例を次に示します。

```
<$if isCheckin or isUpdate or isQuery$>
    <$isFormSubmit = 1$>
    <$if not isQuery$>
        <$isEditMode = 1$>
        <$formName = "Checkin"$>
    <$endif$>
    <$else$>
        <$isFormSubmit = ""$>
        <$isEditMode = ""$>
<$endif$>
```

ISEDITREV

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・アイテム（5-6 ページ）
- ワークフロー（5-44 ページ）

説明

現在のリビジョンが、ワークフローの Editor ステップかどうかをチェックします。

- TRUE に設定されている場合、レビューアはリビジョンをチェックアウト、編集およびチェックインできます。
- この変数は、ワークフロー管理ツールで、ワークフロー・ステップを Editor ステップとして定義すると設定されます。この変数は、チェックイン・ページで評価されます。

出力

- 現在のリビジョンが、ワークフローの Editor ステップである場合は、TRUE を返します。
- 現在のリビジョンが、ワークフローの Editor ステップでない場合は、FALSE を返します。

例

ワークフローの詳細の例を次に示します。

```
<$if IsEditRev$>
   addCheckinValue("IdcService", "WORKFLOW_CHECKIN");
if (form.isFinished.checked)
   addCheckinValue("isFinished", form.isFinished.value);
<$else$>
   addCheckinValue("IdcService", "CHECKIN_SEL");
<$endif$>
```

ISEXCLUDED

タイプおよび使用方法

- 共通フィールド表示変数（4-27 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドをページから完全に除外します。

出力

- メタデータ・フィールドがページから除外される場合は、TRUE を返します。
- メタデータ・フィールドがページから除外されない場合は、FALSE を返します。

例

メタデータ・フィールドをページから除外します。

```
<$isExcluded=1$>
```

次の例では、現在のメタデータ・フィールドの *isExcluded* のアクティブな値に応じて、カスタム変数を設定します。

```
<$isCustomExcluded = getValue("#active", fieldName & ":isExcluded")$>
$isCustomRelocated = getValue("#active", fieldName & ":isRelocated")$>
<$if isCustomExcluded or (isCustomRelocated and not isRelocated) or isExcluded
or (isFieldHidden and not isFormSubmit)$>
    <$isFieldExcluded = 1$>
<$endif$>
```

関連項目

- isFieldExcluded（6-397 ページ）
- isHidden（6-406 ページ）
- isInfoOnly（6-408 ページ）
- isRelocated（6-432 ページ）
- optionListScript（6-568 ページ）

ISEXTERNALUSER

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- ユーザー（5-40 ページ）

説明

ユーザーが外部システム（プロキシ・サーバーや LDAP システムなど）からコンテンツ・サーバーにアクセスしているかどうかをチェックします。

出力

- ユーザーが外部システムからコンテンツ・サーバーにアクセスしている場合は、TRUE を返します。
- ユーザーがコンテンツ・サーバーに直接アクセスしている場合は、FALSE を返します。

例

ユーザーが外部の場所にいるかどうかをチェックします。

```
<$IsExternalUser$>
```

ISFAILEDCONVERSION

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・アイテム（5-6 ページ）
- Inbound Refinery（5-11 ページ）

説明

Inbound Refinery によるコンテンツ・アイテムの変換が失敗したかどうかをチェックします。

出力

- 変換処理が失敗した場合は、TRUE を返します。
- 変換の失敗が検出されなかった場合は、FALSE を返します。

例

次の例では、変換処理が不完全な場合、テキストを表示します。

```
<$if IsFailedConversion$>
    <p><font face="arial" size="2">
    The Refinery was unable to complete the conversion process.</p>
<$endif$>
```

ISFAILEDINDEX

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・アイテム（5-6 ページ）
- 索引付け（5-28 ページ）

説明

インデクサによるコンテンツ・アイテムの索引付けが失敗したかどうかをチェックします。

出力

- インデクサがコンテンツ・アイテムを索引付けできなかった場合は、TRUE を返します。
- コンテンツ・アイテムが正常に索引付けされた場合は、FALSE を返します。

例

次の例では、コンテンツ・アイテムが索引付けされなかった場合、テキストを表示します。

```
<$if IsFailedIndex $>
   <p><font face="arial" size="2">
   Unable to index content item.</p>
<$endif$>
```

isFALSE

タイプおよび使用方法

❖ グローバル関数（4-21 ページ）

❖ Idoc スクリプト（5-27 ページ）

説明

文字列または式が FALSE になるかどうかをチェックします。

パラメータ

1 つのパラメータを取ります。

❖ パラメータは、評価される文字列または式のみです。

出力

パラメータが文字列の場合 :

- ❖ 文字列が F、f、N または n で始まるか、"0" の場合は、TRUE を返します。
- ❖ 文字列がこれ以外の文字で始まる場合は、FALSE を返します。

パラメータが文字列でない場合 :

- ❖ 値が 0 の場合は、TRUE を返します。
- ❖ 値が 0 でない場合は、FALSE を返します。

例

文字列 "false" を評価し、TRUE (1) を返します。

```
<$isFalse("false")$>
```

整数の 5 は 1 より大きいと評価し、FALSE (0) を返します。

```
<$isFalse(5>1)$>
```

式の結果を 0 として評価し、TRUE (1) を返します。

```
<$isFalse(1-1)$>
```

文字列が等しいかどうかを比較する文を true として評価し、FALSE (0) を返します。

```
<$isFalse(strEquals("abc","abc"))$>
```

関連項目

– isTrue（6-444 ページ）

ISFIELDEXCLUDED

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドが除外されることを指定します。

出力

- フィールドが除外される場合は、TRUE を返します。
- フィールドが除外されない場合は、FALSE を返します。

例

メタデータ・フィールドが除外されるかどうかを判定します。

```
<$if isCustomExcluded or (isCustomRelocated and not isRelocated) or isExcluded
or (isFieldHidden and not isFormSubmit)$>
    <$isFieldExcluded = 1$>
<$endif$>
```

関連項目

- isExcluded（6-392 ページ）
- isFieldHidden（6-398 ページ）
- isFieldInfoOnly（6-399 ページ）
- isFieldMemo（6-400 ページ）

ISFIELDHIDDEN

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドが非表示であることを指定します。

出力

- フィールドが非表示の場合は、TRUE を返します。
- フィールドが非表示でない場合は、FALSE を返します。

例

メタデータ・フィールドが非表示かどうかを判定します。

```
<$if isHidden or isCustomHidden$>
   <$isFieldHidden = 1$>
<$else$>
   <$isFieldHidden = ""$>
<$endif$>
```

関連項目

- isHidden（6-406 ページ）
- isFieldExcluded（6-397 ページ）
- isFieldInfoOnly（6-399 ページ）
- isFieldMemo（6-400 ページ）

ISFIELDINFOONLY

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドが参考情報フィールドであることを指定します。

出力

- フィールドが参考情報フィールドである場合は、TRUE を返します。
- フィールドが参考情報フィールドでない場合は、FALSE を返します。

例

メタデータ・フィールドが参考情報かどうかを判定します。

```
<$if isInfo or isCustomInfo or isFieldHidden or isInfoOnly$>
   <$isFieldInfoOnly = 1$>
<$else$>
   <$isFieldInfoOnly = ""$>
<$endif$>
```

関連項目

- isInfoOnly（6-408 ページ）
- isFieldExcluded（6-397 ページ）
- isFieldHidden（6-398 ページ）
- isFieldMemo（6-400 ページ）

ISFIELDMEMO

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドがメモ・フィールドであることを指定します。

出力

- フィールドがメモ・フィールドの場合は、TRUE を返します。
- フィールドがメモ・フィールドでない場合は、FALSE を返します。

例

フィールドがメモ・フィールドかどうかを判定します。

```
<@dynamichtml compute_std_entry_type_info@>
<$if not hasOptionList and #active.fieldType like "Memo"$>
    <$isFieldMemo = 1$>
<$else$>
    <$isFieldMemo = ""$>
<$endif$>
<@end@>
```

関連項目

- isFieldExcluded（6-397 ページ）
- isFieldHidden（6-398 ページ）
- isFieldInfoOnly（6-399 ページ）

IsFilePresent

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

現在表示されているページが、コンテンツ情報ページでリビジョン履歴表を作成する際にループ処理されるリビジョンのページかどうかをチェックします。

出力

- ループ処理されるリビジョンが現在のリビジョンと同じ場合は、TRUE を返します。
- ループ処理されるリビジョンが現在のリビジョンと同じでない場合は、FALSE を返します。

例

現在のリビジョンでのループの場合は、リビジョン番号がプレーン・テキストとして表示されます。現在のリビジョンでのループでない場合は、リビジョン番号がアクティブなボタンとして表示されます。

```
<@dynamichtml doc_rev_info@>
    <$if IsFilePresent$>
        <td width=10% align=center><span class=strongHighlight>
        <$dRevLabel$></span></td>
        <td nowrap width=30%><span class=strongHighlight>
        <$dInDate$></span></td>
        <td nowrap width=30%><span class=strongHighlight>
        <$if dOutDate$><$dOutDate$><$else$><$lc("wwNone")$>
        <$endif$></span></td>
        <td width=20%><span class=strongHighlight>
        <$rptDisplayMapValue("StatusList", dStatus)$></span></td>
    <$else$>
        <td width=10%><form action="<$HttpCgiPath$>" method=GET>
        <input type=hidden name=IdcService value="DOC_INFO">
        <input type=hidden name=dID value="<$dID$>">
        <input type=hidden name=dDocName value="<$dDocName$>">
        <input type=submit value="       <$dRevLabel$>      ">
        </form></td>
...
<@end@>
```

ISFORMSPRESENT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

注意：この変数は、PDF Forms が別個の製品であった、4.5 より前のバージョンに適用されます。現在では、PDF Forms は PDF Converter に組み込まれているため、IsContentRefineryPresent（6-382 ページ）変数を使用します。

PDF Forms ソフトウェアがインストールされているかどうかを指定します。

- 初期のコンテンツ・サーバーのインストール中に、この値は FALSE に設定されます。
- PDF Forms のインストール中に、この値は TRUE に設定されます。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IsFormsPresent=false
```

関連項目

– IsContentRefineryPresent（6-382 ページ）

ISFORMSUBMIT

タイプおよび使用方法

- ページ表示変数（4-25 ページ）
- ページ表示（5-30 ページ）

説明

現在のページが、発行可能な HTML フォームかどうかを指定します。

- TRUE に設定されている場合、現在のページは発行可能な HTML フォームです。
- FALSE に設定されている場合、現在のページは発行可能な HTML フォームではありません。

出力

- ページが発行可能な HTML フォームの場合は、TRUE を返します。
- ページが発行可能な HTML フォームでない場合は、FALSE を返します。

例

ページの先頭で次のように設定します。

```
isFormSubmit=1
```

次の例では、ページが発行可能な HTML フォームでない場合、フィールド・キャプションの後にコロンを追加します。

```
<@dynamichtml std_field_caption@>
    <span class=<$fieldCaptionStyle$>>
    <$fieldCaption$><$if not isFormSubmit$><$":"$><$endif$>
    </span>
<@end@>
```

rule:IsFragmentOnly

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Dynamic Converter（5-17 ページ）

説明

特定の強制変換ルールによって、Dynamic Converter にフラグメントが作成されることを指定します。

- フラグメントは、別のコンテンツ・アイテムに含められるコンテンツのピースです。通常、フラグメントには <HTML> タグや <BODY> タグが含まれず、単独で表示されることを目的としていません。
- この変数を 1 に設定すると、指定された強制変換ルールによって、ファイルがフラグメントに変換されます。
- 指定する強制変換ルールは、ForcedConversionRules（6-236 ページ）設定に含める必要があります。
- デフォルトはありません。

場所

<*install_dir*>/config/config.cfg

例

RuleA 強制変換ルールによって、ファイルがフラグメントに変換されることを指定します。

```
RuleA:IsFragmentOnly=1
ForcedConversionRules=RuleA
```

関連項目

– ForcedConversionRules（6-236 ページ）

ISFULLTEXTINDEXED

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- コンテンツ・アイテム（5-6 ページ）
- 索引付け（5-28 ページ）
- 検索（5-34 ページ）

説明

インデクサがコンテンツ・アイテムをフルテキスト索引付けしたかどうかをチェックします。

出力

- コンテンツ・アイテムがフルテキスト索引付けされている場合は、TRUE を返します。
- コンテンツ・アイテムがフルテキスト索引付けされていない場合は、FALSE を返します。

例

次の例では、コンテンツ・アイテムがフルテキスト索引付けされている場合、指定された URL を提供します。

```
<$if IsFullTextIndexed$>
    <a href="<$redirect$>">
<$endif$>
```

ISHIDDEN

タイプおよび使用方法

- 共通フィールド表示変数（4-27 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドはユーザーに表示されませんが、フィールドは非表示のデータとしてページに含まれます。

出力

- ページでメタデータ・フィールドが非表示の場合は、TRUE を返します。
- ページでメタデータ・フィールドが非表示でない場合は、FALSE を返します。

例

現在のメタデータ・フィールドをユーザーに対して非表示にします。

```
<$isHidden=1$>
```

次の例では、現在のメタデータ・フィールドの *isHidden* のアクティブな値に応じて、カスタム変数を設定します。

```
<$isCustomHidden = getValue("#active", fieldName & ":isHidden")$>
    <$if isHidden or isCustomHidden$>
<$isFieldHidden = 1$>
<$else$>
    <$isFieldHidden = ""$>
<$endif$>
```

関連項目

- isExcluded（6-392 ページ）
- isFieldHidden（6-398 ページ）
- isInfoOnly（6-408 ページ）
- isRelocated（6-432 ページ）
- optionListScript（6-568 ページ）

ISINFO

タイプおよび使用方法

- ページ表示変数（4-25 ページ）
- ページ表示（5-30 ページ）

説明

現在のページが参考情報ページかどうかを指定します。

- TRUE に設定されている場合、現在のページは参考情報ページとして識別されます。
- FALSE に設定されている場合、現在のページは参考情報ページとして識別されません。

出力

- ページが参考情報ページである場合は、TRUE を返します。
- ページが参考情報ページでない場合は、TRUE を返します。

例

ページの先頭で次のように設定します。

```
isInfo=1
```

スクリプトとしての使用例を次に示します。

```
<$if isInfo$>
    <$captionFieldWidth="30%"$>
    <$captionEntryWidth="70%"$>
<$elseif isEditMode$>
    <$captionFieldWidth="20%"$>
    <$captionEntryWidth="80%"$>
<$endif$>
```

ISINFOONLY

タイプおよび使用方法

❖ 共通フィールド表示変数（4-27 ページ）

❖ フィールド表示（5-24 ページ）

説明

メタデータ・フィールドを入力フィールドではなく、参考情報として表示します。

出力

❖ メタデータ・フィールドが参考情報として表示される場合は、TRUE を返します。

❖ メタデータ・フィールドが編集可能な場合は、FALSE を返します。

例

現在のメタデータ・フィールドを参考情報として表示します。

```
<$isInfoOnly=1$>
```

次の例では、現在のメタデータ・フィールドの *isInfoOnly* のアクティブな値に応じて、カスタム変数を設定します。

```
<<$isCustomInfo = getValue("#active", fieldName & ":isInfoOnly")$>
<$if isInfo or isCustomInfo or isFieldHidden or isInfoOnly$>
    <$isFieldInfoOnly = 1$>
<$else$>
    <$isFieldInfoOnly = ""$>
<$endif$>
```

関連項目

– isExcluded（6-392 ページ）
– isFieldInfoOnly（6-399 ページ）
– isHidden（6-406 ページ）
– isRelocated（6-432 ページ）
– optionListScript（6-568 ページ）

ISINTRANETAUTHONLY

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Web サーバー（5-42 ページ）
- 外部セキュリティ（5-37 ページ）

説明

デフォルトの認可タイプを *intranet* に設定します。

- TRUE に設定すると、Web サーバー・フィルタは、認可タイプを *intranet* とみなします。ユーザーが初めてコンテンツ・サーバーにログインし、CGI URL にアクセスすると、Stellent データベース・セキュリティではなく、NTLM セキュリティが使用されます。
- FALSE に設定すると、認可タイプは推測されません。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IsIntranetAuthOnly=true
```

ISJAVA

タイプおよび使用方法

- 設定可能変数（4-34 ページ）
- デバッグ（5-20 ページ）

説明

コンテンツ・サーバー Web ページのローカル・データを表示します。

- この変数は、ページにフラグとして設定するか、サービス・コールのパラメータとして設定できます。

出力

- TRUE に設定すると、DataBinder のローカル・データを返します。

例

コンテンツ・サーバー Web ページのコードに含まれている場合、ページのローカル・データが表示されます。

```
<$IsJava=1$>
```

コンテンツ・サーバーの URL に含まれている場合、「New Checkin」ページのローカル・データが表示されます。

```
http://myinstance.com/idcplg/idc_cgi_isapi-<instance>.dll?IdcService=
CHECKIN_NEW_FORM&IsJava=1
```

IsJdbc

タイプおよび使用方法

- 構成変数（4-3 ページ）
- データベース（5-18 ページ）

説明

Java Database Connectivity（JDBC）を有効にします。

- この設定は、ネットワーク・エラーによってシステムをデータベースに接続できない場合を除き、コンテンツ・サーバーのインストール中に自動的に構成されます。データベースが変更された場合、またはコンテンツ・サーバーのインストール中にシステムがデータベースを検出できなかった場合は、この設定を手動で変更する必要があります。
- Microsoft Access または MSDE のランタイム・バージョンを使用する場合は、設定するデータベース構成オプションはありません。

場所

- 「System Properties」→「Database」タブ→「Use Java Database Connectivity」
- <*install_dir*>/config/config.cfg

例

構成設定として、Java Database Connectivity を無効にします。

```
IsJdbc=false
```

スクリプトとして、構成設定の値を返します。

```
<$IsJdbc$>
```

関連項目

- JdbcConnectionString（6-459 ページ）
- JdbcDriver（6-460 ページ）

IsJdbcLockTrace

タイプおよび使用方法

- 構成変数（4-3 ページ）
- データベース（5-18 ページ）
- デバッグ（5-20 ページ）

説明

データベース・ロック・アクティビティに関するトレース情報をコンテンツ・サーバー・コンソールにダンプします。

- SQL Server および Oracle データベースでのみ使用されます。
- Solaris では、この出力は、<*install_dir*>/etc/log ディレクトリのログ・ファイルでも表示されます。
- デフォルトは、空の文字列です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IsJdbcLockTrace=true
```

関連項目

– IsJdbcQueryTrace（6-413 ページ）

注意：この変数は、Content Server 7.0 リリース以降では不要になりました。この機能は、「Admin Applets」の「System Audit Information」ページで利用できるようになりました。

ISJDBCQUERYTRACE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- データベース（5-18 ページ）
- デバッグ（5-20 ページ）

説明

データベース問合せアクティビティに関するトレース情報をコンテンツ・サーバー・コンソールにダンプします。

- SQL Server および Oracle データベースでのみ使用されます。
- Solaris では、この出力は、<*install_dir*>/etc/log ディレクトリのログ・ファイルでも表示されます。
- デフォルトは、空の文字列です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IsJdbcQueryTrace=true
```

関連項目

– IsJdbcLockTrace（6-412 ページ）

注意：この変数は、Content Server 7.0 リリース以降では不要になりました。この機能は、「Admin Applets」の「System Audit Information」ページで利用できるようになりました。

ISJSPSERVERENABLED

タイプおよび使用方法

- 構成変数（4-3 ページ）
- JSP（5-29 ページ）

説明

Java Server Page 機能を有効にします。

- Java Server Page のサポートにより、開発者は、標準の Stellent コンポーネント・アーキテクチャではなく、Java Server Page を使用して、Content Server のコンテンツ、ResultSet、パーソナライズおよびセキュリティの定義、事前定義済の変数および構成設定にアクセスし、変更できます。実行可能なコンテンツとしてコンテンツ・サーバーにある Stellent サービスおよび Idoc スクリプト関数も Java Server Page から実行できます。
- デフォルトは、FALSE です。

場所

- 「System Properties」→「Server」タブ→「Execute Java Server Page (Jsp)」
- 「Admin Server」→「General Configuration」→「Enable Java Server Page (Jsp)」
- <*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IsJspServerEnabled=true
```

関連項目

– JspEnabledGroups（6-468 ページ）

ISLAYOUTENABLED

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

特定の Content Server レイアウトがインストールされ、有効であるかどうかを確認するために使用します。

パラメータ

1 つのパラメータを取ります。

- レイアウトの ID

例

std_resources.htm ファイルでの例を次に示します。

```
<$if isLayoutEnabled("Trays")$><$do Publish=1$><$endif$>
```

ISLOCALSEARCHCOLLECTIONID

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- コンテンツ・アイテム（5-6 ページ）
- 検索（5-34 ページ）

説明

コンテンツ・アイテムがローカル検索コレクションにあるかどうかをチェックします。

- ローカル検索コレクションでコンテンツ・アイテムのコンテンツ ID を検索します。

出力

- コンテンツ・アイテムがローカル・コレクションにある場合は、TRUE を返します。
- コンテンツ・アイテムがローカル・コレクションにない場合は、FALSE を返します。

例

ローカル・コレクションからのコンテンツ・アイテムかどうかを評価します。

```
<@dynamichtml searchapi_define_result_doc_parameters@>
<$exec IsLocalSearchCollection="1"$>
<$if not IsLocalSearchCollectionID$>
   <!--Collection has external ID-->
<$exec IsLocalSearchCollection=""$>
```

IsLoggedIn

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- 内部セキュリティ（5-36 ページ）
- ユーザー（5-40 ページ）

説明

現在のユーザーがコンテンツ・サーバーにログインしているかどうかをチェックします。

出力

- ユーザーがログインしている場合は、TRUE を返します。
- ユーザーがログインしていない場合は、FALSE を返します。

例

機能を実行する前に、ユーザーがログインしているかどうか、および電子メール・アドレスがあるかどうかをチェックします。

```
<@dynamichtml subscription_action_script@>
function allowSubscription(form)
    {<$if IsLoggedIn$>
    <$if IsUserEmailPresent$>
    <$else$>
    …}
<@end@>
```

IsMac

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- クライアント（5-4 ページ）

説明

クライアントのブラウザが Macintosh オペレーティング・システムで実行されているかどうかをチェックします。

出力

- クライアントのブラウザが Macintosh で実行されている場合は、TRUE を返します。
- クライアントのブラウザが Macintosh で実行されていない場合は、FALSE を返します。

例

次の例では、ブラウザが Macintosh で実行されている場合、別の URL にリダイレクトします。

```
<$if IsMac$>
   <a href="<$redirect$>">
<$endif$>
```

IsMaxRows

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- ページ表示（5-30 ページ）

説明

「Work In Progress」ページまたは「Report 」ページで、結果の数が MaxQueryRows（6-525 ページ）設定の許容値を超過しているかどうかをチェックします。

出力

- 結果の数が、許容される行数を超えている場合は、TRUE を返します。
- 結果の数が、許容される行数より少ない場合は、FALSE を返します。

例

次の例では、最大行数を超過した場合、エラー・メッセージを返します。

```
<$elseif IsMaxRows$>
<table border=0 cellpadding=1 cellspacing=1 width="100%">
    <tr>
    <span class=smallHighlight><$lc("wwOutputLimitedByMaxRows")$>
    </span>
    </tr>
</table>
```

ISMULTIOPTION

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドのオプション・リストで、複数値の選択が可能であることを指定します。

出力

- フィールドが複数選択のオプション・リストの場合は、TRUE を返します。
- フィールドが複数選択のオプション・リストでない場合は、FALSE を返します。

例

次の例では、フィールドが複数選択のオプション・リストであることを指定します。

```
<$isMultiOption=1$>
```

オプション・リストのタイプを判断します。

```
<$if #active.fieldOptionListType like "*multi*"$>
    <$exec isMultiOption=1$>
<$elseif #active.fieldOptionListType like "access*"$>
    <$exec isAccessList=1$>
    <$exec isStrictList=1$>
<$elseif not (#active.fieldOptionListType like "*combo*")$>
    <!--Strict choice-->
    <$exec isStrictList=1$>
<$endif$>
```

関連項目

– isStrictList（6-439 ページ）

ISMULTIPAGE

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- ページ表示（5-30 ページ）
- 検索（5-34 ページ）

説明

検索結果に複数のページが必要かどうかをチェックします。

- この変数は、ResultCount 入力変数で定義される 1 ページ当たりの行数に左右されます（デフォルトは 25）。

出力

- 検索結果が、1 ページ当たりの許容行数を超過する場合は、TRUE を返します。
- 検索結果が、1 ページ当たりの許容行数より少ない場合は、FALSE を返します。

例

行数を評価し、複数のページが必要かどうかを判断します。

```
<$IsMultiPage$>
```

isNew

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・アイテム（5-6 ページ）
- ページ表示（5-30 ページ）

説明

新規のコンテンツ・アイテムかリビジョンかをチェックします。

- この変数は、CHECKIN_NEW_FORM および CHECKIN_SIMILAR_FORM 標準サービスによって TRUE に設定されます。

出力

- 新規のコンテンツ・アイテムの場合は、TRUE を返します。
- コンテンツ・アイテムが、既存コンテンツ・アイテムのリビジョンである場合は、FALSE を返します。

例

次の例では、新規のコンテンツ・アイテムである場合、指定されたサービスを実行します。

```
<$if isNew$>
    <input type=hidden name=IdcService value="CHECKIN_NEW">
<$endif$>
```

次の例では、コンテンツ・アイテムがリビジョンである場合、元のコンテンツ・アイテムの作成者を使用します。

```
<$if not isNew$>value="<$dDocAuthor$>"<$endif$>
```

次の例では、新規のコンテンツ・アイテムである場合、現在のユーザーのデフォルト・アカウントを使用します。

```
<$if isNew$>
    <$defaultAccounts$>
<$endif$>
```

IsNotLatestRev

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

チェックインするリビジョンが、最後のリビジョンかどうかをチェックします。

- これは必ずしも最後にリリースされるリビジョンである必要はありません。

出力

- チェックインするコンテンツ・アイテムが、最新のリビジョン以外の場合は、TRUE を返します。
- チェックインするコンテンツ・アイテムが、最新のリビジョンである場合は、FALSE を返します。

例

最新のリビジョンかどうかをチェックします。

```
<$IsNotLatestRev$>
```

関連項目

- ClientControlled（6-74 ページ）
- HasLocalCopy（6-284 ページ）

IsNotSyncRev

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

リビジョン ID（dID）を比較して、クライアント・コンピュータ上のファイルが最新のリビジョンと一致するかどうかをチェックします。

- 通常、この変数は、コンテンツ・アイテムのローカル・コピーが最新リビジョンに更新されていない場合にエラー・メッセージを表示するために使用されます。
- この変数は、Oracle Desktop などのクライアント側製品とのインタフェースに使用されます。

出力

- リビジョンが一致しない場合は、TRUE を返します。
- リビジョンが一致する場合は、FALSE を返します。

例

最新のリビジョンと一致するかどうかをチェックし、エラー・メッセージを表示します。

```
<$if IsNotSyncRev$>
The local copy of this content item has not been updated to the latest
revision.Use Get Native File or Check Out to update your local copy of
<$dDocName$>.
<$endif$>
```

IsNTCompressionPresent

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Compression（5-18 ページ）

説明

Stellent Compression を有効にします。

- TRUE に設定されている場合、Stellent Compression は有効です。
- FALSE に設定すると、Stellent Compression は無効になります。
- デフォルトは、FALSE です。

場所

<*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
IsNTCompressionPresent=true
```

ISOVERRIDEFORMAT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- フィールド表示（5-24 ページ）

説明

チェックイン時に、ユーザーが別の変換形式を選択できるようにします。

- TRUE に設定すると、チェックイン・ページに、プライマリ・ファイルおよび代替ファイルの「Format」オプション・リストが表示されます。
- デフォルトは、FALSE です。

場所

- 「System Properties」→「Options」タブ→「Allow override format on check in」
- 「Admin Server」→「General Configuration」→「Allow override format on check in」
- <*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
IsOverrideFormat=true
```

スクリプトとして、形式の上書き機能の値を返します。

```
<$IsOverrideFormat$>
```

isPhysicallySplitDir

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

vault ディレクトリと weblayout ディレクトリが異なるファイル・システムにあることを示します。

- TRUE に設定すると、各ディレクトリが異なるファイル・システムにあることをサーバーに認識させます。
- デフォルトは、FALSE です。

場所

<*install_dir*>/bin/intradoc.cfg

例

次は構成エントリの使用例です。

```
isPhysicallySplitDir=true
```

関連項目

– WeblayoutDir（6-909 ページ）

ISPRIMARYUSERBASE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

重要：この設定は、バージョン 7.0 以降では不要です。LDAP プロバイダをチェックする順序の設定は、6-596 ページの「Priority」を参照してください。

LDAP プロバイダが、外部ユーザーを定義するプライマリ LDAP サーバーかどうかを指定します。

- TRUE に設定すると、プロバイダは、以前にコンテンツ・サーバーにログインしていない新規の LDAP ユーザーの認証に使用されます。
- FALSE に設定すると、プロバイダは、そのプロバイダからの資格証明を使用して以前にログインしている LDAP ユーザーのみの認証に使用されます。
- デフォルトは、TRUE です。

場所

- 「Add/Edit LDAP Provider」ページ→「Primary User Base」
- <*install_dir*>/data/providers/*provider_name*/provider.hda

例

次は構成エントリの使用例です。

```
IsPrimaryUserBase=false
```

ISPROMPTINGFORLOGIN

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- 外部セキュリティ（5-37 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

ログインを要求するようにコンテンツ・サーバーが設定されているか、ログインがプログラムで処理されるかをチェックします。

- この変数は、Cookie ログイン・ページなどで、ユーザーが未ログインであるために最後のリクエストが失敗した場合に TRUE に設定されます。

出力

- コンテンツ・サーバーがログインを要求するように設定されている場合は、TRUE を返します。
- ログインがプログラムで処理されている場合は、FALSE を返します。

例

ログインを要求するようにサーバーが設定されているかどうかを評価します。

```
<$IsPromptingForLogin$>
```

IsProxiedServer

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

コンテンツ・サーバーがプロキシ・サーバーであることを指定します。

- TRUE に設定すると、コンテンツ・サーバーはプロキシ・サーバーです。
- マスター・コンテンツ・サーバーでは、デフォルトはありません。
- プロキシ・コンテンツ・サーバーでは、デフォルトは TRUE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
IsProxiedServer=true
```

ISQUERY

タイプおよび使用方法

- ページ表示変数（4-25 ページ）
- ページ表示（5-30 ページ）

説明

現在のページが検索ページかどうかを指定します。

- TRUE に設定されている場合、現在のページは検索ページとして識別されます。
- FALSE に設定されている場合、現在のページは検索ページとして識別されません。

出力

- ページが検索ページの場合は、TRUE を返します。
- ページが検索ページでない場合は、FALSE を返します。

例

ページの先頭で次のように設定します。

```
isQuery=1
```

スクリプトとしての使用例を次に示します。

```
<$if isCheckin or isUpdate or isQuery$>
    <$isFormSubmit = 1$>
    <$if not isQuery$>
        <$isEditMode = 1$>
        <$formName = "Checkin"$>
    <$endif$>
    <$else$>
        <$isFormSubmit = ""$>
        <$isEditMode = ""$>
<$endif$>
```

isRelocated

タイプおよび使用方法

- 共通フィールド表示変数（4-27 ページ）
- フィールド表示（5-24 ページ）

説明

インクルードを評価する際にローカル変数が TRUE に設定されている場合を除き、メタデータ・フィールドを除外します。

- この変数は、通常、メタデータ・フィールド用のカスタムな場所を設定するときに使用します。ページにフィールドを 1 回以上定義でき、実際に生成されるフィールドの場所は、この変数の値に左右されます。

出力

- ページのその場所にメタデータ・フィールドが含まれる場合は、TRUE を返します。
- ページのその場所からメタデータ・フィールドが除外される場合は、FALSE を返します。

例

現在のメタデータ・フィールドをページに含めます。

```
<$isRelocated=1$>
```

現在のメタデータ・フィールドの *isRelocated* のアクティブな値に応じて、カスタム変数を設定します。

```
<$isCustomExcluded = getValue("#active", fieldName & ":isExcluded")$>
<$isCustomRelocated = getValue("#active", fieldName & ":isRelocated")$>
<$if isCustomExcluded or (isCustomRelocated and not isRelocated) or isExcluded
or (isFieldHidden and not isFormSubmit)$>
    <$isFieldExcluded = 1$>
<$endif$>
```

関連項目

- isExcluded（6-392 ページ）
- isHidden（6-406 ページ）
- isInfoOnly（6-408 ページ）
- optionListScript（6-568 ページ）

ISREPROMPTLOGIN

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ワークフロー（5-44 ページ）

説明

この構成エントリは、ワークフロー・ステップに付加される接尾辞で、新しい認証チャレンジを強制します。

- TRUE に設定すると、この構成変数は、ワークフローの特定のステップでログインを強制します。これは、そのステップのデジタル署名として機能します。
- この変数は、ワークフローの複数のステップに追加できます。

場所

<*install_dir*>/config/config.cfg

例

次の例で、*VPApproval* というレビュー・ステップは、ワークフロー内の処理を続ける前に、デジタル署名を要求します。ワークフローを開始する前に、`isRepromoptLogin` 構成エントリを次のように設定します。

```
VPApproval:isRepromptLogin=true
```

関連項目

–『ワークフロー・インプリメンテーション・ガイド』

IsRequestError

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- デバッグ（5-20 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

StatusCode（6-756 ページ）変数を評価して、コンテンツ・サーバーにリクエスト・エラーが存在するかどうかをチェックします。

- *StatusCode* が負の数値（-1）に設定されている場合、コンテンツ・サーバーにリクエスト・エラーが存在します。
- リクエスト・エラーが存在する場合、一般的な動作は、現在のページの表示を中断してエラー・ページに置き換えることです。6-2 ページの「abortToErrorPage」を参照してください。

出力

- コンテンツ・サーバーにリクエスト・エラーが存在する場合（`StatusCode = -1`）、TRUE を返します。
- コンテンツ・サーバーにリクエスト・エラーが存在しない場合（*StatusCode* が負の数値以外）、FALSE を返します。

例

リクエストのエラー状態を評価します。

```
<$IsRequestError$>
```

関連項目

- abortToErrorPage（6-2 ページ）
- executeService（6-211 ページ）
- StatusCode（6-756 ページ）

isREQUIRED

タイプおよび使用方法

- フィールド情報変数（4-26 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドの値が必須かどうかを指定します。

- TRUE に設定した場合、メタデータ・フィールドは必須です。
- FALSE に設定した場合、メタデータ・フィールドはオプションです。

出力

- メタデータ・フィールドが必須の場合は、TRUE を返します。
- メタデータ・フィールドがオプションの場合は、FALSE を返します。

例

dDocTitle メタデータ・フィールドを必須フィールドとして定義します。

```
<$fieldName = "dDocTitle", fieldCaption = lc("wwTitle"), isRequired = 1,
fieldType = "BigText", requiredMsg = lc("wwSpecifyTitle")$>
```

関連項目

– requiredMsg（6-650 ページ）

ISSAVEDQUERY

タイプおよび使用方法

- 設定可能変数（4-34 ページ）
- ページ表示（5-30 ページ）
- 検索（5-34 ページ）

説明

現在のユーザーのポータル・ナビゲーション・バーに、問合せが保存されているかどうかをチェックします。

出力

- 現在の問合せが保存されている場合は、TRUE を返します。
- 現在の問合せが保存されていないか、問合せが見つからない場合は、FALSE を返します。

例

問合せのステータスを評価します。

```
<$IsSavedQuery$>
```

IsSavedWfCompanionFile

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ワークフロー（5-44 ページ）

説明

コンテンツがワークフローに入ったときに作成されるコンパニオン・ファイルのコピーを保存します。通常、コンパニオン・ファイルは、コンテンツが承認されてワークフローを終了したときに削除されます。

この変数を使用するには、説明されたとおりに変数を config.cfg ファイルに挿入し、Content Server を再起動します。ファイルは、<*install_dir*>/data/workflow/saved ディレクトリに保存されます。

場所

<*install_dir*>/config/config.cfg

例

次は構成変数の例です。

```
IsSaveWfCompanionFiles=true
```

IsSoap

タイプおよび使用方法

- 設定可能変数（4-34 ページ）
- デバッグ（5-20 ページ）

説明

コンテンツ・サーバー Web ページの SOAP データを表示します。

- この変数は、ページにフラグとして設定するか、サービス・コールのパラメータとして設定できます。

出力

TRUE に設定すると、基礎となる SOAP コードを返します。

例

コンテンツ・サーバー Web ページのコードに含める場合の例を次に示します。

```
<$IsSoap=1$>
```

コンテンツ・サーバー URL に含める場合の例を次に示します。

```
http://myinstance.com/idcplg/idc_cgi_isapi-<instance>.dll?IdcService=
CHECKIN_NEW_FORM&IsSoap=1
```

isStrictList

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドのオプション・リストで、複数の値を選択できないことを指定します。

出力

- フィールドが複数選択のオプション・リストでない場合は、TRUE を返します。
- フィールドが複数選択のオプション・リストの場合は、FALSE を返します。

例

次の例では、フィールドが複数選択のオプション・リストでないことを指定します。

```
<$isStrictList=1$>
```

オプション・リストのタイプを判断します。

```
<$if #active.fieldOptionListType like "*multi*"$>
    <$exec isMultiOption=1$>
<$elseif #active.fieldOptionListType like "access*"$>
    <$exec isAccessList=1$>
    <$exec isStrictList=1$>
<$elseif not (#active.fieldOptionListType like "*combo*")$>
    <!--Strict choice-->
    <$exec isStrictList=1$>
<$endif$>
```

関連項目

– isMultiOption（6-420 ページ）

IsSubAdmin

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- 内部セキュリティ（5-36 ページ）
- ユーザー（5-40 ページ）

説明

現在のユーザーが、1 つ以上の管理アプリケーションに対する副管理者の権限を持つかどうかをチェックします。

出力

- ユーザーに副管理者の権限がある場合は、TRUE を返します。
- ユーザーに副管理者の権限がない場合は、FALSE を返します。

例

機能を実行する前に、ユーザーがログインしているかどうか、および副管理者の権限があるかどうかをチェックします。

```
<@dynamichtml subscription_action_script@>
    function allowSubscription(form)
    {
    <$if IsLoggedIn$>
    <$if IsSubAdmin$>
    <$else$>
    …
    }
<@end@>
```

IsSun

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- クライアント（5-4 ページ）

説明

クライアントのブラウザが Sun マシンで実行されているかどうかをチェックします。

出力

- クライアントのブラウザが Sun で実行されている場合は、TRUE を返します。
- クライアントのブラウザが Sun で実行されていない場合は、FALSE を返します。

例

次の例では、ブラウザが Sun で実行されている場合、別の URL にリダイレクトします。

```
<$if IsSun$>
   <a href="<$redirect$>">
<$endif$>
```

ISSYSMANAGER

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- 内部セキュリティ（5-36 ページ）
- ユーザー（5-40 ページ）

説明

現在のユーザーに sysmanager ロールがあるかどうかをチェックします。これは、ユーザーが管理サーバーにアクセスできることを意味します。通常、この変数は、特定のナビゲーション・リンクを条件付きで表示するために使用されます。

出力

- sysmanager ロールを持つユーザーの場合は、TRUE を返します。
- ロールを持つユーザーでない場合は、FALSE を返します。

ISTHUMBNAILPRESENT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

サムネイル機能を有効にします。

- TRUE に設定されている場合、サムネイル・プログラムは有効です。
- FALSE に設定されている場合、サムネイル・プログラムは無効です。
- デフォルトは、FALSE です。

場所

<*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
IsThumbnailPresent=true
```

関連項目

- CreatePDFThumbnails（6-111 ページ）
- ImageAlchemyExePath（6-342 ページ）

ISTRUE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- Idoc スクリプト（5-27 ページ）

説明

文字列または式が TRUE になるかどうかをチェックします。

パラメータ

1 つのパラメータを取ります。

- パラメータは、評価される文字列または式のみです。

出力

パラメータが文字列の場合 :

- 文字列が T、t、Y または y で始まるか、"1" の場合は、TRUE を返します。
- 文字列がこれ以外の文字で始まる場合は、FALSE を返します。

パラメータが文字列でない場合 :

- 値が 0 でない場合は、TRUE を返します。
- 値が 0 の場合は、FALSE を返します。

例

文字列 "yes" を評価し、TRUE (1) を返します。

```
<$isTrue("yes")$>
```

整数の 5 は 1 より大きいと評価し、TRUE (1) を返します。

```
<$isTrue(5>1)$>
```

式の結果を 0 として評価し、FALSE (0) を返します。

```
<$isTrue(1-1)$>
```

文字列が等しいかどうかを比較する文を true として評価し、TRUE (1) を返します。

```
<$isTrue(strEquals("abc","abc"))$>
```

関連項目

– isFalse（6-396 ページ）

isUpdate

タイプおよび使用方法

- ページ表示変数（4-25 ページ）
- ページ表示（5-30 ページ）

説明

現在のページがコンテンツ情報更新ページかどうかを指定します。

- TRUE に設定されている場合、現在のページはコンテンツ情報更新ページとして識別されます。
- FALSE に設定されている場合、現在のページはコンテンツ情報更新ページとして識別されません。

出力

- ページがコンテンツ情報更新ページである場合は、TRUE を返します。
- ページがコンテンツ情報更新ページでない場合は、FALSE を返します。

例

ページの先頭で次のように設定します。

```
isUpdate=1
```

スクリプトとしての使用例を次に示します。

```
<$if isCheckin or isUpdate or isQuery$>
    <$isFormSubmit = 1$>
    <$if not isQuery$>
        <$isEditMode = 1$>
        <$formName = "Checkin"$>
    <$endif$>
    <$else$>
        <$isFormSubmit = ""$>
        <$isEditMode = ""$>
<$endif$>
```

ISUPLOADFIELDSCRIPT

タイプおよび使用方法

- ページ表示変数（4-25 ページ）
- ページ表示（5-30 ページ）

説明

JavaScript 内でインクルードが使用されていることを指定します。メタデータ・フィールドのアップロード方法を決定するために使用されます。

- TRUE に設定されている場合、JavaScript 内でインクルードが使用されています。
- FALSE に設定されている場合、JavaScript 内でインクルードが使用されていません。

出力

- JavaScript 内でインクルードが使用されている場合は、TRUE を返します。
- JavaScript 内でインクルードが使用されていない場合は、FALSE を返します。

例

JavaScript ラッパーを定義するインクルードを設定します。

```
<@dynamichtml std_upload_java_script@>
    <script language="JavaScript">
    // Java script for uploading.
    <$isUploadFieldScript = 1$>
    <$include std_upload_info_script$>
    <$isUploadFieldScript = ""$>
    </script>
<@end@>
```

スクリプトとしての使用例を次に示します。

```
<$if isUploadFieldScript$>
    <$defaultFieldInclude = "std_file_entry"$>
<$else$>
    <$defaultFieldInclude = "std_nameentry_row"$>
    <$fieldCaptionInclude = "std_field_caption"$>
    <$fieldEntryInclude = "std_file_entry"$>
<$endif$>
```

ISUPLOADSOCKETS

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- 外部セキュリティ（5-37 ページ）
- クライアント（5-4 ページ）

説明

Upload アプレットで、アップロード・ソケットを使用するかどうかを決定するために使用されます。

- これはコンテンツ・サーバーによって設定される内部フラグであり、ユーザーが構成するものではありません。

出力

- Upload アプレットで使用されるアップロード・ソケットが定義されている場合は、TRUE を返します。
- アップロード・ソケットを使用しない場合は、FALSE を返します。

例

N/A

ISUSEREMAILPRESENT

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- ユーザー（5-40 ページ）

説明

現在のユーザーの電子メール・アドレスが定義されているかどうかをチェックします。

出力

- ユーザーの電子メール・アドレスが定義されている場合は、TRUE を返します。
- ユーザーの電子メール・アドレスが定義されていない場合は、FALSE を返します。

例

機能を実行する前に、ユーザーがログインしているかどうか、および電子メール・アドレスがあるかどうかをチェックします。

```
<@dynamichtml subscription_action_script@>
    function allowSubscription(form)
    {
    <$if IsLoggedIn$>
    <$if IsUserEmailPresent$>
    <$else$>
    …
    }
<@end@>
```

isUserOverrideSet

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 内部セキュリティ（5-36 ページ）
- ユーザー（5-40 ページ）

説明

ユーザーが別のユーザーのコンテンツをチェックインできるようにします。

- これはチェックイン・ページの「Author」オプション・リストに影響を与えます。デフォルトでは、チェックインの際に別のユーザーを作成者として指定できるのは管理者のみです。

パラメータ

1 つのパラメータを取ります。

- パラメータは、TRUE または FALSE のみです。

出力

ユーザーの上書きが有効な場合は、TRUE になります。

例

別のユーザーを作成者として指定して、コンテンツ・アイテムをチェックインできるようにします。

```
<$isUserOverrideSet(true)$>
```

関連項目

– ExclusiveCheckout（6-208 ページ）

ISVALIDATEFILE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

Upload アプレットのパラメータとして使用します。この変数は、アップロードするファイルが存在することを確認します。使用するには、std_page.htm リソース・ファイルの std_multiupload_applet_definition インクルードに定義されたアプレット定義を上書きするコンポーネントを作成する必要があります。

例

次は構成エントリの使用例です。

```
isValidateFile=true
```

ISVERBOSETRACE

タイプおよび使用方法

❖ 構成変数（4-3 ページ）

❖ デバッグ（5-20 ページ）

説明

コアでセクションがトレースされているかどうかをチェックします。

❖ 冗長トレースは、詳細レポートを生成します。

出力

❖ トレースが冗長に設定されている場合は、TRUE を返します。

❖ トレースが別のデバッグ・レベルに設定されている場合は、FALSE を返します。

例

```
<$isVerboseTrace()$>
```

関連項目

– isActiveTrace（6-372 ページ）

IsWebServerPagesOnly

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

ExtranetLook コンポーネントとともに使用されます。この変数は、Cookie ベースの認証を使用するか否かを決定します。ExtranetLook 機能の使用の詳細は、『システム設定およびプロセスの管理』を参照してください。

- `IsWebServerPagesOnly` を TRUE に設定すると、Basic 認証が使用され、Cookie ベースのログオン / ログオフ機能は無効になります。カスタマイズされたバージョンの Web サーバー・フィルタが使用されます。
- FALSE に設定すると、Cookie ベースの認証が使用されます。

場所

<*install_dir*>/config/config.cfg

例

```
IsWebServerPagesOnly=FALSE
```

関連項目

- CookieAuthTimeoutInDays（6-107 ページ）
- CookieProxyPassword（6-108 ページ）

ISWINDOWS

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- クライアント（5-4 ページ）

説明

クライアントのブラウザが Windows オペレーティング・システムで実行されているかどうかをチェックします。

出力

- クライアントのブラウザが Windows で実行されている場合は、TRUE を返します。
- クライアントのブラウザが Windows で実行されていない場合は、FALSE を返します。

例

次の例では、ブラウザが Windows で実行されている場合、別の URL にリダイレクトします。

```
<$if IsWindows$>
    <a href="<$redirect$>">
<$endif$>
```

ISWORKFLOW

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- ワークフロー（5-44 ページ）

説明

チェックイン・ページのコンテンツ・アイテムがワークフロー内にあるかどうかをチェックします。

出力

- コンテンツ・アイテムがワークフロー内にある場合は、TRUE を返します。
- コンテンツ・アイテムがワークフロー内にない場合は、FALSE を返します。

IsXml

タイプおよび使用方法

❖ 設定可能変数（4-34 ページ）

❖ デバッグ（5-20 ページ）

説明

TRUE に設定されている場合、HCSP または HCSF ファイルの URL に HTML フォームの RAW ダンプを表示します。これは、リモート・アプリケーションからのフォーム・データの抽出またはフォームの発行に役立ちます。

出力

フォームに含まれる XML データを返します。

関連項目

– IsJava
– IsSoap

INDEXERSETCOLLECTIONVALUE

タイプおよび使用方法

- 構成変数
- 検索

説明

この関数は、Verity 検索インデクサのロジックを変更します。この関数は、`verity_indexer_params_override` インクルード・ファイルでのみ使用されます。索引の収集に関する累積統計を計算できるように設計されています。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、使用するフラグの名前です。
- 2 番目のパラメータは、フラグに起因する値です。

任意の名前および値を使用できます。これらの値は、索引の収集のための次回のリソース・インクルードの実行で使用可能になります。

ISZONEFIELD

タイプおよび使用方法

❖ 構成変数（4-3 ページ）

❖ 検索（5-34 ページ）

説明

Verity Integration で使用されます。Verity Integration は、Content Server とともに使用できるアドオン製品であり、購入することができます。

この関数は、指定されたメタデータ・フィールドが Verity ゾーンに存在する場合、TRUE を返します。ゾーンは、Verity の検索パフォーマンスの最適化に使用されます。フィールドがゾーン内にある場合、検索の問合せの構文がいくらか異なります。

パラメータ

1 つのパラメータを取ります。

❖ チェックするフィールドの名前

JDBCCONNECTIONSTRING

タイプおよび使用方法

- 構成変数（4-3 ページ）
- データベース（5-18 ページ）

説明

ホスト名、ポート番号およびインスタンス名を含む、Java Database Connectivity（JDBC）接続（SQL Server データ・ソース名）を定義します。

- この設定は、ネットワーク・エラーによってシステムをデータベースに接続できない場合を除き、コンテンツ・サーバーのインストール中に自動的に構成されます。データベースが変更された場合、またはコンテンツ・サーバーのインストール中にシステムがデータベースを検出できなかった場合は、この設定を手動で変更する必要があります。
- Microsoft Access または MSDE のランタイム・バージョンを使用する場合は、設定するデータベース構成オプションはありません。
- デフォルトは、*JdbcConnectionString=JDBC:ODBC:SQLForInstance_name* です。

場所

- 「System Properties」→「Database」タブ→「JDBC Connection String」
- <*install_dir*>/config/config.cfg

例

JDBC 接続のパスを定義します。

```
JdbcConnectionString=jdbc:oracle:thin:@hostname:port_number:
instance_name
```

関連項目

– JdbcDriver（6-460 ページ）

JDBCDRIVER

タイプおよび使用方法

- 構成変数（4-3 ページ）
- データベース（5-18 ページ）

説明

Java Database Connectivity（JDBC）のデバイス・ドライバ名を定義します。

- この設定は、ネットワーク・エラーによってシステムをデータベースに接続できない場合を除き、コンテンツ・サーバーのインストール中に自動的に構成されます。データベースが変更された場合、またはインストール中にシステムがデータベースを検出できなかった場合は、この設定を手動で変更する必要があります。
- Microsoft Access または MSDE のランタイム・バージョンを使用する場合は、設定するデータベース構成オプションはありません。
- デフォルト値はありません。

場所

- 「System Properties」→「Database」タブ→「JDBC Driver options/ JDBC Driver Name」
- <*install_dir*>/config/config.cfg

例

JDBC デバイス・ドライバを定義します。

```
JdbcDriver=oracle.jdbc.driver.OracleDriver
```

関連項目

- IsJdbc（6-411 ページ）
- JdbcConnectionString（6-459 ページ）

JDBCPASSWORD

タイプおよび使用方法

- 構成変数（4-3 ページ）
- データベース（5-18 ページ）

説明

SQL Server データベースのパスワードを定義します。

- デフォルト値はありません。

場所

- 「System Properties」→「Database」タブ→「JDBC User Password」
- <*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
JdbcPassword=password
```

関連項目

- JdbcUser（6-463 ページ）
- JdbcPasswordEncoding（6-462 ページ）

JDBCPASSWORDENCODING

タイプおよび使用方法

- 構成変数（4-3 ページ）
- データベース（5-18 ページ）

説明

SQL Server データベースのパスワードを格納するときに使用するエンコーティングのタイプを定義します。

- パスワードの暗号化を無効にするには、この値を `ClearText` に設定します。
- デフォルトは、*Intradoc* です。
- 他の空でない値が割り当てられている場合、エンコーディングは実行されません。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
JdbcPasswordEncoding=ClearText
```

JDBCUSER

タイプおよび使用方法

❖ 構成変数（4-3 ページ）

❖ データベース（5-18 ページ）

説明

SQL Server データベースのユーザー名を定義します。

❖ デフォルトは、*sa* です。

場所

❖ 「System Properties」→「Database」タブ→「JDBC User Name」

❖ <*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
JdbcUser=sa
```

関連項目

– JdbcPassword（6-461 ページ）

JS

タイプおよび使用方法

❖ グローバル関数（4-21 ページ）

❖ 文字列（5-39 ページ）

説明

"..." JavaScript リテラル文字列宣言で使用する文字列を書式設定します。

❖ この関数は、二重引用符を一重引用符に変更するなど、文字列操作を実行します。

パラメータ

1 つのパラメータを取ります。

❖ パラメータは、文字列のみです。

出力

JavaScript 文字列宣言で使用する書式設定された文字列を返します。

例

JavaScript 文字列宣言で使用する文字列 *variablestring* を書式設定します。

```
<$js("variablestring")$>
```

JSFILENAME

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- スキーマ（5-33 ページ）

説明

スキーマで使用されます。非 ASCII 文字が含まれる文字列を、オペレーティング・システムおよび JavaScript で必要な有効なファイル名文字列にエンコードします（エンコーティング関数を実行）。

パラメータ

1 つのパラメータを取ります。

- パラメータは、エンコードする文字列のみです。

出力

エンコードされた文字列を返します。

例

```
<$jsFilename(fileName)$>
```

JSPADMINQUERY

タイプおよび使用方法

- 構成変数（4-3 ページ）
- JSP（5-29 ページ）

説明

Web アプリケーション・ファイルとして使用できるファイルを定義します。

- 通常、この設定は、Stellent JavaBean をデプロイする際に Web アプリケーション・アーカイバ（.war ファイル）を検索する問合せを定義するために使用されます。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
JspAdminQuery=dExtension <matches> war
```

関連項目

– IsJspServerEnabled（6-414 ページ）

JSPDEFAULTINDEXPAGE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- JSP（5-29 ページ）

説明

Web アプリケーションのデフォルト・ページを定義します。

- これは検索順序を定義するカンマ区切りのファイル名のリストです。最初のファイルが存在しない場合は、2 番目のファイルがデフォルトのページになり、2 番目のファイルもない場合は 3 番目などとなります。
- デフォルトは、*index.html,index.htm,index.jsp* です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
JspDefaultIndexPage=index.html,index.htm,index.jsp
```

関連項目

– IsJspServerEnabled（6-414 ページ）

JSPENABLEDGROUPS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- JSP（5-29 ページ）

説明

Java Server Page 機能で有効となるセキュリティ・グループを定義します。

- 通常、Java Server Page は、ホスト・マシンのすべてのリソースに対する完全な権限を持つため、Java Server Page は、特定の権限を持つコントリビュータのみが許可されるセキュリティ・グループに制限することが重要です。

場所

- 「System Properties」→「Server」タブ→「Jsp Enabled Groups」
- 「Admin Server」→「General Configuration」→「Jsp Enabled Groups」
- <*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
JspEnabledGroups=jsp,group1
```

関連項目

– IsJspServerEnabled（6-414 ページ）

JVMCOMMANDLINE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）
- Inbound Refinery（5-11 ページ）

説明

Java 仮想マシン・コマンドラインの実行可能ファイルの場所を定義します。

- デフォルト値はありません。

場所

- Content Server: <*install_dir*>/bin/intradoc.cfg
- 管理サーバー : <*install_dir*>/admin/bin/intradoc.cfg
- Inbound Refinery: <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
JvmCommandLine="c:/jdk1.3.1/bin/java.exe" -classpath $CLASSPATH $STARTUPCLASS
```

LASTENTRYTS

タイプおよび使用方法

- ワークフロー変数（4-39 ページ）
- ワークフロー（5-44 ページ）
- 日付と時間（5-19 ページ）

説明

最後にワークフロー・ステップに入った時間のタイムスタンプを取得します。

- この変数は、条件文の作成に使用できますが、ハードコードまたは変更することはできません。
- 最終エントリ時間は、コンパニオン・ファイルでローカライズされ、キーで維持されます。

  `<step_name>@<workflow_name>.lastEntryTs`

出力

dateCurrent（6-121 ページ）によって定義された書式でタイムスタンプを返します。

例

次のコードは、LastEntry というジャンプを定義し、ステップへの最終エントリから 1 週間を超過している場合、親ワークフローに移動してレビューアに通知します。

```
<$if parseDate(wfCurrentGet("lastEntryTs")) < parseDate(dateCurrent(-7))$>
    <$wfSet("WfJumpName","LastEntry")$>
    <$wfSet("WfJumpTargetStep",wfExit(0,0))$>
    <$wfSet("WfJumpEntryNotifyOff","0")$>
<$endif$>
```

LATESTRELEASEDTIMESTAMPTIMEOUT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Dynamic Converter（5-17 ページ）

説明

変換済のコンテンツ・アイテムのタイムスタンプをチェックする最小間隔（ミリ秒）を定義します。

- キャッシュされた特定の Web で表示可能なファイルの場合、Dynamic Converter は、この設定に指定された時間が経過した後にのみタイムスタンプを再び確認します。
- キャッシュされたコンテンツ・アイテムのタイムスタンプが変更された場合は、Dynamic Converter によってファイルが再変換されます。
- デフォルトは、1500 ミリ秒（1.5 秒）です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
LatestReleasedTimestampTimeout=60000
```

関連項目

– EnableOptimizedLatestReleaseQuerying（6-198 ページ）

LC

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ローカライゼーション（5-29 ページ）

説明

現在のユーザーのロケールに基づいてローカライゼーション文字列の値を取得します。

パラメータ

1 つの必須パラメータおよび任意の数のオプション・パラメータを取ります。

- 最初のパラメータは、ローカライゼーション・キーです（*apLabelHelp* または *wwMyString*）。
- オプションのパラメータは、ローカライズ文字列の中カッコ内の引数に基づいて解決される式です（たとえば、`{1}`）。

注意：ローカライゼーション文字列およびロケールの詳細は、『Using Content Server in International Environments』を参照してください。

出力

- 現在のユーザーのロケール用のローカライゼーション文字列の値を返します。
- 値が見つからなかった場合は、文字列 ID（*wwMyString* など）を返します。

例

検索ページでメタデータ検索演算子のオプションを取得します。

```
<select name="op" >
    <option value="Contains"><$lc("wwContains")$>
    <option value="Matches"><$lc("wwMatches")$>
    <option value="Starts"><$lc("wwStarts")$>
    <option value="Ends"><$lc("wwEnds")$>
    <option selected value="Substring"><$lc("wwSubstring")$>
</select>
```

「ワークフロー開始」電子メール通知の件名行を設定します。ワークフロー（*dWfName*）の名前が *Marketing* の場合、英語の値は "Workflow 'Marketing' has been started" になります。

```
<@dynamichtml wf_started_subject@>
    <$lc("wwWfStarted", dWfName)$>
<@end@>
```

LCCAPTION

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ローカライゼーション（5-29 ページ）

説明

文字列をキャプションにラップする関数です。通常、文字列の右側にコロンが追加されます。ヘブライ語やアラビア語など、右側から左側へ読み取る言語の場合は、文字列の左側にコロンが追加されます。

パラメータ

1 つの必須パラメータを取ります。

- ラップする文字列。
- オプションのパラメータは、ローカライズ文字列の中カッコ内の引数に基づいて解決される式です（たとえば、`{1}`）。

LDAPADMINDN

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

LDAP サーバーに対する読取り権限を持つユーザー名を指定します。

- この設定が未定義の場合、プロバイダは、匿名で LDAP サーバーに接続します。
- LDAP プロバイダが Active Directory と通信している場合は、ユーザー名とパスワードが必要です。ユーザー名は、*DOMAIN¥username* 形式の有効なドメイン・ユーザーである必要があります。

場所

- 「Add/Edit LDAP Provider」ページ→「LDAP Admin DN」
- <*install_dir*>/data/providers/*provider_name*/provider.hda

例

次は構成エントリの使用例です。

```
LdapAdminDN=ldap_admin
```

関連項目

– LdapAdminPassword（6-476 ページ）

LDAPADMINPASSWORD

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

LdapAdminDN（6-475 ページ）設定によって指定された、LDAP サーバーに対する読取り権限を持つユーザー名のパスワードを指定します。

場所

- 「Add/Edit LDAP Provider」ページ→「LDAP Admin Password」
- <*install_dir*>/data/providers/*provider_name*/provider.hda

例

次は構成エントリの使用例です。

```
LdapAdminPassword=idc2
```

関連項目

– LdapAdminDN（6-475 ページ）

LDAPPORT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

LDAP サーバーがリスニングするポート番号を定義します。

- SSL を使用する場合を除き、デフォルトは 389 です。
- SSL を使用する場合（6-878 ページの「UseSecureLdap」を参照）、この値を 636 に設定します。

場所

- 「Add/Edit LDAP Provider」ページ→「LDAP Port」
- <*install_dir*>/data/providers/*provider_name*/provider.hda

例

次は構成エントリの使用例です。

```
LdapPort=636
```

関連項目

– UseSecureLdap（6-878 ページ）

LDAPSERVER

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

LDAP サーバーのホスト名です。

- プロバイダが Active Directory と通信している場合は、プライマリ・ドメイン・コントローラのホスト名です。

場所

- 「Add/Edit LDAP Provider」ページ→「LDAP Server」
- <*install_dir*>/data/providers/*provider_name*/provider.hda

例

次は構成エントリの使用例です。

```
LDAPServer=iplanet_dir
```

LDAPSUFFIX

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

すべての LDAP 操作に使用するルートの接尾辞（ネーミング・コンテキスト）を指定します（*o=company.com* または *dc=company,dc=com* など）。

- コンテンツ・サーバーのロールおよびアカウントへのすべての LDAP グループのマッピングは、このルートで始まります。
- カンマの前後に空白を挿入しないでください。

場所

- 「Add/Edit LDAP Provider」ページ→「LDAP Suffix」
- <*install_dir*>/data/providers/*provider_name*/provider.hda

例

次は構成エントリの使用例です。

```
LdapSuffix=dc=company,dc=com
```

LKDISABLEONREBUILD

タイプおよび使用方法

- Link Manager の構成変数（4-19 ページ）
- 検索（5-34 ページ）

説明

Link Manager 機能とともに使用します。

この変数は、索引付けの再構築サイクル中にリンクの抽出を制御します。

TRUE に設定すると、この変数は、索引の再構築サイクル中に Link Manager によるリンクの抽出を防止します。FALSE に設定すると、リンクは抽出されます。

デフォルトは、FALSE です。

注意 : Link Manager のすべての処理が索引付けサイクル中に実行されるため、ドキュメントの索引付けおよびコレクションの再構築に必要な時間が増加します。

ただし、ほとんどの時間は、コレクションへのドキュメントの索引付けに費やされるため、時間がかかっても目立たないことがあります。所要時間は、含まれるドキュメントのタイプおよびサイズによって異なります。したがって、ファイルを変換する必要がある場合、テキスト・ベース（HTML）ファイルよりも所要時間が長くなります。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
LkDisableOnRebuild=TRUE
```

関連項目

– Link Manager の使用の詳細は、『リポジトリ・コンテンツの管理』を参照してください。

LKDISABLEREFERENCEINFO

タイプおよび使用方法

- Link Manager の構成変数（4-19 ページ）
- 検索（5-34 ページ）

説明

Link Manager 機能とともに使用します。

この変数は、「Content Information」ページの「Resources」セクションを有効または無効にします。TRUE に設定すると、「References」セクションを抑止するように Content Server が構成されます。Link Manager は、プレゼンテーションのリンク参照を計算しません。

FALSE に設定すると、Content Server では、「Content Information」ページに「Show」と「Hide」のトグル・スイッチが表示されます。トグル・スイッチをクリックすると、リンクの参照が表示または非表示になります。

デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
LkDisableReferenceInfo=TRUE
```

関連項目

– Link Manager の使用の詳細は、『リポジトリ・コンテンツの管理』を参照してください。

LKDISALLOWCONVERSIONFORMATS

タイプおよび使用方法

- Link Manager の構成変数（4-19 ページ）
- 検索（5-34 ページ）

説明

Link Manager 機能とともに使用します。

この変数は、リンク抽出処理の際に Link Manager によって無視されるファイル形式のリストを提供します。デフォルトでは、この変数は空です。

リンクを抽出する前に、HtmlExport による変換が必要ないくつかのファイル形式があります（Word など）。ただし、テキスト・ベースのファイル形式（HTML）のリンクは、HtmlExport による変換の必要なしに、Link Manager によって抽出できます。

たとえば、実際にはテキスト・ベースである PHP ファイル（または他のカスタム・フォーマットのファイル）があるとします。このようなファイルでは、Link Manager がリンク抽出処理を行う前に、HtmlExport による変換は必要ではありません。このようなフォーマットは、この構成変数にリストできます。

パラメータ

パラメータは、フォーマットのみです。

場所

<*install_dir*>/config/config.cfg

例

次の例では、完全な MIME タイプのフォーマットを指定すると、様々な変化形が、除外するタイプのリストで制限されます。たとえば、`application/msword` をリストすると、`application/vnd.msword` やその他の変化形は除外されません。この場合は、除外する特定の MIME タイプの変化形をリストに含める必要があります。

```
LkDisallowConversionFormats=application/msword,audio/wav,video/avi
```

次の例では、リストの各 MIME タイプのすべての変化形が除外されます。MIME タイプの短縮形を使用すると、柔軟性が高くなるという点で、より効果的です。

```
LkDisallowConversionFormats=msword,wav,avi
```

重要： Link Manager では、ファイル形式に hcs、htm、image、text、xml、jsp および asp の文字列が含まれるファイルの変換に HtmlExport を使用しません。これらのファイルは、変換の必要なしに Link Manager によって処理されます。現在のシステムのファイル形式および拡張子のマッピングを確認するには、Configuration Manager の「File Formats」ウィンドウを使用します。

関連項目

– Link Manager の使用の詳細は、『リポジトリ・コンテンツの管理』を参照してください。

LkIsSecureSearch

タイプおよび使用方法

- Link Manager の構成変数（4-19 ページ）
- 検索（5-34 ページ）

説明

Link Manager 機能とともに使用します。

この関数は、リンク検索の際に非管理ユーザーのセキュリティ制限チェックを管理するために使用されます。

TRUE に設定すると、Link Manager は、管理対象のリンク検索の実装中に非管理ユーザーに対するセキュリティ・チェックを実行します。これがデフォルトの設定です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
LkIsSecureSearch=TRUE
```

関連項目

– Link Manager の使用の詳細は、『リポジトリ・コンテンツの管理』を参照してください。

LKREEXTRACTONREBUILD

タイプおよび使用方法

- Link Manager の構成変数（4-19 ページ）
- 検索（5-34 ページ）

説明

Link Manager 機能とともに使用します。

この変数は、再構築の際に、以前に索引付けされたドキュメントからのリンクの抽出を制御します。TRUE に設定すると、Link Manager は、再構築の際にシステムにドキュメントが索引付けされるとき、ドキュメントからリンクを抽出します。これがデフォルトの設定です。

FALSE に設定すると、Link Manager は、再構築の際にシステムにドキュメントが索引付けされるとき、ドキュメントからリンクを抽出できません。

注意 : Link Manager のすべての処理が索引付けサイクル中に実行されるため、ドキュメントの索引付けおよびコレクションの再構築に必要な時間が増加します。

ただし、ほとんどの時間は、コレクションへのドキュメントの索引付けに費やされるため、時間がかかっても目立たないことがあります。所要時間は、含まれるドキュメントのタイプおよびサイズによって異なります。したがって、ファイルを変換する必要がある場合、テキスト・ベース（HTML）ファイルよりも所要時間が長くなります。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
LkReExtractOnRebuild=FALSE
```

関連項目

– Link Manager の使用の詳細は、『リポジトリ・コンテンツの管理』を参照してください。

LKREFRESHBATCHSIZE

タイプおよび使用方法

- Link Manager の構成変数（4-19 ページ）
- 検索（5-34 ページ）

説明

リフレッシュ処理中に Site Studio に送信されるリンク数を制御するために使用します。デフォルトでは、バッチ・サイズの値は 100 に設定されます。

この構成変数は、値に応じて、リフレッシュ処理の中断リクエストに対する受容力が高く（または低く）なります。`LkRefreshErrorsAllowed` の計算は、各バッチの処理中ではなく、処理の後に完了します。したがって、`LkRefreshBatchSize` の値が小さいほど、許容エラー数をすぐに超過するため、リフレッシュ・アクティビティが即時に終了する可能性が高くなります。

技術ヒント：Site Studio を使用する場合は、`LkRefreshBatchSize` を使用するほうが、`LkRefreshErrorPercent` と `LkRefreshErrorThreshold` の設定の組合せを使用するよりも、リフレッシュの中断オプションに対する制御が向上します。たとえば、パーセント値を 5 に設定し、しきい値を 20 に設定した場合、最初のエラーの後にリフレッシュ・アクティビティが中断すると予測されます。ところが、実際には、中断する前に、Link Manager によって複数のエラーが処理されます。

その理由は、リフレッシュ・アクティビティの際、Site Studio のリンクとして認識されるすべてのリンク（または Site Studio による処理が必要なリンク）は、グループ化されてバッチとして Site Studio に送信されます。このため、リフレッシュは効率化されますが、Site Studio では中断およびエラー総数が認識されないため、この間に中断リクエストは応答しなくなります。

Site Studio は、現在のバッチで検出されたエラーの数は認識しています。このため、Link Manage では中断を計算できない状況が発生し、エラーの構成値（パーセントおよびしきい値）は、中断が発生する時期についての Link Manager への提案にすぎません。`LkRefreshBatchSize` を使用すると、Site Studio リンクのバッチを含むリフレッシュ・アクティビティ中の中断の受容性をより正確に制御できます。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
LkRefreshBatchSize=200
```

関連項目

– Link Manager の使用の詳細は、『リポジトリ・コンテンツの管理』を参照してください。

LKREFRESHERRORPERCENT

タイプおよび使用方法

- Link Manager の構成変数（4-19 ページ）
- 検索（5-34 ページ）

説明

Link Manager 機能とともに使用します。

この変数は、LkRefreshErrorThreshold と併用して、リフレッシュ・アクティビティを中断するかどうかを計算します。リフレッシュ・アクティビティでのリンク処理がしきい値を超えると、Link Manager はエラーの割合を計算します。エラー数がパーセント値を超過すると、リフレッシュ・アクティビティは中断されます。

パーセント値は、デフォルトで 10 に設定されます。

注意： LkRefreshErrorsAllowed の値を設定した場合は、しきい値とパーセント値の組合せが上書きされる可能性があります。たとえば、この値が、リンク / エラーに対して計算されるしきい値 / パーセント値よりも小さい場合、処理されたリンク数がしきい値を超過していない場合でも、リフレッシュ・アクティビティが中断される可能性があります。したがって、LkRefreshErrorsAllowed 構成設定か、LkRefreshErrorThreshold と LkRefreshErrorPercent の構成設定の組合せのいずれかを使用することをお薦めします。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
LkRefreshErrorPercent=20
```

関連項目

- Link Manager の使用の詳細は、『リポジトリ・コンテンツの管理』を参照してください。
- LkRefreshErrorsAllowed（6-489 ページ）
- LkRefreshErrorThreshold（6-490 ページ）

LKREFRESHERRORSALLOWED

タイプおよび使用方法

- Link Manager の構成変数（4-19 ページ）
- 検索（5-34 ページ）

説明

Link Manager 機能とともに使用します。

この変数は、リフレッシュ処理の絶対エラー・カウントを設定します。設定されたエラー数が検出されると、リフレッシュ・アクティビティは中断されます。この構成設定は、デフォルトでは使用されません。

注意： `LkRefreshErrorsAllowed` の値を設定した場合は、しきい値とパーセント値の組合せが上書きされる可能性があります。たとえば、この値が、リンク / エラーに対して計算されるしきい値 / パーセント値よりも小さい場合、処理されたリンク数がしきい値を超過していない場合でも、リフレッシュ・アクティビティが中断される可能性があります。したがって、LkRefreshErrorsAllowed 構成設定か、`LkRefreshErrorThreshold` と `LkRefreshErrorPercent` の構成設定の組合せのいずれかを使用することをお薦めします。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
LkRefreshErrorsAllowed=10
```

関連項目

- Link Manager の使用の詳細は、『リポジトリ・コンテンツの管理』を参照してください。
- LkRefreshErrorPercent（6-488 ページ）
- LkRefreshErrorThreshold（6-490 ページ）

LKREFRESHERRORTHRESHOLD

タイプおよび使用方法

- Link Manager の構成変数（4-19 ページ）
- 検索（5-34 ページ）

説明

Link Manager 機能とともに使用します。

この変数は、LkRefreshErrorPercent と併用して、リフレッシュ・アクティビティを中断するかどうかを計算します。リフレッシュ・アクティビティでのリンク処理がしきい値を超えると、Link Manager はエラーの割合を計算します。エラー数がパーセント値を超過すると、リフレッシュ・アクティビティは中断されます。

しきい値は、デフォルトで 100 に設定されます。

注意： LkRefreshErrorsAllowed の値を設定した場合は、しきい値とパーセント値の組合せが上書きされる可能性があります。たとえば、この値が、リンク / エラーに対して計算されるしきい値 / パーセント値よりも小さい場合、処理されたリンク数がしきい値を超過していない場合でも、リフレッシュ・アクティビティが中断される可能性があります。したがって、LkRefreshErrorsAllowed 構成設定か、LkRefreshErrorThreshold と LkRefreshErrorPercent の構成設定の組合せのいずれかを使用することをお薦めします。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
LkRefreshErrorThreshold=200
```

関連項目

- Link Manager の使用の詳細は、『リポジトリ・コンテンツの管理』を参照してください。
- LkRefreshErrorsAllowed（6-489 ページ）
- LkRefreshErrorPercent（6-488 ページ）

LMDEFAULTLAYOUT

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

使用するデフォルトのレイアウトを定義します（たとえば、Top Menus または Trays）。

デフォルトのインストール値は、`Trays` です。

場所

<*install_dir*>/config/config.cfg

例

```
LMDefaultLayout=Top Menus
```

LMDEFAULTSKIN

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

使用するデフォルトのスキンを定義します（たとえば、Collegiate、Stellent、Windows）。

デフォルトのインストール値は、`Stellent` です。

場所

<*install_dir*>/config/config.cfg

例

```
LMDefaultSkin=Collegiate
```

LMGETLAYOUT

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

ユーザーによって選択されたレイアウトを取得します（たとえば、Top Menus または Trays）。

パラメータ

なし。

例

```
<$lmGetLayout()$>
```

関連項目

– lmGetSkin（6-494 ページ）

LMGETSKIN

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

ユーザーによって選択されたスキンを取得します。

パラメータ

なし。

例

```
<$lmGetSkin()$>
```

関連項目

– lmGetLayout（6-493 ページ）

LOADCOLLECTIONINFO

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 検索（5-34 ページ）

説明

検索コレクションのメタデータをロードします。

- 検索コレクションに関するメタデータをロードするために検索サービスによって使用されます。
- **この関数は、ユーザー構成の対象ではありません。**

パラメータ

1 つのパラメータを取ります。

- ロードする検索コレクションの名前

出力

なし。

例

検索コレクション情報をロードします。

```
<$loadCollectionInfo(collection_name)$>
```

LOADDOCMETADEFINITION

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

DocMetaDefinition データベース表を ResultSet としてアクティブなデータ内にロードします。

- *DocMetaDefinition* データベース表は、ロード後にループ処理できます。

パラメータ

なし。

出力

なし。

例

DocMetaDefinition 表を ResultSet としてアクティブなデータ内にロードします。

```
<$loadDocMetaDefinition()$>
```

LOADDOCUMENTPROFILE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- コンテンツ・プロファイル（5-7 ページ）

説明

ページ表示のトリガー値によって指定されたコンテンツ・プロファイルをロードします。この関数は、「Search」、「Check In」、「Info」および「Update」の各ページを表示する際にコールされます。この関数はリクエストでコールされ、コールされたページによってアクション（検索または情報など）が決定されます。

この情報は、コンテキストに依存します。ローカル・データに、*isCheckin*、*isUpdate*、*isQuery* または *isInfo* 変数が設定され、ロードおよび表示されるページに `loadDocumentProfile` 関数を指示します。

パラメータ

なし。

出力

なし。

例

```
<$loadDocumentProfile()$>
```

関連項目

– utLoadDocumentProfiles（6-895 ページ）

LCCAPTION

タイプおよび使用方法

- グローバル関数
- ローカライゼーション

説明

文字列をキャプションにラップする関数です。通常、文字列の右側にコロンが追加されます。ヘブライ語やアラビア語など、右側から左側へ読み取る言語の場合は、文字列の左側にコロンが追加されます。

パラメータ

1 つの必須パラメータを取ります。

- ラップする文字列。
- オプションのパラメータは、ローカライズ文字列の中カッコ内の引数に基づいて解決される式です（たとえば、{1}）。

注意：ローカライゼーション文字列およびロケールの詳細は、『Using Content Server in International Environments』を参照してください。

LOADENTERPRISESEARCHCOLLECTIONS

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

この関数は、ResultSet `EnterpriseSearchCollections` をロードします。これにより、Enterprise Search を使用した問合せが可能なリモート Content Server のリストが表示されます。このデータは、ユーザーが問合せを行うサーバーを選択できるように、検索ページで使用されます。

パラメータ

1 つのパラメータを取ります。

- 0 に設定すると、自動的に検索されるサーバーの名前が ResultSet に含まれません。

LOADENTERPRISESEARCHRESULTS

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

この関数は、Enterprise Search をサポートするために、ページのいくつかの ResultSet の名前を変更します。ResultSet `EnterpriseSearchResults` に対するループ処理を行うときに使用されます。特定のサーバーから ResultSet をロードし、それを `SearchResults` に名前変更します。これにより、検索結果ページでの IdocScript の使用が簡易化されます。

パラメータ

なし。

LOADSCHEMADATA

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- スキーマ（5-33 ページ）

説明

スキーマの結果セットのデータを、現在のデータ・バインダのローカル・データにロードします。

パラメータ

この関数は、パラメータを 0 個、1 個または 2 個取ることができます。

- 0 個の引数を渡された場合は、アクティブな結果セットの行のデータをデータ・バインダに挿入します。値を返しません。
- 1 個の引数を渡された場合、その引数は、現在の行から値を取得する結果セットの名前です。値を返しません。
- 2 個の引数を渡された場合、最初の引数は結果セットの名前で、2 番目の引数はロードするデータ・オブジェクトを識別するキーです。データが存在しない場合は 0 を、存在する場合は 1 を返します。戻り値に基づいて条件付きスクリプトを実行するには、True() または False() を使用します。

例

```
<$loadSchemaData()$>
```

関連項目

– clearSchemaData（6-73 ページ）

LOADSEARCHOPERATORTABLES

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

演算子名の完全なセットと検索構文のマッピングをロードします。

- 検索ページで使用されます。

パラメータ

なし。

出力

演算子名に関連付けられた式を返します（検索演算子マップ）。

例

```
<$loadSearchOperatorTables()$>
```

LOADUSERMETADEFINITION

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

この関数は、カスタム・ユーザー・メタデータ定義をロードします。これは、ResultSet としてページに置かれる `UserMetaDefinition` 表に存在します。LDAP 管理者ページで使用され、管理者がユーザー・メタデータを Content Server にマップする際に役立ちます。

パラメータ

なし。

LOCALEARCHIVERDATEFORMAT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ローカライゼーション（5-29 ページ）
- アーカイバ（5-3 ページ）

説明

アーカイバのバッチ・ファイルのサブディレクトリでのネーミング・フォーマットを指定します。多くの場合、異なるバージョンの Content Server を使用してエクスポートおよびインポートする場合に使用されます。

- デフォルトは、*yy-MMM-dd_HH.mm.ss_SSS* です。
  たとえば、*03-feb-04_15.04.14_174*。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
LocaleArchiverDateFormat=dd-mm-yy
```

LOCALGROUPSERVER

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

NTLM セキュリティで、指定された Windows Server 上のローカル・グループが、グローバル（ドメイン）グループであるかのように処理されます。

- UseLocalGroups（6-845 ページ）設定が TRUE の場合、この設定で指定された Windows Server 上のローカル・グループは、デフォルト・マスター・ドメインのグローバル（ドメイン）グループであるかのようにロールおよびアカウントにマップされます。
- デフォルトは、デフォルト・マスター・ドメインのドメイン・コントローラです。

場所

- 「Configure NT Domain Security」ページ→「Local Group Server」
- <*install_dir*>/data/users/config/filter.hda

例

次は構成エントリの使用例です。

```
LocalGroupServer=server1
```

関連項目

– UseLocalGroups（6-845 ページ）

LOCALPAGETYPE

タイプおよび使用方法

- ページ表示変数（4-25 ページ）
- ページ表示（5-30 ページ）

説明

この関数は、ライブラリ・フォルダのページのページ・タイプを返します。これはディレクトリまたは問合せページです。通常、この関数は、Layout Manager API によって使用されます。`Trays` をレイアウトとして選択すると、ライブラリ・ページのツリー表示の構築に使用されます。

パラメータ

1 つのパラメータを取ります。

- ツリー表示の構築に使用されるリンク・データ

LOGDIR

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ディレクトリおよびパス（5-22 ページ）
- デバッグ（5-20 ページ）

説明

logs ディレクトリのパスを定義します。

- デフォルトは、*<install_dir>/weblayout/groups/secure/logs/* です。

場所

<install_dir>/config/config.cfg

例

次は構成設定の例です。

```
LogDir=c:/stellent_logs/
```

LOGMESSAGEPREFIX

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）
- デバッグ（5-20 ページ）

説明

ログ・エントリに接頭辞を追加して、エントリを作成するコンテンツ・サーバーを指定します。

- 通常、マスター / プロキシ環境で使用します。
- デフォルト値はありません。

場所

<*install_dir*>/bin/intradoc.cfg

例

次は構成エントリの使用例です。

```
LogMessagePrefix=Master
```

LOGNAME

タイプおよび使用方法

- 構成変数（4-3 ページ）
- デバッグ（5-20 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

Content Server Analyzer ユーティリティによって作成されるログ・ファイルのファイル名を指定します。

- このログ・ファイルは、IdcAnalyzeLogDir（6-330 ページ）設定によって指定されたディレクトリに作成されます。
- デフォルトは、*IdcAnalyze.log* です。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
LogName=contentserver1_analyze.log
```

関連項目

- IdcAnalyzeLogDir（6-330 ページ）
- SafeDir（6-695 ページ）
- TempDir（6-795 ページ）

LOGWITHCONNECTIONNAME

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Inbound Refinery（5-11 ページ）
- デバッグ（5-20 ページ）

説明

Inbound Refinery ログに接続名を追加します。

- TRUE に設定すると、各ログ・エントリは、「The connection <ConnectionName> logs:」という文字列で始まります。これは、複数のリファイナリが 1 つのコンテンツ・サーバーを指している場合に役立ちます。
- デフォルトは、FALSE です。

場所

<*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
LogWithConnectionName=true
```

MACSUPPORTSSIGNEDAPPLETS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）
- クライアント（5-4 ページ）

説明

Macintosh クライアントのアップロードおよびダウンロード・アプレットを有効にします。

- Macintosh クライアントを使用する場合にのみ使用されます。
- TRUE に設定すると、Macintosh クライアントは、アップロードおよびダウンロード・アプレットを使用して、複数のファイルの入出力をチェックできます。
- FALSE に設定すると、Macintosh クライアントは、アップロードおよびダウンロード・アプレットを使用できません。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
MacSupportsSignedApplets=true
```

MAILSERVER

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

コンテンツ・サーバーの電子メールの処理に使用される電子メール・サーバーを定義します。

- 電子メールのサポートは、Oracle Collaboration Server のワークフロー通知、サブスクリプション通知およびプロジェクト通知で必要です。
- デフォルト値はありません。

場所

- 「System Properties」→「Internet」タブ→「Mail Server」
- 「Admin Server」→「Internet Configuration」→「Mail Server」
- <*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
MailServer=mail.company.com
```

関連項目

– SysAdminAddress（6-787 ページ）
– SmtpPort（6-750 ページ）

MAJORREVSEQ

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

リビジョン番号のメジャー・シーケンスを定義します。

注意：リビジョン・シーケンスの詳細は、『Content Server System Administration Guide』のリビジョン・ラベル・シーケンスに関する項を参照してください。

- メジャー・リビジョン・ラベル・シーケンスを返します（構成設定の値を返します）。
- 文字列を返します。
- デフォルトは、*1-99* です。

場所

- 「System Properties」→「Options」タブ→「Major Revision Label Sequence」
- 「Admin Server」→「General Configuration」→「Major Revision Label Sequence」
- <*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
MajorRevSeq=A-Z
```

スクリプトとして、構成設定の値を返します。

```
<$MajorRevSeq$>
```

関連項目

– MinorRevSeq（6-535 ページ）

MAXARCHIVEERRORSALLOWED

タイプおよび使用方法

- 構成変数（4-3 ページ）
- アーカイバ（5-3 ページ）

説明

アーカイバがインポートまたはエクスポートを停止するまでのエラー数を設定します。

- デフォルトは、50 です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
MaxArchiveErrorsAllowed=10
```

MAXCOLLECTIONSIZE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）

説明

インデクサに渡される 1 バッチ当たりのファイル数を定義します。

- 有効な範囲は、1 ～ 65535 です。
- 大きい索引コレクションの場合、推奨される値は 2000 です。値を小さくすると、索引付けのパフォーマンスが低下します。
- 1 パッチ当たりのファイル数を返します。
- デフォルトは、25 です。

場所

- 「Repository Manager」→「Indexer」タブ→「Configure」→「Content Items Per Indexer Batch」
- <*install_dir*>/config/config.cfg

例

構成設定として、バッチ・サイズを定義します。

```
MaxCollectionSize=25
```

スクリプトとして、構成ファイルから値を返します。

```
<$MaxCollectionSize$>
```

MAXCONVERSIONCACHESIZEINMEGS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Dynamic Converter（5-17 ページ）

説明

Dynamic Converter キャッシュの最大サイズ（MB）を定義します。

- クリーンアップ・スキャンを実行すると（6-122 ページの「DatedCacheIntervalDays」を参照）、次のように、この設定に基づいてキャッシュをクリーンアップします。
  - このサイズと比較して、Dynamic Converter キャッシュがより大きい場合は、このサイズより小さくなるまで、キャッシュされた最も古いファイル（最終アクセス日によって決定）が削除されます。
  - このサイズと比較して、Dynamic Converter キャッシュがより小さい場合は、ConversionCacheExpirationPeriodInDays（6-105 ページ）より古く、変換ルール（強制変換または事前変換）を使用して変換されなかったファイルのみが削除されます。
- デフォルトは、10,000（10GB）です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
MaxConversionCacheSizeInMegs=5,000
```

関連項目

- ConversionCacheExpirationPeriodInDays（6-105 ページ）
- DatedCacheIntervalDays（6-122 ページ）

MAXDOCINDEXERRORS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）

説明

インデクサが索引付けまたは再構築を停止するまでのエラー数を設定します。

- デフォルトは、50 です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
MaxDocIndexErrors=50
```

MAXERRORSALLOWED

タイプおよび使用方法

- 構成変数（4-3 ページ）
- バッチ・ローダー（5-3 ページ）

説明

バッチ・ローダーがバッチ・ロード・ファイルからのレコード処理を停止するまでのエラー数を設定します。

- 夜間に多数のコンテンツ・アイテムを使用してバッチ・ローダーを実行する場合は、デフォルト値を増加します。バッチ・ローダーで少量のコンテンツ・アイテムが計測された場合は、デフォルト値を減らします。
- デフォルトは、50 です。
- 値を整数として返します。

場所

<*install_dir*>/bin/intradoc.cfg

例

次は構成エントリの使用例です。

```
MaxErrorsAllowed=50
```

MAXHITCOUNT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）

説明

検索中に見つけることができる最大結果数を設定します。

- デフォルトは、100000 です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
MaxDocIndexErrors=50000
```

MAXINTEGERALLOWED

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

カスタム整数メタデータ・フィールドで許可される最大値を設定します。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
MaxIntegerAllowed=1000000
```

関連項目

– MinIntegerAllowed（6-533 ページ）

MAXLENGTH

タイプおよび使用方法

- 共通フィールド表示変数（4-27 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドで許可される最大文字数を設定します。

出力

フィールドの最大長を返します。

例

フィールドの最大長を 100 文字に設定します。

```
<$maxLength=100$>
```

現在のフィールドの長さに基づいてカスタム・フィールドの長さを指定します。

```
<$customFieldLength = getValue("#active", fieldName & ":maxLength")$>
```

MAXMERGEBASECOUNT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）

説明

Verity Integration で使用されます。Verity Integration は、Content Server とともに使用できるアドオン製品であり、購入することができます。

Verity Indexer の自動チューンアップ機能の実行頻度を設定します。

- この値は、Verity Indexer 実行コマンドにチューンアップ・パラメータを含めるかどうかを決定します。この値は、索引付けされたファイルの数量とファイル・サイズを考慮したしきい値です。この設定を増加すると、自動クリーンアップの頻度が低下することがあります。
- デフォルトは、200 です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
AutoMaxMergeTrace=true
```

関連項目

- DoAutoMaxMerge（6-170 ページ）
- AdditionalIndexBuildParams（6-11 ページ）
- ScriptDebugTrace（6-709 ページ）

MAXNUMBERCONVERTEDHISTORYSHOW

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Inbound Refinery（5-11 ページ）

説明

Inbound Refinery 画面に表示される変換の数を定義します。

- デフォルトは、50 です。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
MaxNumberConvertedHistoryShow=25
```

MAXNUMRECURSIVESTEPDEFINITIONS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Inbound Refinery（5-11 ページ）

説明

カスタム変換を定義するときに使用できる再帰型の最大レベル数を指定します。

- カスタム変換エンジンは別の変換をコールできますが、ループを回避するために、この設定によって、このようなコールのレベル数を制限します。
- デフォルトは、2 です。

場所

<*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
MaxNumRecursiveStepDefinitions=3
```

MAXQUERYROWS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

Repository Manager、「Active Report」ページおよび「Work In Progress」ページに表示される最大行数を設定します。

- この値を増加すると、レスポンス時間が遅くなります。
- デフォルトは、200 です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
MaxQueryRows=50
```

関連項目

– IsMaxRows（6-419 ページ）

MAXREQUESTTHREADCOUNT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

最大同時リクエスト・スレッド数を設定します。

- デフォルトは、100 です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
MaxRequestThreadCount=50
```

MAXRESULTS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）

説明

検索で返される最大コンテンツ・アイテム数を設定します。

- 値を指定すると、プログラムされたデフォルトを上書きします。
- デフォルト値はありません。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
MaxResults=200
```

MAXSAVEDSEARCHRESULTS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）

説明

ユーザーが作成できる保存済の検索の最大数を設定します。値を指定すると、デフォルト（10）を上書きします。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
MaxSavedSearchResults=25
```

MAXSEARCHCONNECTIONS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）

説明

一度にオープンできる検索の最大接続数を設定します。

- たとえば、MaxSearchConnections=5 のとき、6 つの検索リクエストを同時に行うと、6 番目のリクエストは、最初の 5 つのリクエストのいずれかが完了するまでキューに保持されます。
- デフォルトは、10 です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
MaxSearchConnections=5
```

MAXSIZERESOURCEFILECACHEMILLIONS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

リソース・ファイルの最大サイズを、数百万のダブルバイト文字の単位で設定します。

- この値は、ファイルの読取りおよび解析を行う前に、ファイルのサイズを制限します。
- メモリー内の実際のサイズのより正確な見積りを取得するには、ファイル・サイズに 2.5 を掛けます。
- デフォルトは、10 です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
MaxSizeResourceFileCacheMillions=20
```

MAXSTANDARDDATABASERESULTS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

表示される最大ユーザー数を設定します。

- この値は、リストからユーザーを選択できる任意のアプレットで有効です。
- デフォルトは、500 です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
MaxStandardDatabaseResults=1000
```

MEMOFIELDSIZE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- フィールド表示（5-24 ページ）
- データベース（5-18 ページ）

説明

データベースに作成されるメモ・フィールドのフィールド・サイズを定義します。

- データベースでサポートされるサイズを選択する必要があります。
- デフォルトは、255 文字です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
MemoFieldSize=300
```

関連項目

– MinMemoFieldSize（6-534 ページ）

MININTEGERALLOWED

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

カスタム整数メタデータ・フィールドで許可される最小値を設定します。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
MinIntegerAllowed=0
```

関連項目

– MaxIntegerAllowed（6-520 ページ）

MINMEMOFIELDSIZE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- フィールド表示（5-24 ページ）
- データベース（5-18 ページ）

説明

LONG テキスト・フィールドではなく、メモ・フィールドとして定義されるカスタム・メタデータ・フィールドのサイズを定義します。

- データベースで *DocMeta* 表のフィールド・サイズが直接に編集されている場合、コンテンツ・サーバーは、*varchar* フィールド・タイプをフィールドの長さで分類します。
- デフォルトは、255 文字です。
- この設定が定義されている場合、次のフィールド・タイプ・ルールが適用されます。

フィールド・タイプ	評価ルール
Text	50 より小さい
Long Text	50 以上かつ MinMemoFieldSize より小さい
Memo	MinMemoFieldSize 以上

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
MinMemoFieldSize=300
```

関連項目

– MemoFieldSize（6-532 ページ）

MINORREVSEQ

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

リビジョン番号のマイナー・シーケンスを定義します。

注意：リビジョン・シーケンスの詳細は、『Content Server System Administration Guide』のリビジョン・ラベル・シーケンスに関する項を参照してください。

- マイナー・リビジョン・ラベル・シーケンスを返します（構成設定の値を返します）。
- 文字列を返します。

場所

- 「System Properties」→「Options」タブ→「Minor Revision Label Sequence」
- 「Admin Server」→「General Configuration」→「Minor Revision Label Sequence」
- <*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
MinorRevSeq=a-c
```

スクリプトとして、構成設定の値を返します。

```
<$MinorRevSeq$>
```

関連項目

– MajorRevSeq（6-513 ページ）

MRSIDLOWESTCOMPRESSIONLEVEL

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Compression（5-18 ページ）

説明

Compression アドオン・モジュールの再試行機能に対する圧縮の最低レベルを設定します。

- この設定を有効にするには、AllowMrSidRetryCompressionLevel（6-31 ページ）が TRUE である必要があります。
- 特定のファイルについて Compression の「Zoom Level」が高すぎる場合は、圧縮が成功するまで、次に最も低いレベルで圧縮を再試行します。この設定で指定されたレベルでファイルを圧縮できない場合は、ネイティブ・ファイルを JPG ファイルに変換します。
- デフォルトは、3 です。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
MrSidLowestCompressionLevel=4
```

関連項目

– AllowMrSidRetryCompressionLevel（6-31 ページ）

MSIE

タイプおよび使用方法

- 値変数（4-34 ページ）
- クライアント（5-4 ページ）

説明

クライアント・ブラウザが Microsoft Internet Explorer かどうかをチェックします。

出力

- クライアント・ブラウザが Internet Explorer の場合は、TRUE を返します。
- クライアント・ブラウザが Internet Explorer でない場合は、FALSE を返します。

例

ブラウザが Internet Explorer 以外の場合、別の URL にリダイレクトします。

```
<$if not MSIE$>
   <a href="<$redirect$>">
<$endif$>
```

MSPUBEXEPATH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

MS Publisher 実行可能ファイルのパスを定義します。

- 文字列としてパスを返します。
- デフォルトは、空の文字列です。

場所

- 「Shared Inbound Refinery Configuration」画面→「Paths」タブ→「MS Publisher Executable Path」
- <*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
MSPubexePath=c:/Program Files/Microsoft Publisher/mspub.exe
```

MULTIUPLOAD

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）
- クライアント（5-4 ページ）

説明

複数ファイルの Upload アプレットを有効にします。

- TRUE に設定すると、Upload アプレットが有効になり、複数のファイルを ZIP して単一のコンテンツ・アイテムとしてチェックインできます。
- FALSE に設定した場合、Upload アプレットは無効です。
- デフォルトは、FALSE です。

場所

- 「System Properties」→「Options」タブ→「Enable upload applet」
- 「Admin Server」→「General Configuration」→「Enable upload applet」
- <*install_dir*>/config/config.cfg

例

構成設定として、Upload アプレットを有効にします。

```
MultiUpload=true
```

スクリプトとして、Upload アプレットの状態を評価します。

```
<$MultiUpload$>
```

関連項目

– DownloadApplet（6-181 ページ）
– UploadApplet（6-826 ページ）

NETWORKADMINGROUP

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

特定の Windows ネットワーク・グループを admin ロールにマップします。

- NTLM セキュリティで使用します。
- デフォルトは、*Domain Admins* です。

場所

- 「Configure NT Domain Security」ページ→「Network Admin Group」
- <*install_dir*>/data/users/config/filter.hda

例

次は構成エントリの使用例です。

```
NetworkAdminGroup=admins
```

NEVERTIMEOUTFORCEDCONVERSIONSTEMPLATECHANGE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Dynamic Converter（5-17 ページ）

説明

テンプレートまたはレイアウトが変更された場合、Dynamic Converter でファイルを再変換するかどうかを定義します。

- TRUE に設定すると、強制変換または事前変換によって変換されたファイルは、テンプレートまたはレイアウトが変更されたときに再変換されません。
- 別のテンプレートまたはレイアウトを使用するように変換ルールが変更された場合、通常の評価ロジックが使用されます。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
NeverTimeoutForcedConversionsTemplateChange=true
```

関連項目

– TimeoutChangedTemplateForcedConversionsInMins（6-810 ページ）
– TimeoutMismatchedForcedConversionsInMins（6-811 ページ）

NOAUTOMATION

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

自動化されたコンテンツ・サーバーのすべてのアクティビティを無効にします。

- この設定は、IsAutoArchiver（6-375 ページ）、IsAutoQueue（6-377 ページ）および IsAutoSearch（6-378 ページ）の設定を上書きします。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
NoAutomation=true
```

関連項目

- IsAutoArchiver（6-375 ページ）
- IsAutoQueue（6-377 ページ）
- IsAutoSearch（6-378 ページ）

NOINFOCOLUMN

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）
- ページ表示（5-30 ページ）

説明

検索結果ページに、「Info」列を表示するかどうかを定義します。

- TRUE に設定すると、標準の検索結果ページに、「Content Information」ページへのリンクを含む列が表示されます。
- FALSE に設定すると、標準の検索結果ページに、「Info」リンクが表示されません。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
noInfoColumn=1
```

関連項目

- showDocActions（6-742 ページ）
- showSearchResultsCheckoutColumn（6-746 ページ）
- noRevisionsColumn（6-546 ページ）

NOMATCHES

タイプおよび使用方法

- 値変数（4-34 ページ）
- 検索（5-34 ページ）

説明

検索問合せから一致が見つかったかどうかをチェックします。

- 通常、検索結果ページにメッセージを表示するために使用されます。

出力

- 一致が見つからない場合は、TRUE を返します。
- 一致が見つかった場合は、FALSE を返します。

例

問合せで一致が見つからなかった場合はテキストを表示します。

```
<$if NoMatches$>
    <p><font face="arial" size="2">
    Found no matches out of <$TotalDocsProcessed$> documents searched matching
    the query.</p>
<$endif$>
```

NOMCPREFILL

タイプおよび使用方法

- ページ表示変数（4-25 ページ）
- ページ表示（5-30 ページ）

説明

MultiCheckin コンポーネントでメタデータ・フィールドに事前に値を挿入しないことを指定します。

- この変数は、カスタム HCST および Folders コンポーネント・ページなどの特殊なページで使用できます。
- TRUE に設定した場合、MultiCheckin コンポーネントでは、メタデータ・フィールドに事前に値を挿入しません。
- FALSE に設定した場合、MultiCheckin コンポーネントでは、メタデータ・フィールドに事前に値が挿入されます。

出力

- MultiCheckin コンポーネントでメタデータ・フィールドに事前に値が挿入されない場合は、TRUE を返します。
- MultiCheckin コンポーネントでメタデータ・フィールドに事前に値が挿入される場合は、FALSE を返します。

例

ページの先頭付近で次のように設定します。

```
noMCPrefill=1
```

スクリプトとしての使用例を次に示します。

```
<$noMCPrefill$>
```

noRevisionsColumn

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）
- ページ表示（5-30 ページ）

説明

検索結果ページに、「Revisions」列を表示するかどうかを定義します。

- TRUE に設定すると、標準の検索結果ページに、「Revision History」ページへのリンクを含む列が表示されます。
- FALSE に設定すると、標準の検索結果ページに、「Revision History」リンクは表示されません。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
noRevisionsColumn=1
```

関連項目

- showDocActions（6-742 ページ）
- showSearchResultsCheckoutColumn（6-746 ページ）
- noInfoColumn（6-543 ページ）

NoSearchConnectionCloseOnDocChange

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）

説明

ドキュメントが変更された場合（挿入、更新または削除）、すべての検索接続の自動終了を無効にします。

警告：この変数を TRUE に設定すると、システムの負荷が高いとき、正しい検索結果が得られない場合があります。

- TRUE に設定すると、ドキュメントが変更された場合でも、検索接続はオープンしたままになります。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
NoSearchConnectionCloseOnDocChange=true
```

NOTATIONFORREQUIREDFIELDS

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

必須メタデータ・フィールドの表示方法を設定します。

- 設定すると、必須フィールドは、メタデータ・フィールドの前に指定した文字列が付加されて表示されます。
- この文字列は、Cascading Style Sheet に含まれます。
- `StyleForRequiredFields` が定義されている場合、`NotationforRequiredFields` は、`StyleForRequiredFields` で定義されたスタイルで書式設定されます。

パラメータ

1 つのパラメータを取ります。

- パラメータは、文字列のみです。

場所

<*install_dir*>/config/config.cfg

例

必須フィールドの前にアスタリスクを設定する例を次に示します。

```
NotationForRequiredFields=*
```

関連項目

– StyleForRequiredFields（6-783 ページ）

NOTIFICATIONINTERVALINDAYS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

この構成変数は、通知の問合せを実行する頻度を定義します。通知の問合せは、期限切れコンテンツの検索に使用する基準を定義します。

NotificationIntervalInDays を設定しない場合、デフォルト値は 1 日です。

場所

<*install_dir*>/config/config.cfg

例

```
NotificationIntervalInDays=3
```

関連項目

- EnableExpirationNotifier（6-196 ページ）
- NotificationMaximum（6-550 ページ）
- NotificationQuery（6-551 ページ）
- NotifyExtras（6-553 ページ）
- NotifyTime（6-554 ページ）

NOTIFICATIONMAXIMUM

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

この構成変数は、通知の問合せから返される最大コンテンツ・アイテム数を定義します。通知の問合せは、期限切れコンテンツの検索に使用する基準を定義します。

NotificationMaximum を設定しない場合、デフォルト値は 1000 です。

場所

<*install_dir*>/config/config.cfg

例

```
NotificationMaximum=500
```

関連項目

- EnableExpirationNotifier（6-196 ページ）
- NotificationIntervalInDays（6-549 ページ）
- NotificationQuery（6-551 ページ）
- NotifyExtras（6-553 ページ）
- NotifyTime（6-554 ページ）

NOTIFICATIONQUERY

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

この設定は、期限切れコンテンツを検索する自動問合せの基準を定義します。

NotificationQuery を設定しない場合、デフォルト値は、7 日以内に期限切れになるすべてのコンテンツです。

パラメータ

1 つのパラメータ（使用する問合せ）を取ります。問合せは、次のいずれかです。問合せのサンプルは、「例」の項を参照してください。

- Idoc スクリプトから作成された Idoc スクリプト問合せ。
- URL エンコードされた問合せ。検索を実行するとき、Web ブラウザ・アドレスに表示される URL を使用します。
- 検索の変数を定義するプレーン・テキストの問合せ。

注意 : 検索エンジンとして Verity または FAST を使用する場合、すでに期限切れになったコンテンツの電子メール通知は送信できません。

場所

<*install_dir*>/config/config.cfg

例

Idoc スクリプトの例

データベースの索引付けとともに使用する場合、次の問合せは、7 日以内に期限が切れるすべてのドキュメントの電子メール通知を提供します。

```
NotificationQuery=dOutDate < '<$dateCurrent(7)$>'>
```

URL エンコードの例

次の問合せは、2007 年 8 月 1 日以降に期限切れになるすべてのコンテンツを返します。QueryText 部分で始まり、ブラウザのアドレス行から URL をコピーします。

```
NotificationQuery=QueryText=dOutDate+%3C+%608%2F1%2F06%60&SearchProviders= [...}
```

プレーン・テキストの問合せ

次の問合せは、2007 年 8 月 1 日以降に期限切れになるすべてのコンテンツを返します。

```
NotificationQuery=8/1/07
```

関連項目

- EnableExpirationNotifier（6-196 ページ）
- NotificationIntervalInDays（6-549 ページ）
- NotificationMaximum（6-550 ページ）
- NotifyExtras（6-553 ページ）
- NotifyTime（6-554 ページ）

NOTIFYEXTRAS

タイプおよび使用方法

❖ 構成変数（4-3 ページ）

❖ コンテンツ・アイテム（5-6 ページ）

説明

この構成変数は、各コンテンツ・アイテムの作成者に加えて、期限切れコンテンツのリストを受信するユーザーを定義します。通知の問合せは、期限切れコンテンツの検索に使用する基準を定義します。

`NotifyExtras` を設定しない場合、デフォルト値は sysadmin です。config.cfg に `NotifyExtras` が設定され、値が定義されていない場合、追加の通知は送信されません。

場所

<*install_dir*>/config/config.cfg

例

```
NotificationExtras=jwilson, nadams
```

関連項目

– EnableExpirationNotifier（6-196 ページ）
– NotificationIntervalInDays（6-549 ページ）
– NotificationMaximum（6-550 ページ）
– NotificationQuery（6-551 ページ）
– NotifyTime（6-554 ページ）

NOTIFYTIME

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

この構成変数は、通知の問合せを実行する時間を定義します（24 時間表記で指定）。通知の問合せは、期限切れコンテンツの検索に使用する基準を定義します。

NotifyTime を設定しない場合、デフォルト値は深夜（00:01）です。

場所

<*install_dir*>/config/config.cfg

例

次の例では、時間を 11:30AM に設定します。

```
NotifyTime=11:30
```

次の例では、時間を 1:30PM に設定します。

```
NotifyTime=13:30
```

関連項目

- EnableExpirationNotifier（6-196 ページ）
- NotificationIntervalInDays（6-549 ページ）
- NotificationMaximum（6-550 ページ）
- NotificationQuery（6-551 ページ）
- NotifyExtras（6-553 ページ）

NTLMSECURITYENABLED

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

コンテンツ・サーバーの Windows ネットワーク認証を有効にします。

- このオプションは、IIS Web サーバーを使用する場合にのみ利用できます。
- TRUE に設定すると、Windows ネットワーク（NTLM）または Active Directory で定義されたユーザーは、ネットワーク資格証明を使用してコンテンツ・サーバーにログインできます。
- デフォルトは、FALSE です。

場所

- 「System Properties」→「Internet」タブ→「Use Microsoft Security」
- 「Admin Server」→「Internet Configuration」→「Use Microsoft Security」
- <*install_dir*>/config/config.cfg
- <*install_dir*>/data/users/config/filter.hda

例

NTLM セキュリティを有効にします。

```
NtlmSecurityEnabled=true
```

関連項目

- UseNtlm（6-848 ページ）
- UseAdsi（6-831 ページ）
- AllowIntranetUsers（6-28 ページ）
- HasExternalUsers（6-282 ページ）
- DefaultAuth（6-131 ページ）
- IntradocRealm（6-369 ページ）

NUMADDITIONALRENDITIONS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・アイテム（5-6 ページ）
- Inbound Refinery（5-11 ページ）

説明

コンテンツ・アイテムのレンディション数を設定するために Thumbnails プログラムによって使用されます。

- 0 に設定すると、検索結果ページにサムネイルは作成されません。
- 1 に設定すると、検索結果ページにサムネイルが作成されます。
- デフォルトは、0 です。

場所

<*install_dir*>/config/config.cfg

例

次の例では、この変数を使用して、`useThumbnails` を設定するかどうかを決定しています。

```
<@dynamichtml searchapi_result_definitions@>
    <$if NumAdditionalRenditions and NumAdditionalRenditions > 0$>
        <$useThumbnails=1$>
    <$endif$>
<@end@>
```

NUMCONNECTIONS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- データベース（5-18 ページ）

説明

データベースへのオープン接続数を設定します。

- デフォルトは、15 です。スタンドアロン・アプレットのデフォルトは、4 です。
- この設定は、コンテンツ・サーバー、スタンドアロン・アプリケーションおよびユーティリティに適用されるため、各アプリケーションで、指定された接続数が使用されます。

技術ヒント：データベースで許容される一度にオープン可能な接続数が十分でない場合、アプリケーションまたはユーティリティを起動するときにエラーが発生します。

この問題を回避するには、次のように、コンテンツ・サーバーの *NumConnections* 設定を、アプリケーションおよびユーティリティの設定から分離します。

1. コンテンツ・サーバーの *NumConnections* 設定は、*<install_dir>/config/config.cfg* ではなく、*<install_dir>/bin/intradoc.cfg* ファイルに指定します。
2. *<install_dir>/bin/* ディレクトリのコピーを作成し、一意の名前を付けます（*<install_dir>/applet_bin/* など）。
3. 新しいディレクトリの *intradoc.cfg* ファイルに、アプリケーションおよびユーティリティの *NumConnections* 値を設定します（たとえば、*NumConnections=2*）。
4. （Windows のスクリプトおよびショートカットを変更して）新しいディレクトリからアプリケーションおよびユーティリティを実行し、コンテンツ・サーバーは引き続き *<install_dir>/bin/* ディレクトリから実行します。

必要に応じて、アプリケーションおよびユーティリティごとに、ディレクトリおよび *NumConnections* 設定を個別に作成します。

場所

<install_dir>/config/config.cfg

例

次は構成エントリの使用例です。

```
NumConnections=10
```

ODBCDATEFORMAT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- アーカイバ（5-3 ページ）

説明

Content Server 7.x のアーカイブのインポートに下位互換性を提供するため、Content Server バージョン 6.x で使用されます。

- Content Server 7.x には、以前のバージョンに存在しないミリ秒のデータ・フォーマットがあります。6.x のアーカイバで、7.x のアーカイブをインポートしようとすると、ミリ秒フィールドが原因でエラーになります。

例

次の手順を実行して、このエントリを 6.x *config.cfg* ファイルに追加します。

1. Content Server 6.x *config.cfg* ファイルをコピーして別の名前を付けます（バックアップとして保存するため）。
2. テキスト・エディタで *config.cfg* ファイルを編集して、次の設定を追加します。
   ```
   OdbcDateFormat='{ts' "yyyy-MM-dd HH:mm:ss{.SSS}"'}'
   ```
3. サーバーを再起動します。

ONEMATCH

タイプおよび使用方法

- 値変数（4-34 ページ）
- 検索（5-34 ページ）

説明

検索問合せから一致が 1 つのみ見つかったかどうかをチェックします。

- 通常、検索結果ページにメッセージを表示するために使用されます。

出力

- 1 つの一致のみが見つかった場合は、TRUE を返します。
- 一致するものがないか、複数の一致が見つかった場合は、FALSE を返します。

例

問合せで一致が 1 つのみ見つかった場合はテキストを表示します。

```
<$if OneMatch$>
    <p><font face="arial" size="2">
    Found <$TotalRows$> document matching the query.</p>
<$endif$>
```

OPENOFFICELISTENERPORT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

OpenOffice リスナーのソケット・ポートを定義します。

- この設定は、PDF Converter の OpenOffice 接続文字列で使用されます。
  `uno:socket,host=<OpenOfficeListenerURL>,`
  `port=<OpenOfficeListenerPort>;urp;StarOffice.ServiceManager`
- デフォルトは、8100 です。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
OpenOfficeListenerPort=8110
```

関連項目

– OpenOfficeListenerURL（6-561 ページ）

OPENOFFICELISTENERURL

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

OpenOffice リスナーの URL（ソケット・ホスト名）を定義します。

- この設定は、PDF Converter の OpenOffice 接続文字列で使用されます。
  `uno:socket,host=<OpenOfficeListenerURL>,`
  `port=<OpenOfficeListenerPort>;urp;StarOffice.ServiceManager`
- デフォルトは、*localhost* です。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
OpenOfficeListenerURL=/mycomputer/localhost
```

関連項目

– OpenOfficeListenerPort（6-560 ページ）

OPENOFFICEPATH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

OpenOffice リスナーのパスを定義します。

- PDF Converter が OpenOffice 接続に接続できない場合は、このパスで定義されたリスナーのオープンを試みます。
- デフォルトは、空の文字列です。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
OpenOfficePath=C:¥Program Files¥Open Office.org.1.0.1¥program¥quickstart.exe
```

OPTIMIZEPDF

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

変換中に PDF ファイルを最適化するかどうかを定義します。

- 最適化により、コンテンツ・アイテムで複数回表示されるグラフィックおよびテキストが削除され、各アイテムの最初の表示へのポインタに置換されます。最適化すると、コンテンツ・アイテム全体がダウンロードされる前に、PDF ファイルの最初のページの表示を開始できます。（この機能は、Adobe Acrobat バージョン 5.0 では、Fast Web View と呼ばれます。）
- TRUE に設定すると、変換中に PDF ファイルが最適化されます。
- デフォルトは、FALSE です。

場所

- 「Local Inbound Refinery Configuration」画面→「Native Options」タブ→「Optimize PDF」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
OptimizePDF=true
```

OPTIONLISTKEY

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

オプション・リストの値が含まれる ResultSet の列名を指定します。

出力

オプション・リストの ResultSet 列の値を返します。

例

チェックイン・ページの標準 *Format* フィールドのオプション・リスト値を取得するための *DocFormats* ResultSet の列として *dFormat* を指定します。

```
<@dynamichtml std_override_format_field@>
<$if not isInfo and IsOverrideFormat$>
    <$fieldIsOptionList = 1, optionListResultSet = "DocFormats",
    optionListKey = "dFormat", optionListValueInclude =
    "std_override_format_option_value", addEmptyOption = 1,
    emptyOptionLabel = lc("wwEmptyFormatOption")$>
    <$include std_display_field$>
<$endif$>
<@end@>7
```

関連項目

– optionListResultSet（6-567 ページ）

OPTIONLISTNAME

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

オプション・リストの名前を指定します。

- 標準メタデータ・フィールドの場合、これは内部オプション・リストの名前です（3-33 ページの「内部オプション・リスト」を参照）。
- カスタム・メタデータ・フィールドの場合、これは接尾辞 .options を持つフィールドの名前です。

出力

オプション・リストの名前を返します。

例

フィールドが単一ユーザーに制限されていない場合、オプション・リストの名前を *docAuthors* に設定します。

```
<$if SingleUser$>
    <$isInfoOnly = 1$>
<$else$>
    <$fieldIsOptionList = 1, optionListName = "docAuthors"$>
    <$if HasExternalUsers$><$fieldOptionListType= "combo"$>
    <$endif$>
<$endif$>
```

デフォルトのオプション・リスト・スクリプトを定義します。

```
<$if optionsAllowPreselect and fieldValue$>
    <$defaultOptionListScript = "<$optList " & optionListName &
    ":fieldValue$>"$>
<$else$>
    <$defaultOptionListScript = "<$optList " & optionListName &
    ":noselected$>"$>
<$endif$>
```

関連項目

- defaultOptionListScript（6-142 ページ）
- fieldValue（6-229 ページ）
- optionListScript（6-568 ページ）
- optList（6-572 ページ）
- fieldIsOptionList（6-224 ページ）
- fieldOptionListType（6-227 ページ）

OPTIONLISTRESULTSET

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

オプション・リストの値が含まれる ResultSet の名前を指定します。

出力

オプション・リストの ResultSet を返します。

例

DocFormats を、検索ページの標準 *Format* フィールドのオプション・リストの ResultSet として指定します。

```
<@dynamichtml std_format_fields@>
<$if ClientControlled or dFormat or dExtension$>
    <$fieldName = "dFormat", fieldCaption = lc("wwNativeFormat"),
    optionListResultSet = "DocFormats"$> <$include std_display_field$>
    <$fieldName = "dExtension", fieldCaption = lc("wwNativeExtension"),
    fieldWidth = 10$>
    <$include std_display_field$>
<$endif$>
<@end@>
```

ResultSet に対するループ処理によって、オプション・リストを作成します。

```
<@dynamichtml compute_resultset_option_list_script@>
    <$if not optionListKey$>
        <$optionListKey = fieldName$>
    <$endif$>
    <$defaultOptionListScript = "<$loop " & optionListResultSet & "$>" &
    "<$inc('std_resultset_option_list_item')$>" & "<$endloop$>"$>
<@end@>
```

関連項目

- optionListKey（6-564 ページ）
- defaultOptionListScript（6-142 ページ）
- optionListValueInclude（6-569 ページ）

OPTIONLISTSCRIPT

タイプおよび使用方法

- 共通フィールド表示変数（4-27 ページ）
- フィールド表示（5-24 ページ）

説明

この変数を空以外の値に設定すると、フィールドのオプション・リストを表示するときに eval（6-205 ページ）関数が使用されます。この変数により、オプション・リストの標準実装（defaultOptionListScript（6-142 ページ）変数で定義）を上書きできます。

出力

- 値が空以外の場合は、TRUE を返します。
- 値が空の文字列の場合は、FALSE を返します。

例

オプション・リストを作成するカスタム・スクリプトを定義します。

```
<$customOptionListScript = getValue("#active",
fieldName & ":optionListScript")$>
```

関連項目

– defaultOptionListScript（6-142 ページ）

OPTIONLISTVALUEINCLUDE

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

オプション・リストの値を定義するインクルードを指定します。

出力

インクルード・コードを返します。

例

std_override_format_option_value をオプション・リスト値のインクルードとして定義します。

```
<@dynamichtml std_override_format_field@>
<$if not isInfo and IsOverrideFormat$>
    <$fieldIsOptionList = 1, optionListResultSet = "DocFormats",
    optionListKey = "dFormat", optionListValueInclude =
    "std_override_format_option_value", addEmptyOption = 1,
    emptyOptionLabel = lc("wwEmptyFormatOption")$>
    <$include std_display_field$>
<$endif$>
<@end@>

<@dynamichtml std_override_format_option_value@>
<$dDescription$>
<@end@>
```

ResultSet からオプション・リストのオプションを表示するために使用するインクルードを指定します。

```
<@dynamichtml std_resultset_option_list_item@>
    <$curValue = getValue("#active", optionListKey)$>
    <option value="<$curValue$>" <$if optionsAllowPreselect and
    strEquals(curValue, fieldValue)$>selected<$endif$>>
    <$if optionListValueInclude$>
        <$inc(optionListValueInclude)$>
    <$else$>
        <$curValue$>
    <$endif$>
<@end@>
```

関連項目

– optionListResultSet（6-567 ページ）

OPTIONSALLOWPRESELECT

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

メタデータ・フィールドのオプション・リストに、前回のオプション・リストの値を事前に含めることが可能かどうかを指定します。

出力

- オプション・リストに事前に値を挿入できる場合は、TRUE を返します。
- オプション・リストに事前に値を挿入できない場合は、FALSE を返します。

例

オプション・リストに事前に値を挿入できることを指定します。

```
<$optionsAllowPreselect=1$>
```

オプション・リストに事前に値が挿入されるかどうかを決定します。

```
<$if optionsAllowPreselect and fieldValue$>
    <$defaultOptionListScript = "<$optList " & optionListName & ":fieldValue$>"$>
<$else$>
    <$defaultOptionListScript = "<$optList " & optionListName & ":noselected$>"$>
<$endif$>
```

OPTLIST

タイプおよび使用方法

❖ グローバル関数（4-21 ページ）

❖ フィールド表示（5-24 ページ）

説明

オプション・リストを生成します。

❖ この関数は、コンテンツ・サーバー・ページのオプション・リストの作成によく使用されています。

パラメータ

1 つのパラメータおよびパラメータに対する 1 つのオプションの引数があります。

❖ パラメータは、フィールド名、オプション・リスト・キーまたは変数のみです。

- **フィールド名**の構文は、`<$optList fieldName$>` です。カスタム・メタデータ・フィールドの場合、フィールド名は `<$optList xFieldName.options$>` になります。または、`.options` 接尾辞を直接に指定することもできます。
- **オプション・リスト・キー**の構文は、`<$optList ListName$>` です。
  - 標準メタデータ・フィールドの場合、*ListName* は、内部オプション・リストの名前です（*docAuthors* など）。3-33 ページの「内部オプション・リスト」を参照してください。
  - カスタム・メタデータ・フィールドの場合、*ListName* は、オプション・リスト・キーの名前です（*FieldNameList* など）。
- **変数**の構文は、`<$optList variable$>` です。変数は、フィールド名またはオプション・リスト・キーである必要があります。

❖ 次のいずれかのオプションの引数をパラメータに追加できます。

- **:noselected**: オプション・リストが表示されたとき、値は選択されていません。
- **:fieldValue**: *fieldValue* 変数によって指定された値が、オプション・リストのデフォルト値として選択されます。

出力

値のリストを返します。

例

このスクリプトは、内部 *docAuthors* リストから可能な作成者のリストを生成します。

```
<$optList docAuthors$>
```

このスクリプトは、*xRegion* カスタム・メタデータ・フィールドに指定されたオプションのリストを生成します。

```
<$optList xRegion.options$>
```

このスクリプトは、変数 *optionListName* からオプション・リストを生成し、デフォルト値を指定します。

```
<$if optionsAllowPreselect and fieldValue$>
    <$defaultOptionListScript = "<$optList " & optionListName & ":fieldValue$>"$>
<$else$>
    <$defaultOptionListScript = "<$optList " & optionListName & ":noselected$>"$>
<$endif$>
```

関連項目

- オプション・リスト（3-33 ページ）
- defaultOptionListScript（6-142 ページ）
- optionListName（6-565 ページ）
- optionListScript（6-568 ページ）
- optionsAllowPreselect（6-571 ページ）
- rsMakeFromList（6-677 ページ）

PAGEMAKEREXEPATH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

Adobe PageMaker 実行可能ファイルのパスを定義します。

- 文字列としてパスを返します。
- デフォルトは、空の文字列です。

場所

- 「Shared Inbound Refinery Configuration」画面→「Paths」タブ→「PageMaker Executable File Path」
- <*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
PageMakerExePath=c:/Program Files/Adobe/PageMaker 6.5/pagemaker.exe
```

PAGEPARENT

タイプおよび使用方法

- 値変数（4-34 ページ）
- ページ表示（5-30 ページ）

説明

ライブラリ内のディレクトリ・ページに親ページがあるかどうかをチェックします。

出力

- ディレクトリ・ページが、別のディレクトリ・ページの子（サブフォルダ）である場合は、TRUE を返します。
- ディレクトリ・ページが子（サブフォルダ）でない場合は、FALSE を返します。

例

ディレクトリ・ページがサブフォルダかどうかをチェックします。

```
<$PageParent$>
```

PARSECLBRAPROJECT

タイプおよび使用方法

- 動的変数（4-20 ページ）
- Collaboration Manager（5-5 ページ）

説明

この関数は、Collaboration Manager プロジェクト名のアカウント名を返します。

パラメータ

1 つのパラメータを取ります。

- 返されるアカウント名のプロジェクト

出力

アカウント名を返します。

PARSEDATAENTRYDATE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 日付と時間（5-19 ページ）

説明

日付を解析しますが、代替解析フォーマットを使用するためのフェイルオーバー・ロジックを使用します。

- 標準の比較演算子とともに使用する場合、日付はミリ秒に変換されます。たとえば、式（60*60*1000）は、1 時間と同等です。
- この関数は、秒数、分数、時間数または日数を表す乗算式を使用して、現在の時間を調整するために一般的に使用されます。

パラメータ

1 つのパラメータを取ります。

- パラメータは、解析する式のみです。

出力

Java 日付オブジェクトを返します。この日付は、表示のために文字列に変換されます。

例

```
<$parseDataEntryDate$>
```

PARSEDATE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 日付と時間（5-19 ページ）

説明

評価および算術演算ができるように、日付 / 時間を解析します。

- 標準の比較演算子とともに使用する場合、日付はミリ秒に変換されます。たとえば、式（60*60*1000）は、1 時間と同等です。
- この関数は、秒数、分数、時間数または日数を表す乗算式を使用して、現在の時間を調整するために一般的に使用されます。

パラメータ

1 つのパラメータを取ります。

- パラメータは、解析する式のみです。

出力

Java 日付オブジェクトを返します。この日付は、表示のために文字列に変換されます。

例

次の各式は、1 日前の日付と時間を返します。

```
<$parseDate(dateCurrent(-1))$>
<$parseDate(dateCurrent()-(24*60*60*1000))$>
<$dateCurrent(-1)$>
<$dateCurrent()-(24*60*60*1000)$>
```

1 時間後の時間を返します。最初の行で、時間の乗算式を使用して 1 時間加算し、その時間と日付をカスタム変数に割り当て、出力を抑制します。2 行目で、カスタム変数を参照し、時間のみを表示するように定義します。

```
<$exec my_customParseTime parseDate(dateCurrent()+(1000*60*60))$>
<$formatTimeOnly(my_customParseTime)$>
```

1 年後の日付を返します。最初の行で、時間の乗算式を使用して 1 年を加算し、その時間と日付をカスタム変数に割り当て、出力を抑制します。2 行目で、カスタム変数を参照し、長い書式の日付のみを表示するように定義します。

```
<$exec my_customParseTime parseDate(dateCurrent()+(1000*60*60*24*365))$>
<$formatTimeOnly(my_customParseTime)$>
```

このスクリプトは、7 日後の日付が有効期限より大きいかどうかを評価し、true の場合はユーザーにメッセージを返します。

```
<$if dOutDate$>
    <$if dateCurrent(7) > parseDate(dOutDate)$>
        Content item expires in one week.
    <$endif$>
<$endif$>
```

このスクリプトでは、カスタマイズされたワークフロー・ジャンプの条件文の中で `parseDate` を使用しています。このスクリプトでは、このステップに最後に入った時間から 4 日間経過している場合、ワークフロー *wf_late* の最初のステップに移動し、次のステップに戻されるように設定しています。

```
<$if parseDate(wfCurrentGet("lastEntryTs")) < dateCurrent(-4)$>
    <$wfSet("wfJumpName", "lateJump")$>
    <$wfSet("wfJumpTargetStep", "step_1@wf_late")$>
    <$wfSet("wfJumpReturnStep", wfCurrentStep(1))$>
    <$wfSet("wfJumpEntryNotifyOff", "0")$>
<$endif$>
```

関連項目

– dateCurrent（6-121 ページ）

PARSEDATEWITHPATTERN

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 日付と時間（5-19 ページ）

説明

日付 / 時間を指定された日付 / 時間パターンに解析します。

- 標準の比較演算子とともに使用する場合、日付はミリ秒に変換されます。たとえば、式（`60*60*1000`）は、1 時間と同等です。
- この関数は、秒数、分数、時間数または日数を表す乗算式を使用して、現在の時間を調整するために一般的に使用されます。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、コンテンツ・サーバーによって使用される日付文字列、または parseDate（6-578 ページ）または dateCurrent（6-121 ページ）関数で作成された日付オブジェクトです。
- 2 番目のパラメータは、日付 / 時間パターンです。これは、標準の Java *SimpleDateFormat* パターン文字列です（MM/dd/yyyy など）。

出力

Java 日付オブジェクトを返します。この日付は、表示のために文字列に変換されます。

例

現在の日付と時間をパターンで指定された書式で表示します
（たとえば、Wed, 4 Jul 2001 12:08:56 -0700）。

```
<$parseDateWithPattern(<$dateCurrent()$>,"EEE, d MMM yyyy HH:mm:ss Z")$>
```

関連項目

- parseDate（6-578 ページ）
- dateCurrent（6-121 ページ）
- formatDateWithPattern（6-251 ページ）

PATH_INFO

タイプおよび使用方法

- Web サーバー変数（4-36 ページ）
- Web サーバー（5-42 ページ）

説明

重要 : この設定は、バージョン 7.0 以降では不要です。Web サーバー・フィルタで、この情報は送信されなくなりました。

CGI マシンのファイル・システム・パスに関する追加の情報を取得します。

- PATH_TRANSLATED（6-582 ページ）変数によって仮想パスが返されるとき、このパスの末尾にある追加の情報も PATH_INFO として返されます。
- この変数は、現在のゲートウェイ・プログラム・リクエストに固有のものです。

出力

追加の仮想パス情報を文字列として返します。

例

追加の CGI パス情報を取得します。

```
<$PATH_INFO$>
```

関連項目

– PATH_TRANSLATED（6-582 ページ）

PATH_TRANSLATED

タイプおよび使用方法

- Web サーバー変数（4-36 ページ）
- Web サーバー（5-42 ページ）

説明

CGI マシンのファイル・システム・パスを取得します。

- たとえば、`c:/stellent/weblayout/idcplg/idc_cgi_isapi-idcm1.dll`。
- この変数は、現在のゲートウェイ・プログラム・リクエストに固有のものです。

出力

文字列として仮想パスを返します。

例

CGI パスを取得します。

```
<$PATH_TRANSLATED$>
```

PDFVERSION

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

PDF Converter によって作成された PDF ファイルのバージョンを設定します。

- この設定を有効にするには、SetPdfVersion（6-736 ページ）が TRUE である必要があります。
- 1.3 に設定すると、PDF Converter によって、PDF バージョン 1.3（Acrobat 4.0 準拠）ファイルが作成されます。
- 1.4 に設定すると、PDF Converter によって、PDF バージョン 1.4（Acrobat 5.0 準拠）ファイルが作成されます。
- この設定を使用して、旧式の PDF ファイルを新しいバージョンに再保存できます。
- デフォルトは、1.4 です。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
PdfVersion=1.3
```

関連項目

– SetPdfVersion（6-736 ページ）

PLUGIN_DEBUG

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Web サーバー（5-42 ページ）
- デバッグ（5-20 ページ）

説明

Web サーバー・プラグイン・フィルタ内で発生するイベントのログを有効にします。

- TRUE に設定すると、このフラグを理解するプラグイン内で発生したイベントは、次のログ・ファイルに記録されます。
  - **IIS:** <*install_dir*>/idcplg/idc_cgi_isapi-<*instance*>.dll.log
  - **Apache または NES:** <*install_dir*>/data/users/authfilt.log
- IIS または Content Server v3.72 以下でこの設定を変更する場合、コンテンツ・サーバーおよび Web サーバーを再起動する必要があります。
- Content Server v3.72 以降での Apache および Netscape でこの設定を変更する場合は、コンテンツ・サーバーのみを起動する必要があります。
- デフォルトは、FALSE です。

場所

<*install_dir*>/data/users/config/filter.hda

例

次は構成エントリの使用例です。

```
PLUGIN_DEBUG=true
```

関連項目

- CGI_DEBUG（6-64 ページ）
- CGI_RECEIVE_DUMP（6-65 ページ）
- CGI_SEND_DUMP（6-66 ページ）
- FILTER_DEBUG（6-234 ページ）

PNENAVIGATION

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ページ表示（5-30 ページ）

説明

左サイドバー・ナビゲーションを有効にします。

パラメータ

名前 / 値ペアとして設定します。

```
pneNavigation=1
```

- デフォルト値は、1（有効）です。
- この機能を無効にするには、NULL 文字列に設定します。

出力

TRUE または FALSE。

例

サイドバー・ナビゲーションを有効にします。

```
<$pneNavigation=1$>
```

サイドバー・ナビゲーションをオフにするには、NULL 文字列に設定します。

```
<$pneNavigation=""$>
```

定義を 1 以外または NULL 文字列値に設定するのは無効であり、サイドバー・ナビゲーションは無効になりません。

```
<$pneNavigation=0$>
```

POSTCONVERSIONTIMEOUT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Inbound Refinery（5-11 ページ）

説明

変換後処理のデフォルトのタイムアウトを定義します。

- 変換後処理のタイムアウト値を返します。
- この変数の後に、次のいずれかのフラグが必要です。

フラグ	説明
#factor	この値とファイル・サイズを掛け合せて、変換後の処理の許容時間を決定します。（結果の時間は、#max および #min フラグによって制限されます。）デフォルト値は、1 です。ビジーまたは処理の遅いシステムでは、この値を増加します。
#max	変換後の処理が完了するまでの Inbound Refinery の最大待機時間を秒数で指定します。デフォルトは、10 秒です。
#min	変換後の処理が完了するまでの Inbound Refinery の最小待機時間を秒数で指定します。デフォルトは、1 秒です。

場所

- 「Shared Inbound Refinery Configuration」画面→「Timeout Values」タブ→「Post PDF Production Timeout」
- <*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
PostConversionTimeOut#factor=1
```

POSTPROCESSPDFPATH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

変換後の PDF ファイルの処理に使用される実行可能ファイルのパスを定義します。

- コマンドライン・パラメータは、PDF ファイルのパスのみです。
- 指定するファイルは、Win32 実行可能ファイルである必要があります。
- このプロセスは、PDF への変換後、PDF の最適化の前に起動されます。Inbound Refinery によって処理されるファイルごとに 1 回起動されるため、ファイルの変換に同期させる必要のある他のイベントに使用することもできます。値が空でない場合、Inbound Refinery はプロセスの実行を試みます。
- デフォルトは、空の文字列です。

場所

- 「Local Inbound Refinery Configuration」画面→「General」タブ→「Post PDF Production Path」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
PostprocessPDFPath=c:/stellent/IdcRefinery/ConversionProcess/
IDCWaterPost.exe
```

POSTSCRIPTPRINTERNAME

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

ファイルを PostScript に変換するために使用するプリンタの名前を指定します。

- 有効な PostScript プリンタ名を指定する必要があります。
- デフォルトは、*IDC PDF Converter* です。

場所

<*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
PostscriptPrinterName=Acrobat Distiller
```

POWERPOINTPRINTOUTPUT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

Microsoft PowerPoint コンテンツ・アイテムの変換方法を定義します。

- *slide* に設定すると、PDF ファイルの各ページに 1 枚のスライドが含まれます。
- *notes* に設定すると、PDF ファイルの各ページに、1 枚のスライドとその注釈が含まれます。
- *handout* に設定すると、PDF ファイルの各ページに 6 枚のスライドが含まれます。
- デフォルトは、*slide* です。

場所

- 「Local Inbound Refinery Configuration」画面→「Native Options」タブ→「Convert in PowerPoint format options」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
PowerPointPrintOutput=notes
```

関連項目

– ProcessPowerpointHyperlinks（6-606 ページ）

PRECONVERSIONPATH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Inbound Refinery（5-11 ページ）

説明

変換前のファイルの処理に使用される実行可能ファイルのパスを定義します。

- 指定するファイルは、Win32 実行可能ファイルである必要があります。
- デフォルトは、空の文字列です。

場所

- 「Local Inbound Refinery Configuration」画面→「Legacy Conversions」タブ→「Preconversion Path」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
PreconversionPath=c:/out/preconversion.bat
```

PRECONVERSIONWAITTIME

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Inbound Refinery（5-11 ページ）

説明

変換前処理のタイムアウトを定義します。

- 変換前処理のタイムアウト値を返します。
- この変数の後に、次のいずれかのフラグが必要です。

フラグ	説明
#factor	この値とファイル・サイズを掛け合せて、変換前の処理の許容時間を決定します。（結果の時間は、#max および #min フラグによって制限されます。）デフォルト値は、1 です。ビジーまたは処理の遅いシステムでは、この値を増加します。
#max	変換前処理が完了するまでの Inbound Refinery の最大待機時間を秒数で指定します。デフォルトは、10 秒です。
#min	変換前処理が完了するまでの Inbound Refinery の最小待機時間を秒数で指定します。デフォルトは、1 秒です。

場所

- 「Local Inbound Refinery Configuration」画面→「Legacy Conversions」タブ→「Pre-Converter Timeout」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
PreconversionWaitTime#max=60
```

PREVIEWOUTPUTEXTENSION

タイプおよび使用方法

- 構成変数（4-3 ページ）
- HTML プレビュー（5-26 ページ）

説明

HTML プレビューで使用される、Content Publisher から出力されるファイル拡張子を定義します。

- デフォルトは、*HCSP* です。

場所

<*install_dir*>/bin/intradoc.cfg

例

次は構成エントリの使用例です。

```
PreviewOutputExtension=jsp
```

PREVIEWPATH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- HTML プレビュー（5-26 ページ）

説明

HTML プレビュー実行可能ファイルのパスを定義します。

- *tcpreview.exe* ファイルは、Content Publisher のインストール・ディレクトリにあります。

場所

<*install_dir*>/bin/intradoc.cfg

例

次は構成エントリの使用例です。

```
PreviewPath=C:/Program Files/Stellent Content Publisher/tcpreview.exe
```

PRIMARYWORKQUEUETIMEOUT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）
- ワークフロー（5-44 ページ）

説明

ワークフローおよびサブスクリプションの電子メール通知が送信されるまでの秒数を設定します。

- 電子メールを延期する必要がない場合、またはデバッグ中に即時のレスポンスが必要な場合は、小さい数値に設定します。
- デフォルトは、600（10 分）です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
PrimaryWorkQueueTimeout=30
```

PRINTERPORTPATH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

Inbound Refinery によって作成される PostScript ファイルの場所を定義します。

- この設定は、Inbound Refinery によって使用されるプリンタのポート設定と一致している必要があります。
- デフォルトは、*c:/temp/idcoutput.ps* です。

場所

<*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
PrinterPortPath=c:/temp/idcrefinery/idcoutput.ps
```

関連項目

– PostscriptPrinterName（6-588 ページ）

PRIORITY

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

ユーザーの資格証明について LDAP プロバイダをチェックする順序を指定します。

- このフィールドは、以前にコンテンツ・サーバーにログインしていないユーザーの場合にのみ使用されます。ユーザーが以前に資格証明を要求している場合は、そのユーザーの SourcePath（6-754 ページ）が保存されるため、ソース・パスに指定された LDAP プロバイダが使用されます。
- コンテンツ・サーバーの各 LDAP プロバイダに、一意の優先度の数値を指定する必要があります。
- デフォルトは、1 です。

場所

- 「Add/Edit LDAP Provider」ページ→「Priority」
- <*install_dir*>/data/providers/*provider_name*/provider.hda

例

次は構成エントリの使用例です。

```
Priority=2
```

関連項目

– SourcePath（6-754 ページ）
– IsPrimaryUserBase（6-428 ページ）

PRJUSECUSTOMROUTING

タイプおよび使用方法

- Collaboration Manager の構成変数（4-4 ページ）
- Collaboration Manager（5-5 ページ）

説明

コラボレーション・プロジェクト内でのワークフロー・コンテンツのルーティングを有効にするかどうかを指定します。

- Collaboration Manager のインストール中に追加されます。
- TRUE に設定すると、ワークフロー・ルーティングは有効になります。
- デフォルトは、TRUE です。

場所

<*install_dir*>/<*instancename*>/data/components/CollaborationManager/config.cfg

例

次は構成エントリの使用例です。

```
prjUseCustomRouting=true
```

PRJWIZARDTOKENS

タイプおよび使用方法

- Collaboration Manager の構成変数（4-4 ページ）
- Collaboration Manager（5-5 ページ）

説明

ワークフロー・トークンおよびメタデータ・フィールドの作成に使用する名前のリストをカンマで区切って指定します。

- Collaboration Manager のインストール中に追加されます。

場所

<*install_dir*>/<*instancename*>/data/components/CollaborationManager/config.cfg

例

```
prjWizardTokens=token1,token2,token3
```

PRJWIZARDWORKFLOWFIELD

タイプおよび使用方法

- Collaboration Manager の構成変数（4-4 ページ）
- Collaboration Manager（5-5 ページ）

説明

Collaboration ワークフローの基準として使用するメタデータ・フィールドを指定します。

- Collaboration Manager のインストール中に追加されます。

場所

<*install_dir*>/<*instancename*>/data/components/CollaborationManager/config.cfg

例

```
prjWizardWorkflowField=metadataname
```

PROCESSEXCELURLHYPERLINKS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

Microsoft Excel コンテンツ・アイテムの URL ハイパーリンク処理を有効にします。

- TRUE に設定すると、Excel ファイルの URL ハイパーリンクを処理します。
- FALSE に設定すると、Excel ファイルの URL ハイパーリンクは処理されません。
- デフォルトは、FALSE です。

場所

- 「Local Inbound Refinery Configuration」画面→「Native Options」タブ→「Process Excel URL links」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
ProcessExcelUrlHyperlinks=true
```

PROCESSINDESIGNBOOKMARKS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

Adobe InDesign コンテンツ・アイテムの内部ドキュメント・リンクの変換を有効にします。

- TRUE に設定すると、Adobe InDesign ファイルの内部ドキュメント・リンクが処理されます。
- FALSE に設定すると、Adobe InDesign ファイルの内部ドキュメント・リンクは処理されません。
- デフォルトは、TRUE です。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
ProcessInDesignBookmarks=false
```

関連項目

- ProcessInDesignHyperlinks（6-602 ページ）
- ProcessInDesignInternalPDFThumbnail（6-603 ページ）
- ProcessInDesignLinkUpdate（6-604 ページ）
- ProcessInDesignPDFCompatibility50（6-605 ページ）

PROCESSINDESIGNHYPERLINKS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

Adobe InDesign コンテンツ・アイテムの URL リンクの変換を有効にします。

- TRUE に設定すると、Adobe InDesign ファイルの URL リンクが処理されます。
- FALSE に設定すると、Adobe InDesign ファイルの URL リンクは処理されません。
- デフォルトは、TRUE です。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
ProcessInDesignHyperlinks=false
```

関連項目

- ProcessInDesignBookMarks（6-601 ページ）
- ProcessInDesignInternalPDFThumbnail（6-603 ページ）
- ProcessInDesignLinkUpdate（6-604 ページ）
- ProcessInDesignPDFCompatibility50（6-605 ページ）

PROCESSINDESIGNINTERNALPDFTHUMBNAIL

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

Adobe InDesign コンテンツ・アイテムを作成するとき、PDF ページのサムネイルの作成を有効にします。

- TRUE に設定すると、PDF ファイルにページのサムネイルが含まれます。
- FALSE に設定すると、PDF ファイルにページのサムネイルは含まれません。
- この設定は、検索結果ページのサムネイルの作成には影響しません。
- デフォルトは、TRUE です。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
ProcessInDesignInternalPDFThumbnail=false
```

関連項目

- ProcessInDesignBookMarks（6-601 ページ）
- ProcessInDesignHyperlinks（6-602 ページ）
- ProcessInDesignLinkUpdate（6-604 ページ）
- ProcessInDesignPDFCompatibility50（6-605 ページ）

PROCESSINDESIGNLINKUPDATE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

変換する前に、Adobe InDesign コンテンツ・アイテムのすべてのリンクを更新できます。

- TRUE に設定すると、Adobe InDesign ファイルのリンクは、変換の前に更新されます。リンク・オブジェクトが使用可能でない場合、変換は失敗します。
- FALSE に設定すると、Adobe InDesign ファイルのリンクは、変換の前に更新されません。
- デフォルトは、FALSE です。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
ProcessInDesignLinkUpdate=true
```

関連項目

- ProcessInDesignBookMarks（6-601 ページ）
- ProcessInDesignHyperlinks（6-602 ページ）
- ProcessInDesignInternalPDFThumbnail（6-603 ページ）
- ProcessInDesignPDFCompatibility50（6-605 ページ）

PROCESSINDESIGNPDFCOMPATIBILITY50

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

PDF Converter によって、Adobe InDesign コンテンツ・アイテムから作成される PDF ファイルのバージョンを設定します。

- TRUE に設定すると、PDF Converter によって、Adobe InDesign ファイルから PDF バージョン 1.4（Acrobat 5.0 準拠）ファイルが作成されます。
- FALSE に設定すると、PDF Converter によって、Adobe InDesign ファイルから PDF バージョン 1.3（Acrobat 4.0 準拠）ファイルが作成されます。
- この設定は、Adobe InDesign ファイルの PdfVersion（6-583 ページ）の設定を上書きします。
- デフォルトは、FALSE です。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
ProcessInDesignPDFCompatibility50=true
```

関連項目

– PdfVersion（6-583 ページ）
– SetPdfVersion（6-736 ページ）

PROCESSPOWERPOINTHYPERLINKS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

Microsoft PowerPoint コンテンツ・アイテムのハイパーリンクの処理を有効にします。

- この変数を TRUE に設定し、PowerPointPrintOutput（6-589 ページ）設定を SLIDE に設定すると、PowerPoint ファイルで次のハイパーリンクが処理されます。
  - URL リンク
  - 次のスライド
  - 前のスライド
- FALSE に設定すると、PowerPoint ファイルのハイパーリンクは処理されません。
- *PowerPointPrintOutput* 設定が NOTES または HANDOUT に設定されている場合、PowerPoint ファイルのハイパーリンクは処理されません。
- デフォルトは、FALSE です。

場所

- 「Local Inbound Refinery Configuration」画面→「Native Options」タブ→「Process PowerPoint Hyperlinks」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
ProcessPowerpointHyperlinks=true
```

関連項目

– PowerPointPrintOutput（6-589 ページ）

PROCESSVISIOINTERNALHYPERLINKS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

Microsoft Visio コンテンツ・アイテムの内部ハイパーリンクの変換を有効にします。

- TRUE に設定すると、Visio ファイルの内部ハイパーリンクを処理します。
- FALSE に設定すると、Visio ファイルの内部ハイパーリンクは処理されません。
- デフォルトは、FALSE です。

場所

- 「Local Inbound Refinery Configuration」画面→「Native Options」タブ→「Process internal Visio links」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
ProcessVisioInternalHyperlinks=true
```

関連項目

– ProcessVisioUrlHyperlinks（6-608 ページ）

PROCESSVISIOURLHYPERLINKS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

Microsoft Visio コンテンツ・アイテムの URL リンクの変換を有効にします。

- TRUE に設定すると、Visio ファイルの URL リンクを処理します。
- FALSE に設定すると、Visio ファイルの URL リンクは処理されません。
- デフォルトは、FALSE です。

場所

- 「Local Inbound Refinery Configuration」画面→「Native Options」タブ→「Process Visio URL links」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
ProcessVisioUrlHyperlinks=true
```

関連項目

– ProcessVisioInternalHyperlinks（6-607 ページ）

PROCESSWORDBOOKMARKS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

Microsoft Word コンテンツ・アイテムのページ参照ブックマークの変換を有効にします。

- TRUE に設定すると、Microsoft Word ファイルのページ参照ブックマークが PDF ブックマークに変換されます。（他のタイプのブックマークはすべて無視されます。）
- FALSE に設定すると、Microsoft Word ファイルのブックマークは処理されません。
- デフォルトは、TRUE です。

場所

- 「Local Inbound Refinery Configuration」画面→「Native Options」タブ→「Process document bookmarks」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
ProcessWordBookMarks=false
```

関連項目

– ProcessWordHyperlinks（6-614 ページ）

PROCESSWORDCOMMENTS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

Microsoft Word コンテンツ・アイテムのコメントを PDF の注釈に変換できます。

- TRUE に設定すると、Microsoft Word ファイルのコメントが PDF の注釈に変換されます。
- FALSE に設定すると、Microsoft Word ファイルのコメントは処理されません。
- デフォルトは、FALSE です。

場所

- 「Local Inbound Refinery Configuration」画面→「Native Options」タブ→「Process document comments to PDF notes」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
ProcessWordComments=true
```

関連項目

– ProcessWordHyperlinks（6-614 ページ）

PROCESSWORDEMAILLINKS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

Microsoft Word コンテンツ・アイテムの電子メール・ハイパーリンクの変換を有効にします。

- この設定が有効になるには、ProcessWordUrlHyperlinks（6-616 ページ）設定が TRUE である必要があります。
- TRUE に設定すると、Microsoft Word ファイルの電子メール・ハイパーリンクが処理されます。
- FALSE に設定すると、Microsoft Word ファイルの電子メール・ハイパーリンクは処理されません。
- デフォルトは、FALSE です。

場所

- 「Local Inbound Refinery Configuration」画面→「Native Options」タブ→「Process mailto links」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
ProcessWordEmailLinks=true
```

関連項目

- ProcessWordHyperlinks（6-614 ページ）
- ProcessWordUrlHyperlinks（6-616 ページ）

PROCESSWORDFOOTENDNOTE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

Microsoft Word コンテンツ・アイテムの脚注および巻末の注をハイパーリンクに変換できます。

- TRUE に設定すると、Microsoft Word ファイルの脚注および巻末の注がハイパーリンクに変換されます。
- FALSE に設定すると、Microsoft Word ファイルの脚注および巻末の注は処理されません。
- デフォルトは、FALSE です。

場所

- 「Local Inbound Refinery Configuration」画面→「Native Options」タブ→「Process document links to footnotes and endnotes」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
ProcessWordFootEndNote=true
```

関連項目

– ProcessWordHyperlinks（6-614 ページ）

PROCESSWORDHEADINGS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

Microsoft Word コンテンツ・アイテムの標準見出しを PDF ブックマークに変換できます。

- TRUE に設定すると、Microsoft Word ファイルの標準見出しの段落スタイル（見出し 1 ～見出し 9 およびこれらの標準スタイルに基づいたカスタム・スタイル）は、PDF ブックマークに変換されます。
- ProcessWordToc（6-615 ページ）設定を使用して目次がすでに処理されている場合、この設定は無視され、見出しはブックマークに変換されません。
- FALSE に設定すると、Microsoft Word ファイルの見出しスタイルは処理されません。
- デフォルトは、TRUE です。

場所

- 「Local Inbound Refinery Configuration」画面→「Native Options」タブ→「Process heading styles to PDF bookmarks」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
ProcessWordHeadings=false
```

関連項目

- ProcessWordHyperlinks（6-614 ページ）
- ProcessWordToc（6-615 ページ）

PROCESSWORDHYPERLINKS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

Microsoft Word コンテンツ・アイテムのハイパーリンクの変換を有効にします。

- TRUE に設定すると、Microsoft Word ファイルのすべてのタイプのハイパーリンクが処理されます。
- FALSE に設定すると、Microsoft Word ファイルのハイパーリンクは処理されません。
- デフォルトは、TRUE です。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
ProcessWordHyperlinks=true
```

関連項目

- ProcessWordBookMarks（6-609 ページ）
- ProcessWordComments（6-610 ページ）
- ProcessWordEmailLinks（6-611 ページ）
- ProcessWordFootEndNote（6-612 ページ）
- ProcessWordHeadings（6-613 ページ）
- ProcessWordToc（6-615 ページ）
- ProcessWordUrlHyperlinks（6-616 ページ）

PROCESSWORDTOC

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

Microsoft Word コンテンツ・アイテムの目次を PDF ブックマークに変換できます。

- TRUE に設定すると、Microsoft Word ファイルで自動的に生成される目次エントリは PDF ブックマークに変換されます。
- FALSE に設定すると、Microsoft Word ファイルの目次は処理されません。（ただし、目次にハイパーリンクがある場合、ハイパーリンクは ProcessWordUrlHyperlinks（6-616 ページ）設定に基づいて処理されます。）
- デフォルトは、FALSE です。

場所

- 「Local Inbound Refinery Configuration」画面→「Native Options」タブ→「Process Table of Contents to PDF bookmarks」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
ProcessWordToc=true
```

関連項目

- ProcessWordHyperlinks（6-614 ページ）
- ProcessWordUrlHyperlinks（6-616 ページ）
- ProcessWordHeadings（6-613 ページ）

PROCESSWORDURLHYPERLINKS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

Microsoft Word コンテンツ・アイテムの URL ハイパーリンクの変換を有効にします。

- TRUE に設定すると、Microsoft Word ファイルの URL ハイパーリンクを処理します。
- FALSE に設定すると、Microsoft Word ファイルの URL ハイパーリンクは処理されません。
- デフォルトは、TRUE です。

場所

- 「Local Inbound Refinery Configuration」画面→「Native Options」タブ→「Process URL links」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
ProcessWordUrlHyperlinks=false
```

関連項目

– ProcessWordHyperlinks（6-614 ページ）

PROFILECACHETIMEOUTINMINS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Web サーバー（5-42 ページ）

説明

Web サーバー・フィルタがキャッシュのタイムアウトとして使用する時間（分）を設定します。

- デフォルトは、2 分です。

場所

<*install_dir*>/data/users/config/filter.hda

例

次は構成エントリの使用例です。

```
ProfileCacheTimeoutInMins=10
```

PROVIDERCLASS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

プロバイダを実装する Java クラスの名前を指定します。

- デフォルトは、プロバイダのタイプごとに異なります。

プロバイダ・タイプ	デフォルト値
Outgoing	intradoc.provider.SocketOutgoingProvider
Database	intradoc.jdbc.JdbcWorkspace
Incoming	intradoc.provider.SocketIncomingProvider
Preview	intradoc.provider.SocketOutgoingProvider
Ldapuser	intradoc.provider.LdapUserProvider

場所

- 「Add/Edit Provider」ページ→「Provider Class」
- <*install_dir*>/data/providers/*provider_name*/provider.hda

例

次は構成エントリの使用例です。

```
ProviderConnection=intradoc.provider.SocketOutgoingProvider
```

PROVIDERCONFIG

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

プロバイダに対する追加の構成を実行する Java クラスの名前を指定します。

- この設定は、接続クラスがすでにプロバイダである、データベース・プロバイダに最も役立ちます。
- デフォルトはありません。

場所

- 「Add/Edit Provider」ページ→「Configuration Class」
- <*install_dir*>/data/providers/*provider_name*/provider.hda

例

次は構成エントリの使用例です。

```
ProviderConfig=intradoc.provider.CustomClass
```

PROVIDERCONNECTION

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

プロバイダ接続を実装する Java クラスの名前を指定します。

- デフォルトは、プロバイダのタイプごとに異なります。

プロバイダ・タイプ	デフォルト値
Outgoing	intradoc.provider.SocketOutgoingConnection
Database	intradoc.jdbc.JdbcConnection
Incoming	intradoc.provider.SocketIncomingConnection
Preview	intradoc.provider.SocketOutgoingConnection
Ldapuser	intradoc.provider.LdapConnection

場所

- 「Add/Edit Provider」ページ→「Connection Class」
- <*install_dir*>/data/providers/*provider_name*/provider.hda

例

次は構成エントリの使用例です。

```
ProviderConnection=intradoc.provider.SocketOutgoingConnection
```

PROVIDERDIR

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

providers ディレクトリのパスを定義します。

- デフォルトは、*<install_dir>/data/providers* です。

場所

<install_dir>/config/config.cfg

例

次は構成設定の例です。

```
ProviderDir=c:/stellent/data/providers
```

PROVIDERTYPE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

プロバイダのタイプを指定します。

- 使用可能なプロバイダ・タイプは次のとおりです。

プロバイダ・タイプ	値
Outgoing	outgoing
Database	database
Incoming	incoming
Preview	preview
Ldapuser	ldapuser

場所

<*install_dir*>/data/providers/*provider_name*/provider.hda

例

次は構成エントリの使用例です。

```
ProviderType=ldapuser
```

PROXIEDBROWSERFULLCGIWEBURL

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

プロキシ・コンテンツ・サーバーの完全な CGI パスを返します。

パラメータ

1 つのパラメータを取ります。

- パラメータは、プロキシ・コンテンツ・サーバーの相対 Web ルートのみです。この値は、*HttpRelativeWebRoot*（6-321 ページ）設定にあります。

出力

指定されたプロキシの完全な CGI パスを文字列で返します。

例

http://<localhost>/idcplg/idc_cgi_isapi-idcm1.dll/intradoc4/pxs を返します。

```
<$proxiedBrowserFullCgiWebUrl("/intradoc4/")$>
```

関連項目

– HttpRelativeWebRoot（6-321 ページ）

PROXIEDCGIWEBURL

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

プロキシ・コンテンツ・サーバーの CGI パスを返します。

パラメータ

1 つのパラメータを取ります。

- パラメータは、プロキシ・コンテンツ・サーバーの相対 Web ルートのみです。この値は、*HttpRelativeWebRoot*（6-321 ページ）設定にあります。

出力

指定されたプロキシの CGI パスを文字列で返します。

例

/idcplg/idc_cgi_isapi-idcm1.dll/intradoc4/pxs を返します。

```
<$proxiedCgiWebUrl("/intradoc4/")$>
```

関連項目

– HttpRelativeWebRoot（6-321 ページ）

PUCDEFAULTTIME

タイプおよび使用方法

- Pop-Up Calendar の構成変数（4-19 ページ）
- Pop-Up Calendar（5-31 ページ）

説明

ポップアップ・カレンダを使用するとき、日付 / 時間フィールドへの時間の入力方法を指定します。このオプションを設定しない場合、時間のデフォルトは、選択した日付の午前 0 時です。

この文字列の値は、次のように設定できます。

- current: クライアント・コンピュータの時間と同等の時間に設定します。
- システム・ロケールの書式で表現された任意の時間。値がない場合は、0 として解釈されます。AM/PM の指定がない場合は、24 時間表記の使用として解釈されます。たとえば、AM および PM を指定するシステム・ロケール内で、1:15 PM、01:15 PM および 13:15 は、同一として解釈されます。

場所

<*install_dir*>/config/config.cfg

例

```
pucDefaultTime=current
```

PUCENABLEONSEARCH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Pop-Up Calendar（5-31 ページ）

説明

TRUE に設定した場合、検索ページのポップアップ・カレンダが使用可能かどうかを指定します。FALSE に設定すると、検索ページのポップアップ・カレンダが無効になります。

デフォルトは、TRUE です。

場所

<*install_dir*>/config/config.cfg

例

```
pucEnableOnSearch=false
```

PUCHIDETIMEINDATEFIELDS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Pop-Up Calendar（5-31 ページ）

説明

日付フィールドの表示が、日付のみか、日付と時間かを指定します。

TRUE に設定すると、日付フィールドの時間の表示が無効化され、pucDefaultTime（6-625 ページ）の設定にかかわらず、Content Server によって、チェックイン時間が選択された日付の午前 0 時として解析されます。

false に設定すると、日付フィールドに時間が表示されます。

デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

```
pucHideTimeInDateFields=true
```

関連項目

– pucDefaultTime（6-625 ページ）

PUCUSEMONTHLOCALE

タイプおよび使用方法

- Pop-Up Calendar の構成変数（4-19 ページ）
- Pop-Up Calendar（5-31 ページ）

説明

ポップアップ・カレンダでの月名の表示方法を決定します。

この文字列の値は、次のように設定できます。

- short: ユーザーのロケールに基づいて、月名を切り捨てます。たとえば、January は Jan、February は Feb などのように切り捨てられます。
- long: ユーザーのロケールに基づいて、完全な月名が表示されます。

デフォルトは、long です。

例

```
pucUseMonthLocale=short
```

関連項目

– pucUseWeekdayLocale（6-629 ページ）

PUCUSEWEEKDAYLOCALE

タイプおよび使用方法

- Pop-Up Calendar の構成変数（4-19 ページ）
- Pop-Up Calendar（5-31 ページ）

説明

ポップアップ・カレンダでの曜日の表示方法を決定します。

この文字列の値は、次のように設定できます。

- 1char: 曜日の先頭の 1 文字を使用します。これは数値の 1 であり、L の小文字ではないことに注意してください。
- short: 2 文字を使用して曜日を表します。
- medium: 標準の 3 文字の曜日を使用します。
- long: ユーザーのロケールに基づいて、完全な名前が表示されます。

デフォルトは、long です。

例

```
pucUseWeekdayLocale=short
```

関連項目

– pucUseMonthLocale（6-628 ページ）

PUCXOFFSET

タイプおよび使用方法

- Pop-Up Calendar の構成変数（4-19 ページ）
- Pop-Up Calendar（5-31 ページ）

説明

ポップアップ・カレンダの水平位置を、デフォルトの位置に対して相対的に指定します（ピクセル単位）。この値は、整数として設定します。デフォルトは、0 です。

例

```
pucXOffset=10
```

関連項目

– pucYOffset（6-631 ページ）

PUCYOFFSET

タイプおよび使用方法

- Pop-Up Calendar の構成変数（4-19 ページ）
- Pop-Up Calendar（5-31 ページ）

説明

ポップアップ・カレンダの垂直位置を、デフォルトの位置に対して相対的に指定します（ピクセル単位）。この値は、整数として設定します。デフォルトは、0 です。

例

```
pucYOffset=10
```

関連項目

– pucXOffset（6-630 ページ）

PUCZINDEX

タイプおよび使用方法

- Pop-Up Calendar の構成変数（4-19 ページ）
- Pop-Up Calendar（5-31 ページ）

説明

ポップアップ・カレンダ・レイヤーの重ね順（Z 順序）でのオブジェクト位置を指定します。値が大きいほど、ページ・レイヤーの位置内で、より高い位置に置かれます。ページに指定された他の z-index スタイル属性よりも大きい正の数を使用することで、カレンダを他の要素の上に表示させることができます。

整数の値として設定します。デフォルトは、0 です。

例

```
pucZIndex=10
```

QUERY_STRING

タイプおよび使用方法

- Web サーバー変数（4-36 ページ）
- Web サーバー（5-42 ページ）
- 検索（5-34 ページ）

説明

問合せで URL の ? デリミタの後続の文字列を取得します。

- この変数は、現在の CGI リクエストに固有のものです。

出力

問合せ情報を文字列で返します。

例

次はページまたはログに出力される情報の例です。

```
QUERY_STRING=IdcService=GET_DOC_PAGE&Action=GetTemplatePage&Page=STD_QUERY_PAGE
```

次は Idoc スクリプト文または評価の一部としての使用例です。

```
<$if QUERY_STRING$>
<!--statement-->
```

QUEUEDIR

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Inbound Refinery（5-11 ページ）

説明

Inbound Refinery キューの作業ディレクトリのパスを定義します。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
QueueDir=x:/stellent_refinery/
```

RECORDSMANAGEMENTDENYAUTHORFREEPASS ONRMSECURITY

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Record Manager および Retention Manager（5-32 ページ）

説明

コンテンツ・アイテムが ACL 制限されたフォルダ内に置かれている場合（RW recordsmanager 別名）、アイテムを削除できる作成者を除き、ユーザーは RW アクセスのみを持ちます。この構成変数を true に設定すると、作成者によるアイテムの削除が禁止され、作成者は RW アクセスのみに制限されます。

デフォルトは、FALSE です。

場所

<*install_dir*>/custom/RecordsManagement/records_management_environments.cfg

例

次は構成エントリの使用例です。

RecordsManagementDenyAuthorFreePassOnRMSecurity=true

RECORDSMANAGEMENTNUMBEROVERWRITEONDELETE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Record Manager および Retention Manager（5-32 ページ）

説明

Record Manager 専用 : 破棄アクションのためのディスク・スクラブの実行回数を設定します。

デフォルトは、2 です。

場所

<*install_dir*>/custom/RecordsManagement/records_management_environments.cfg

例

次は構成エントリの使用例です。

```
RecordsManagementNumberOverwriteOnDelete=4
```

REDIRECTHOSTSFILTER

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Web サーバー（5-42 ページ）
- 外部セキュリティ（5-37 ページ）

説明

許可されるリダイレクト・ホストのフィルタを示すために使用されます。現在のホストは、このフィルタにデフォルトで追加されます。

- 複数のアドレスは、パイプ文字（|）を使用して区切ります。パイプの前後に空白が挿入されていないことを確認してください。
- アドレス・グループを示すには、ワイルドカードを使用します（* は 0 個以上の文字を表し、? は任意の 1 文字を表します）。
- デフォルトは、ローカル・ホストの URL です。

場所

- 「System Properties」→「Server」タブ→「IP Address Filter」
- <*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
RedirectHostsFilter=*.google.com|www.cnn.com
```

REGEXMATCHES

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 検索（5-34 ページ）

説明

正規表現を使用して特定のパターンの文字列を検索してマッチングを行います。正規表現の構成には、文字、文字クラス、その他のクラスおよび量指定子を含めることができます。Java API のクラス・パターンの詳細は、http://java.sun.com を参照してください。

- この機能は、JVM 1.4 以上のバージョンでのみ使用できます。これは、Content Server バージョン 7.0 以上のデフォルトのバージョンです。

パラメータ

この関数は、2 つのパラメータを取ります。

- 最初のパラメータは、一致させる文字列です。
- 2 番目のパラメータは、式です。

例

次の例では、文字列が式に一致しないため、FALSE を返します。

```
<$regexMatches("abcdef","abc")$>
```

次の例では、ワイルドカードが存在するため、TRUE を返します。ドット・アスタリスク（.*）表記のかわりに、アスタリスク（*）などの標準のワイルドカードを使用した場合、一致はしません。

```
<$regexMatches("abcdef", ".*abc.*")$>
```

関連項目

- 演算子（3-22 ページ）
- regexReplaceAll（6-639 ページ）
- regexReplaceFirst（6-640 ページ）
- strEquals（6-766 ページ）
- strIndexOf（6-769 ページ）

REGEXREPLACEALL

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 検索（5-34 ページ）

説明

正規表現を使用して特定のパターンの文字列を検索してマッチングおよび置換を行います。正規表現の構成には、文字、文字クラス、その他のクラスおよび量指定子を含めることができます。Java API のクラス・パターンの詳細は、http://java.sun.com を参照してください。

- この機能は、JVM 1.4 以上のバージョンでのみ使用できます。これは、Content Server バージョン 7.0 以上のデフォルトのバージョンです。
- 正規表現のすべてのインスタンスを置換文字列に置き換えます。

パラメータ

この関数は、3 つのパラメータを取ります。

- 最初のパラメータは、元の文字列です。
- 2 番目のパラメータは、一致させる文字列です。
- 3 番目のパラメータは、置換文字列です。

例

次の例は、`xyzdef xyzdef` を返します。

```
<$regexReplaceAll("abcdef abcdef","abc","xyz")$>
```

関連項目

- 演算子（3-22 ページ）
- regexMatches（6-638 ページ）
- regexReplaceFirst（6-640 ページ）
- strEquals（6-766 ページ）
- strIndexOf（6-769 ページ）
- strSubstring（6-779 ページ）

REGEXREPLACEFIRST

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 検索（5-34 ページ）

説明

正規表現を使用して特定のパターンの文字列を検索してマッチングし、最初のインスタンスを置換文字列に置換します。正規表現の構成には、文字、文字クラス、その他のクラスおよび量指定子を含めることができます。Java API のクラス・パターンの詳細は、http://java.sun.com を参照してください。

- この機能は、JVM 1.4 以上のバージョンでのみ使用できます。これは、Content Server バージョン 7.0 以上のデフォルトのバージョンです。

パラメータ

この関数は、3 つのパラメータを取ります。

- 最初のパラメータは、元の文字列です。
- 2 番目のパラメータは、一致させる文字列です。
- 3 番目のパラメータは、置換文字列です。

例

次の例は、`xyzdef abcdef` を返します。

```
<$regexReplaceFirst("abcdef abcdef","abc","xyz")$>
```

関連項目

- 演算子（3-22 ページ）
- regexMatches（6-638 ページ）
- regexReplaceAll（6-639 ページ）
- strEquals（6-766 ページ）
- strIndexOf（6-769 ページ）
- strSubstring（6-779 ページ）

REMOTE_ADDR

タイプおよび使用方法

❖ Web サーバー変数（4-36 ページ）

❖ Web サーバー（5-42 ページ）

説明

リクエストを行っているリモート・ホストの IP アドレスを返します。

❖ この変数は、現在のゲートウェイ・プログラム・リクエストに固有のものです。

出力

IP アドレスを文字列で返します。

例

次はページまたはログに出力される情報の例です。

```
REMOTE_ADDR=207.0.0.1
```

次は Idoc スクリプト文または評価の一部としての使用例です。

```
<$if REMOTE_ADDR$>
<!--statement-->
```

関連項目

– REMOTE_HOST（6-642 ページ）

REMOTE_HOST

タイプおよび使用方法

- Web サーバー変数（4-36 ページ）
- Web サーバー（5-42 ページ）

説明

リクエストを行っているリモート・ホストの名前を返します。

- この変数は、現在のゲートウェイ・プログラム・リクエストに固有のものです。

出力

- ホスト名を文字列で返します。
- サーバーにホスト名が不明な場合、REMOTE_ADDR（6-641 ページ）の値を文字列で返します。

例

次はページまたはログに出力される情報の例です。

```
REMOTE_HOST=207.0.0.1
```

次は Idoc スクリプト文または評価の一部としての使用例です。

```
<$if REMOTE_HOST$>
<!--statement-->
```

関連項目

– REMOTE_ADDR（6-641 ページ）

REQUEST_METHOD

タイプおよび使用方法

- Web サーバー変数（4-36 ページ）
- Web サーバー（5-42 ページ）

説明

リクエストを行うために使用したメソッドを取得します。

- この変数は、現在のゲートウェイ・プログラム・リクエストに固有のものです。

出力

リクエストのメソッドを文字列で返します。

例

次はページまたはログに出力される情報の例です。

```
REQUEST_METHOD=GET
```

次は Idoc スクリプト文または評価の一部としての使用例です。

```
<$if REQUEST_METHOD$>
<!--statement-->
```

REQUESTAUDITADDITIONALVERBOSEFIELDSLIST

タイプおよび使用方法

- 構成変数（4-3 ページ）
- デバッグ（5-20 ページ）

説明

リクエスト監査トレースがアクティブ化されたとき、冗長トレースとともに使用します。

- リクエスト監査トレースは、「System Audit Information」ページに表示されます。このトレースは、2 つの異なる時間間隔で計算された 2 つの異なるサマリーを作成します。トレースは、サービス・リクエスト数およびリクエストの実行にかかった時間を計算します。指定された間隔のサマリーが報告された後、値がリセットされ、次の期間の累計および平均が再び計算されます。
- この構成変数は、リクエストに関する情報を報告する際に表示される、追加のフィールドのカンマ区切りのリストです。

場所

<*install_dir*>/config/config.cfg

例

```
RequestAuditAdditionalVerboseFieldsList=xComment,dDocType
```

関連項目

- RequestAuditIntervalSeconds1（6-645 ページ）
- RequestAuditIntervalSeconds2（6-646 ページ）
- RequestAuditListDepth1（6-647 ページ）
- RequestAuditListDepth2（6-648 ページ）
- RequestAuditSubClassificationField（6-649 ページ）

REQUESTAUDITINTERVALSECONDS1

タイプおよび使用方法

- 構成変数（4-3 ページ）
- デバッグ（5-20 ページ）

説明

監査期間の短いリクエスト監査トレースを指定するために使用します。

- リクエスト監査トレースは、「System Audit Information」ページに表示されます。このトレースは、2 つの異なる時間間隔で計算された 2 つの異なるサマリーを作成します。トレースは、サービス・リクエスト数およびリクエストの実行にかかった時間を計算します。指定された間隔のサマリーが報告された後、値がリセットされ、次の期間の累計および平均が再び計算されます。
- 値を秒数で指定します。
- デフォルト値は、120 秒（2 分）です。

場所

<*install_dir*>/config/config.cfg

例

```
RequestAuditIntervalSeconds1=240
```

関連項目

- RequestAuditAdditionalVerboseFieldsList（6-644 ページ）
- RequestAuditIntervalSeconds2（6-646 ページ）
- RequestAuditListDepth1（6-647 ページ）
- RequestAuditListDepth2（6-648 ページ）
- RequestAuditSubClassificationField（6-649 ページ）

REQUESTAUDITINTERVALSECONDS2

タイプおよび使用方法

- 構成変数（4-3 ページ）
- デバッグ（5-20 ページ）

説明

監査期間の長いリクエスト監査トレースを指定するために使用します。

- リクエスト監査トレースは、「System Audit Information」ページに表示されます。このトレースは、2 つの異なる時間間隔で計算された 2 つの異なるサマリーを作成します。トレースは、サービス・リクエスト数およびリクエストの実行にかかった時間を計算します。指定された期間のサマリーが報告された後、値がリセットされ、次の期間の累計および平均が再び計算されます。
- 値を秒数で指定します。
- デフォルト値は、3600 秒（1 時間）です。

場所

<*install_dir*>/config/config.cfg

例

```
RequestAuditIntervalSeconds2=7200
```

関連項目

- RequestAuditAdditionalVerboseFieldsList（6-644 ページ）
- RequestAuditIntervalSeconds1（6-645 ページ）
- RequestAuditListDepth1（6-647 ページ）
- RequestAuditListDepth2（6-648 ページ）
- RequestAuditSubClassificationField（6-649 ページ）

REQUESTAUDITLISTDEPTH1

タイプおよび使用方法

- 構成変数（4-3 ページ）
- デバッグ（5-20 ページ）

説明

リクエスト監査トレースに表示するサービス数を指定するために使用します。

- この構成変数は、短い監査期間にのみ適用されます。
- リクエスト監査トレースは、「System Audit Information」ページに表示されます。このトレースは、2 つの異なる時間間隔で計算された 2 つの異なるサマリーを作成します。トレースは、サービス・リクエスト数およびリクエストの実行にかかった時間を計算します。
- この変数を使用して、ソートされたリクエストのリストから表示されるサービス数を指定します。リストは、消費された経過時間で降順にソートされます。
- デフォルト値は、5 です。

場所

<*install_dir*>/config/config.cfg

例

```
RequestAuditListDepth1=7
```

関連項目

- RequestAuditAdditionalVerboseFieldsList（6-644 ページ）
- RequestAuditIntervalSeconds1（6-645 ページ）
- RequestAuditIntervalSeconds2（6-646 ページ）
- RequestAuditListDepth2（6-648 ページ）
- RequestAuditSubClassificationField（6-649 ページ）

REQUESTAUDITLISTDEPTH2

タイプおよび使用方法

- 構成変数（4-3 ページ）
- デバッグ（5-20 ページ）

説明

リクエスト監査トレースに表示するサービス数を指定するために使用します。

- この構成変数は、長い監査期間にのみ適用されます。
- リクエスト監査トレースは、「System Audit Information」ページに表示されます。このトレースは、2 つの異なる時間間隔で計算された 2 つの異なるサマリーを作成します。トレースは、リクエスト数およびリクエストの実行にかかった時間を計算します。
- この変数を使用して、表示するサービス数を指定します。リストは、消費された経過時間で降順にソートされます。
- デフォルト値は、20 です。

場所

<*install_dir*>/config/config.cfg

例

```
RequestAuditListDepth2=25
```

関連項目

- RequestAuditAdditionalVerboseFieldsList（6-644 ページ）
- RequestAuditIntervalSeconds1（6-645 ページ）
- RequestAuditIntervalSeconds2（6-646 ページ）
- RequestAuditListDepth1（6-647 ページ）
- RequestAuditSubClassificationField（6-649 ページ）

REQUESTAUDITSUBCLASSIFICATIONFIELD

タイプおよび使用方法

- 構成変数（4-3 ページ）
- デバッグ（5-20 ページ）

説明

この関数を使用して、内部フィールドを指定します。このフィールドは通常、ドキュメントのメタデータ・フィールドで、この値をサービス名に付加することによって、リクエスト監査リスト中の名前を区別します。

- リクエスト監査トレースは、「System Audit Information」ページに表示されます。このトレースは、2 つの異なる時間間隔で計算された 2 つの異なるサマリーを作成します。トレースは、サービス・リクエスト数およびリクエストの実行にかかった時間を計算します。
- 場合によっては、サービス名では、異なるリクエスト・タイプの適切な内訳が作成されないことがあります。このエントリは、エントリのタイプの識別に役立ちます。

場所

<*install_dir*>/config/config.cfg

例

```
RequestAuditSubClassificationField=xDepartment
```

関連項目

- RequestAuditAdditionalVerboseFieldsList（6-644 ページ）
- RequestAuditIntervalSeconds1（6-645 ページ）
- RequestAuditIntervalSeconds2（6-646 ページ）
- RequestAuditListDepth1（6-647 ページ）
- RequestAuditListDepth2（6-648 ページ）

REQUIREDMSG

タイプおよび使用方法

- フィールド情報変数（4-26 ページ）
- フィールド表示（5-24 ページ）

説明

チェックイン時に必須メタデータ・フィールドの値がなかった場合に表示されるエラー・メッセージを指定します。

出力

エラー・メッセージを文字列で返します。

例

dDocTitle メタデータ・フィールドの必須フィールドのエラー・メッセージをローカライズされた文字列 *wwSpecifyTitle* として定義します。

```
<$fieldName = "dDocTitle", fieldCaption = lc("wwTitle"), isRequired = 1,
fieldType = "BigText", requiredMsg = lc("wwSpecifyTitle")$>
```

注意：詳細は、『Customizing Content Server』のローカライズされた文字列の解決に関する項を参照してください。

関連項目

– isRequired（6-435 ページ）

RESULTSTITLE

タイプおよび使用方法

- 値変数（4-34 ページ）
- ページ表示（5-30 ページ）
- 検索（5-34 ページ）

説明

検索結果ページのタイトルを定義します。

- この変数は、検索結果ページの名前付けおよびページの最上部の見出しを表示するために Web レイアウト・エディタで使用されます。

出力

ページ・タイトルを文字列で返します。

例

HDA エントリとして、検索結果ページに名前を付けます。

```
@Properties LocalData
ResultsTitle=Content Items
@end
```

スクリプトとして、定義された名前を返します。

```
<$if ResultsTitle$>ResultsTitle=<$url(ResultsTitle)$>
```

RMAADDDOCWHERECLAUSEFORSCREENING

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Record Manager および Retention Manager（5-32 ページ）

説明

Retention Manager 専用。

スクリーニング・ページまたは「Freeze Information」ページで、retadmin ロールを持つユーザーによる、（ACL で）アクセスを許可されていない凍結アイテムのスクリーニングを可能にします。

デフォルトは、FALSE です。

場所

<*install_dir*>/custom/RetentionManager/retentionmanager_environments.cfg

例

次は構成エントリの使用例です。

```
RmaAddDocWhereClauseForScreening=true
```

RMAHIDEEXTERNALFIELDSFROMCHECKINUPDATE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Record Manager および Retention Manager（5-32 ページ）

説明

TRUE に設定すると、「Checkin」ページおよび「Update」ページで外部フィールドが非表示になります。

デフォルトは、TRUE です。

場所

- <*install_dir*>/custom/RecordsManagement/records_management_environments.cfg
- <*install_dir*>/custom/RetentionManager/retentionmanager_environments.cfg

例

次は構成エントリの使用例です。

```
RmaHideExternalFieldsFromCheckInUpdate=false
```

RMAHIDEEXTERNALFIELDSFROMSEARCHINFO

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Record Manager および Retention Manager（5-32 ページ）

説明

TRUE に設定すると、「Search」ページおよび「Content Information」ページで外部フィールドが非表示になります。デフォルトは、TRUE です。

場所

- <*install_dir*>/custom/RecordsManagement/records_management_environments.cfg
- <*install_dir*>/custom/RetentionManager/retentionmanager_environments.cfg

例

次は構成エントリの使用例です。

```
RmaHideExternalFieldsFromSearchInfo=false
```

RMANOTIFYDISPREVIEWERANDCATAUTHOR

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Record Manager および Retention Manager（5-32 ページ）

説明

処理ルールによってイベントがトリガーされたときに通知を受信するユーザーを制御します。TRUE に設定すると、指定された通知レビューアおよび元のカテゴリの作成者がともにイベントの電子メール通知を受信します。FALSE に設定すると、カテゴリの作成者のみが電子メールを受信します。

デフォルトは、TRUE です。

場所

- <*install_dir*>/custom/RecordsManagement/records_management_environments.cfg
- <*install_dir*>/custom/RetentionManager/retentionmanager_environments.cfg

例

次は構成エントリの使用例です。

```
RmaNotifyDispReviewerAndCatAuthor=false
```

RMANOTIFYREVIEWERANDALTERNATEREVIEWER

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Record Manager および Retention Manager（5-32 ページ）

説明

代替ユーザーが、レビュー・アクションの実行および割り当てられた処理イベントを処理できるかどうかを制御します。TRUE に設定すると、元のユーザーおよび代替レビューアの両者がアクションに関する電子メール通知を受信します。FALSE に設定すると、代替レビューアのみが電子メールを受信します。

デフォルトは、TRUE です。

場所

- <*install_dir*>/custom/RecordsManagement/records_management_environments.cfg
- <*install_dir*>/custom/RetentionManager/retentionmanager_environments.cfg

例

次は構成エントリの使用例です。

```
RmaNotifyReviewerAndAlternateReviewer=false
```

ROLEPREFIX

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

コンテンツ・サーバーのロールにマップされる Active Directory または LDAP の接頭辞を定義します。

- これはセミコロンで区切られた LDAP スタイル文字列のリストで、これらの文字列は、Active Directory または LDAP グループ名の部分文字列です。グループがこれらの文字列のいずれかに一致する場合、深さの定義に従って、グループ定義の残りをコンテンツ・サーバー・ロールにマップできます。詳細な例は、「Active Directory Configuration」ページまたは「Add/Edit LDAP Provider」ページを参照してください。
- ロールを Active Directory または LDAP グループからマップするには、UseGroupFilter（6-842 ページ）設定が TRUE である必要があります。
- デフォルトは、空の文字列です。

場所

Active Directory セキュリティ

- 「Active Directory Configuration」ページ→「Role Prefix」
- <*install_dir*>/data/users/config/filter.hda

LDAP セキュリティ

- 「Add/Edit LDAP Provider」ページ→「Role Prefix」
- <*install_dir*>/data/providers/*provider_name*/provider.hda

例

次は構成エントリの使用例です。

```
RolePrefix=OU=Roles,OU=Stellent[1];
```

関連項目

– UseGroupFilter（6-842 ページ）
– AcctPrefix（6-7 ページ）

RPTDISPLAYMAPVALUE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ローカライゼーション（5-29 ページ）

説明

この関数は、内部使用のみです。

この関数は、内部キー・コードのローカライズされた文字列表現を返します。これらのキー・コードは、コンテンツ・アイテムのステータスおよび状態フラグの保存に使用されます。この関数は、内部状態の説明を表示するために「Content Information」ページおよびワークフロー・ページで使用されます。

パラメータ

パラメータを 2 つ取ります。

- キーが格納されている表の名前
- ローカライズの対象となるキー

RSADDFIELDS

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

ResultSet に新規フィールドを追加します。この関数は、フィールドがまだ存在しない場合にのみフィールドを追加します。setValue(...) でも新規フィールドを追加できますが、ResultSet が空ではなく、現在有効な行にある場合にのみ機能します。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、新規フィールド（列）が追加される ResultSet の名前です。
- 2 番目のパラメータは、追加するフィールド（追加される列名）のカンマ区切りのリストです。

出力

- 関数が成功した場合、TRUE を返します。
- 関数が失敗した場合、FALSE を返します。

例

```
<$rsAddFields(rsName, fieldsList)$>
```

RSADDROWCOUNTCOLUMN

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

この関数は、各行の行カウントの値を持つ新規フィールドを追加します。カウントは 0 から開始します。この関数は、XML データ・アイランド ResultSet（hcsp ファイル内）で rsSort(...) を使用する直前に使用して、元の行位置を保持するのに役立ちます。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、新しい行カウント列を追加するために変更される ResultSet の名前です。
- 2 番目のパラメータは、行カウントを保持するフィールドの名前です。

出力

- 関数が成功した場合、TRUE を返します。
- 関数が失敗した場合、FALSE を返します。

例

```
<$rsAddRowCountColumn(rsName, countFieldName)$>
```

RSAPPEND

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

この関数は、マージのテスト基準がない点を除いて、本質的に rsMerge に類似しています。rsNameSource ResultSet からのすべての行が rsNameTarget ResultSet に追加されます。さらに、rsNameTarget に存在しない rsNameSource 内のすべてのフィールドが rsNameTarget に追加されます。ResultSet の単純な結合を行う場合に役立ちます。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、マージによって変更される ResultSet です。
- 2 番目のパラメータは、ターゲットに追加される行を含む ResultSet です。

出力

- 関数が成功した場合、TRUE を返します。
- 関数が失敗した場合、FALSE を返します。

例

```
<$rsAppend(targetRsetName, sourceRsetName)$>
```

関連項目

– rsMerge（6-682 ページ）

RSAPPENDNEWROW

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

ResultSet の末尾に新しい行を追加します。

- 現在の行は、影響を受けません。

パラメータ

1 つのパラメータを取ります。

- パラメータは、新しい行を受け取る ResultSet の名前です。

出力

- 関数が成功した場合、TRUE を返します。
- 関数が失敗するか、ResultSet が空の場合、FALSE を返します。

例

```
<$rsAppendNewRow("SearchResults")$>
```

RSAPPENDROWVALUES

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

ResultSet に新しい行を追加するユーティリティ関数です。値のリストは、カンマで区切られた、複数の行に分割される値のリストです（',' -> '^'、'^' -> '#^'、'#' -> '##' のエスケープ・ルールを使用して各値をエンコード）。ResultSet 内のフィールド数が nfields であるとします。値のリストは、nfields の値をカウントして新しい行を作成した後、次の nfields の値を取得して後続の各行を作成することによって、複数の行に分割されます。値の数が nfields の倍数でない場合は、最後の行のフィールド値は空の文字列で埋められます。この関数は、Idoc スクリプトを使用して ResultSet を簡潔にハード・コードする場合に役立ちます。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、新しい行を取得する ResultSet の名前です。
- 2 番目のパラメータは、追加する新しい値です。

出力

- 関数が成功した場合、TRUE を返します。
- 関数が失敗した場合、FALSE を返します。

例

```
<$rowValues = "a1, b1, a2, b2"$>
<$rsCreateResultSet("ResultSet1", "ColumnA, ColumnB")$>
<$rsAppendRowvalues("ResultSet1", rowValues$>
<table border=2>
    <tr><td>ColumnA</td><td>ColumnB</td></tr>
    <$loop ResultSet1$>
    <tr><td><$ColumnA$></td><td><$ColumnB$></td>
</tr>
<$endloop$>
</table>
```

結果の HTML は次のようになります。

列 A	**列 B**
A1 A2	B1 B2

RSCOPYFILTERED

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

この関数は、選択された行のみを ResultSet からコピーして、別の ResultSet を作成します。名前が rsNameTarget である以前の ResultSet は置換されます。行の選択には、filterPattern のパターンに対するテストを使用して、各行の filterField の値をテストします。パターン・マッチは、Idoc スクリプトで使用する like 演算子と同じです。

パラメータ

パラメータを 4 つ取ります。

- 最初のパラメータは、コピーする行を提供する ResultSet です。
- 2 番目のパラメータは、コピーによって作成される新しい ResultSet です。
- 3 番目のパラメータは、テストされるフィールドの名前です。
- 4 番目のパラメータは、行をコピーするかどうかを確認するために適用するパターン・マッチです。

出力

- 関数が成功した場合、TRUE を返します。
- 関数が失敗した場合、FALSE を返します。

例

```
<$rsCopyFiltered(rsNameSource, rsNameTarget, filterField, filterPattern)$>
```

RSCREATERESULTSET

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

fieldList 内のフィールド・リストを使用して初期化された ResultSet を作成します。

- すべてのフィールドが String 型として設定されます。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、作成する ResultSet の名前です。
- 2 番目のパラメータは、カンマ区切りのフィールド・リストです。

出力

- 関数が成功した場合、TRUE を返します。
- 関数が失敗した場合、FALSE を返します。

例

```
<$rsCreateResultSet(rsName, fieldList)$>
```

RSDELETEROW

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

ResultSet の現在の行を削除します。現在の行は、次の行に進みます（または、ResultSet の最後の行が削除された場合、ResultSet の末尾の後ろを指します）。

パラメータ

1 つのパラメータを取ります。

- パラメータは、現在の行が削除される ResultSet の名前です。

出力

- 関数が成功した場合、TRUE を返します。
- 関数が失敗するか、ResultSet が空の場合、FALSE を返します。

例

```
<$rsDeleteRow("Folders")$>
```

RSDOCINFOROWALLOWDISCLOSURE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）
- 内部セキュリティ（5-36 ページ）

説明

現在のユーザーが、ResultSet の現在の行によって参照されたコンテンツ・アイテムの URL を表示できるかどうかをチェックします。

- この関数は、コンテンツ・アイテム問合せによって生成される ResultSet の URL を選択的に表示する場合に役立ちます。

パラメータ

1 つのパラメータを取ります。

- パラメータは、現在の行にコンテンツ・アイテムの情報が含まれる ResultSet の名前のみです。

出力

- ユーザーがコンテンツ・アイテムの URL を表示できる場合は、TRUE を返します。
- ユーザーが URL を表示できないか、ResultSet が空の場合は、FALSE を返します。

例

ユーザーが、*resultSet1* の現在行のコンテンツ・アイテムの URL を表示できるかどうかをチェックします。

```
<$rsDocInfoRowAllowDisclosure("resultSet1")$>
```

RSEXISTS

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

結果セットが存在するかどうかをチェックします。

パラメータ

1 つのパラメータを取ります。

- 結果セットの名前

出力

- 結果セットが見つかった場合は、TRUE を返します。
- 結果セットが存在しないか、空の場合は、FALSE を返します。

例

```
<$if rsExists("resultSet1")$>
   code to execute if TRUE
<$endif$>
```

RSFIELDBYINDEX

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

0 から開始して、指定された列索引のフィールド名を取得します。

パラメータ

パラメータを 2 つ取ります。

- ResultSet の名前
- 索引番号

例

```
<$rsFieldName = rsFieldByIndex("resultSet1", "2")$>
```

RSFIELDEXISTS

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

指定された結果セットに特定のフィールドが含まれるかどうかをチェックします。

パラメータ

パラメータを 2 つ取ります。

- 検索される結果セットの名前
- 見つけるフィールドの名前

出力

- フィールドが見つかった場合は、TRUE を返します。
- フィールドが存在しない場合は、FALSE を返します。

例

```
<$if rsFieldExists("resultSet1", "FieldName1")$>
   code to execute if TRUE
<$endif$>
```

RSFINDROWPRIMARY

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

ResultSet の最初の列で一致する値を検索します。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、最初の列が検索される ResultSet の名前です。
- 2 番目のパラメータは、検索する値です。

出力

- 指定された値が、指定された ResultSet の最初の列で見つかった場合は、TRUE を返します。
- 指定された値が見つからない場合は、FALSE を返します。

例

value1 と一致する値が見つかるまで、*resultSet1* の最初の列を検索します。

```
<$rsFindRowPrimary("resultSet1","value1")$>
```

RSFIRST

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

ResultSet の最初の行に移動します。

パラメータ

1 つのパラメータを取ります。

- パラメータは、ResultSet の名前のみです。

出力

- 関数が成功した場合、TRUE を返します。
- 関数が失敗するか、ResultSet が空の場合、FALSE を返します。

例

SearchResults ResultSet の最初の行に移動します。

```
<$exec rsFirst("SearchResults")$>
```

RSINSERTNEWROW

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

操作中の ResultSet の現在行の直前に行を挿入します。挿入後、新しい行が現在行になります。

パラメータ

1 つのパラメータを取ります。

- パラメータは、新しい行が挿入される ResultSet の名前のみです。

出力

- 関数が成功した場合、TRUE を返します。
- 関数が失敗した場合、FALSE を返します。

例

```
<$rsInsertNewRow("SearchResults")$>
```

RSISROWPRESENT

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

結果セットの現在の行のステータスをチェックします。

パラメータ

1 つのパラメータを取ります。

- 結果セットの名前

出力

- 結果セットの現在の行が有効な場合は、TRUE を返します。
- 有効な行でない場合は、FALSE を返します。

例

```
<$if rsIsRowPresent("resultSet1")$>
    code to execute if TRUE
<$endif$>
```

RSMAKEFROMLIST

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

オプション・リストから単一列の ResultSet を作成します。

パラメータ

2 つの必須パラメータと 1 つのオプションのパラメータを取ります。

- 最初のパラメータは、作成する ResultSet の名前です。
- 2 番目のパラメータは、既存のオプション・リストの名前です（たとえば、*docAccounts* または *xMyCustomField.options* など）。
- 3 番目のオプション・パラメータは、ResultSet の列名です。値を指定しない場合、列名のデフォルトは row です。

出力

指定されたオプション・リストの値が移入された、1 つの列を持つ ResultSet を作成します。

例

securityGroups および *docAuthors* オプション・リストから、*ListA* および *ListB* という 2 つの ResultSet を作成します。

```
<$lista = "securityGroups"$>
<$rsMakeFromList("ListA", lista)$>
<$rsMakeFromList("ListB", "docAuthors", "name")$>
<table border=2>
    <tr><td>Security Groups</td><td>Authors</td></tr>
    <tr>
        <td><$loop ListA$><$row$><br>
        <$endloop$></td>
        <td><$loop ListB$><$name$><br>
        <$endloop$></td>
    </tr>
</table>
```

ResultSet は、次のようになります。

```
@ResultSet ListA
1
row
Public
Secure
@end

@ResultSet ListB
1
name
hchang
pkelly
sysadmin
user1
@end
```

結果の HTML は次のようになります。

Security Groups	**Authors**
Public Secure	hchang pkelly sysadmin user1

関連項目

- rsMakeFromString（6-680 ページ）
- オプション・リスト（3-33 ページ）
- optList（6-572 ページ）

RSMAKEFROMSTRING

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

文字列から単一列の ResultSet を作成します。

パラメータ

2 つの必須パラメータと 1 つのオプションのパラメータを取ります。

- 最初のパラメータは、作成する ResultSet の名前です。
- 2 番目のパラメータは、解析される文字列のカンマ区切りのリスト（"a,b,c,d" など）、またはカンマで区切られた文字列を値として持つ変数です。
- 3 番目のオプション・パラメータは、ResultSet の列名です。値を指定しない場合、列名のデフォルトは row です。

出力

指定された文字列の値が移入された、1 つの列を持つ ResultSet が作成されます。

例

StringA および *StringB* という 2 つの ResultSet を作成します。

```
<$stringa = "a,b,c,d"$>
<$rsMakeFromString("StringA", stringa)$>
<$rsMakeFromString("StringB", "A,B,C,D", "name")$>
<table border=2>
    <tr><td>StringA</td><td>StringB</td></tr>
    <tr>
        <td><$loop StringA$><$row$><br>
        <$endloop$></td>
        <td><$loop StringB$><$name$><br>
        <$endloop$></td>
    </tr>
</table>
```

ResultSet は、次のようになります。

```
@ResultSet StringA
1
row
a
b
c
d
@end

@ResultSet StringB
1
name
A
B
C
D
@end
```

結果の HTML は次のようになります。

StringA	StringB
a	A
b	B
c	C
d	D

関連項目

– rsMakeFromList（6-677 ページ）

RSMERGE

タイプおよび使用方法

❖ グローバル関数（4-21 ページ）

❖ ResultSet（5-32 ページ）

説明

ResultSet の行を別の ResultSet にマージします。rsNameSource ResultSet の rsCommonField フィールドの値が、rsNameTarget の行の同フィールドの値と一致する場合、その行は置換されます。それ以外の場合は、rsNameSource からの行が追加されます。同じ値に一致する複数の行が rsNameTarget に存在する場合は、最初の行のみが置換され、その行は、一致する行を持つ rsNameSource の最後の行によって置換されます。rsNameTarget に存在しない rsNameSource ResultSet 内のフィールドは、新しいフィールドとして rsNameTarget に追加されます。複数一致の問題を回避するため、この関数は、rsCommonField フィールド内に一意の値のみを持つ ResultSet で実行すると最善の結果が得られます。

パラメータ

パラメータを 3 つ取ります。

❖ 最初のパラメータ（rsNameTarget）は、マージによって変更される ResultSet です。

❖ 2 番目のパラメータ（rsNameSource）は、ターゲットにマージされる行を含む ResultSet です。

❖ 3 番目のパラメータ（rsCommonField）は、マージのベースとして使用され、両方の結果セットに発生するフィールドです。

出力

❖ 関数が成功した場合、TRUE を返します。

❖ 関数が失敗した場合、FALSE を返します。

例

```
<$rsMerge(rsNameTarget, rsNameSource, commonField$>
```

関連項目

– rsAppend（6-662 ページ）

RSMERGEDELETE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

rsMergeReplaceOnly と同じですが、rsNameTarget ResultSet の一致する行は、置換されるのではなく、削除されます。これは、別の ResultSet ですでに発生している行を ResultSet から削除する場合に役立ちます。

パラメータ

パラメータを 3 つ取ります。

- 最初のパラメータ（rsNameTarget）は、マージによって変更される ResultSet です。
- 2 番目のパラメータ（rsNameSource）は、ターゲットにマージされる行を含む ResultSet です。
- 3 番目のパラメータ（rsCommonField）は、rsNameTarget から削除される行がどれかをテストするために使用され、両方の結果セットに発生するフィールドです。

出力

- 関数が成功した場合、TRUE を返します。
- 関数が失敗した場合、FALSE を返します。

例

```
<$rsMergeDelete(rsNameTarget, rsNameSource, rsCommonField)$>
```

RSMERGEREPLACEONLY

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

rsMerge に類似していますが、行の置換のみが行われます。rsNameTarget ResultSet に追加される行はありません。また、rsNameSource からの新規フィールドは、rsNameTarget に追加されません。

パラメータ

パラメータを 3 つ取ります。

- 最初のパラメータ（rsNameTarget）は、マージによって変更される ResultSet です。
- 2 番目のパラメータ（rsNameSource）は、ターゲットにマージされる行を含む ResultSet です。
- 3 番目のパラメータ（rsCommonField）は、マージのベースとして使用され、両方の結果セットに発生するフィールドです。

出力

- 関数が成功した場合、TRUE を返します。
- 関数が失敗した場合、FALSE を返します。

例

```
<$rsMergeReplaceOnly(rsNameTarget, rsNameSource, rsCommonField) $>
```

RSNEXT

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

ResultSet の次の行に移動します。

パラメータ

1 つのパラメータを取ります。

- パラメータは、ResultSet の名前のみです。

出力

- 関数が成功した場合、TRUE を返します。
- 関数が失敗するか、ResultSet が空の場合、FALSE を返します。

例

SearchResults ResultSet の次の行に移動します。

```
<$exec rsNext("SearchResults")$>
```

RSNUMFIELDS

タイプおよび使用方法

❖ グローバル関数（4-21 ページ）

❖ ResultSet（5-32 ページ）

説明

結果セットのフィールド数のカウントを提供します。これは、`rsFieldByIndex` と併用すると便利です。

パラメータ

1 つのパラメータを取ります。

❖ 結果セットの名前

出力

フィールド数を返します。

例

```
<$numFields = rsNumFields("resultSet1")$>
```

関連項目

– rsFieldByIndex（6-671 ページ）

RSNUMROWS

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

結果セットの行数のカウントを提供します。

パラメータ

1 つのパラメータを取ります。

- 結果セットの名前

出力

行数を返します。

例

```
<$numRows = rsNumRows("resultSet1")$>
```

RSREMOVE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

結果セットを削除します。

パラメータ

1 つのパラメータを取ります。

- 結果セットの名前

例

```
<$rsRemove("resultSet1")$>
```

RSRENAME

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

ResultSet の名前を変更します。

- 変数を使用して新しい ResultSet 名を定義する場合は、*rsFirst*、*rsNext* および *#isRowPresent* を使用して、新しい ResultSet 変数に対するループ処理を手動で実行する必要があります。次に例を示します。

```
<$rsNewName="MySearchResults"$>

<$rsRename("SearchResults", rsNewName)$>

<$rsFirst(rsNewName)$>
<$loopwhile getValue(rsNewName, "#isRowPresent")$>
    <!--output code-->
    <$rsNext(rsNewName)$>
<$endloop$>
```

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、既存の ResultSet の名前です。
- 2 番目のパラメータは、新しい ResultSet の名前です。ResultSet がすでに存在する場合、コールは失敗します。

出力

- 関数が成功した場合、TRUE を返します。
- 関数が失敗した場合、FALSE を返します。

例

SearchResults ResultSet の名前を MySearchResults に変更します。

```
<$rsRename("SearchResults", "MySearchResults")$>
```

RSRENAMEFIELD

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

ResultSet 内のフィールド名を変更します。

パラメータ

パラメータを 3 つ取ります。

- 最初のパラメータは、名前変更されるフィールドを持つ ResultSet の名前です。
- 2 番目のパラメータは、現在のフィールド名です。
- 3 番目のパラメータは、新しいフィールド名です。

出力

- 関数が成功した場合、TRUE を返します。
- 関数が失敗した場合、FALSE を返します。

例

warehouse ResultSet で、invoice フィールドを outlays に名前変更します。

```
<$exec rsRenameField("warehouse", "invoice", "outlays")$>
```

RSSETROW

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

ResultSet 内の指定された行に移動します。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、ResultSet の名前です。
- 2 番目のパラメータは、行の番号です。

出力

- 関数が成功した場合、TRUE を返します。
- 関数が失敗するか、ResultSet が空の場合、FALSE を返します。

例

SearchResults ResultSet の 10 行目に移動します。

```
<$exec rsSetRow("SearchResults",10)$>
```

RSSORT

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

特定の列で ResultSet をソートします。

パラメータ

パラメータを 4 つ取ります。次に例を示します。

```
rsSort(rsName,sortCol,sortType,sortOrder)
```

パラメータ	説明
rsName	ResultSet の名前
sortCol	ソートに使用する列の名前
sortType	ソートのタイプ（デフォルトは *int*） **string:** アルファベット順にソート **int:** 数値でソート **date:** 日付でソート
sortOrder	ソート順序（デフォルトは *asc*） **asc:** 昇順 **desc:** 降順

出力

- 関数が成功した場合、TRUE を返します。
- 関数が失敗するか、ResultSet が空の場合、FALSE を返します。

例

SearchResults ResultSet を *Content ID* の降順にソートします。

```
<$rsSort("SearchResults","dDocName","string","desc")$>
```

RSSORTTREE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

XML データ構造（ノード、親ノードおよび深度の属性を持つ）を表す ResultSet をソートします。

パラメータ

パラメータを 7 つ取ります。次に例を示します。

```
rsSortTree(rsName,itemIdCol,parentIdCol,nestLevelCol,sortCol,sortType,sortOrder)
```

パラメータ	説明
rsName	ResultSet の名前
itemIdCol	各ノードの一意の ID を含む列の名前
parentIdCol	存在する場合、親ノードの ID を含む列の名前
nestLevelCol	各ノードのネスト・レベル（深度）を含む列の名前
sortCol	ソートに使用する列の名前
sortType	ソートのタイプ（デフォルトは *int*） **string:** アルファベット順にソート **int:** 数値でソート **date:** 日付でソート
sortOrder	ソート順序（デフォルトは *asc*） **asc:** 昇順 **desc:** 降順

出力

- ❖ 関数が成功した場合、TRUE を返します。
- ❖ 関数が失敗するか、ResultSet が空の場合、FALSE を返します。

例

discussionPosts ResultSet を *itemNumber* 列の昇順にソートします。

```
<$rsSortTree("discussionPosts", "discussionPosts!itemNumber",
"discussionPosts!parentNumber", "dpItemNestLevel", "discussionPosts!itemNumber",
"int", "asc")$>
```

SAFEDIR

タイプおよび使用方法

- その他の読取り専用変数（4-33 ページ）
- デバッグ（5-20 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

Content Server Analyzer ユーティリティによって、追加のコンテンツ・サーバー・ファイルが移動される場所を取得します。

- デフォルトは、IdcAnalyzeLogDir（6-330 ページ）設定で指定されたログ・ディレクトリ内のサブディレクトリです。
- この変数は設定可能ではありません。

場所

N/A

例

スクリプトとして、構成設定の値を返します。

```
<$SafeDir$>
```

関連項目

– IdcAnalyzeLogDir（6-330 ページ）

SCHEMAALLOWRECURSIVEFILEMOVE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- スキーマ（5-33 ページ）

説明

TRUE に設定すると、UNIX システムでファイルを個別にスキーマの公開場所から本番の場所に移動できます。これにより、スキーマ・ディレクトリから本番ディレクトリへのスワッピングが可能になります。

場所

<*install_dir*>/config/config.cfg

例

```
SchemaAllowRecursiveFileMove=true
```

SCHEMABUILDDIGESTFILE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- スキーマ（5-33 ページ）

説明

TRUE に設定すると、公開されたすべてのファイルのダイジェストを含むファイルが作成されます。これを TRUE に設定した場合、スキーマの公開のための所要時間が増加します。

場所

<*install_dir*>/config/config.cfg

例

```
SchemaBuildDigestFile=true
```

関連項目

– SchemaPublishCheckInterval（6-698 ページ）

SCHEMAPUBLISHCHECKINTERVAL

タイプおよび使用方法

- 構成変数（4-3 ページ）
- スキーマ（5-33 ページ）

説明

非サブジェクト・ドリブンのスキーマ公開リクエストをチェックする間隔を設定します。

- data/schema/publishlock/publish.dat ファイルにアクセスした場合にトリガーされます。
- 値を秒数で指定します。
- デフォルトは、120 秒（2 分）です。
- スキーマ管理の表を変更し、<*install_dir*>/data/schema/publishlock/publish.dat ファイルにアクセスする外部プログラムが存在する場合は、この設定を変更します。
- 開発用のサーバーでは、迅速な公開を確保するために小さい値を使用します（たとえば、`SchemaPublishCheckInterval=2`）。
- 本番サーバーでデフォルト値を小さくすると、パフォーマンスの問題が生じる場合があります。

場所

<*install_dir*>/config/config.cfg

例

この間隔を 10 分に設定するには、次を使用します。

```
SchemaPublishCheckInterval=600
```

SCHEMAPUBLISHINTERVAL

タイプおよび使用方法

- 構成変数（4-3 ページ）
- スキーマ（5-33 ページ）

説明

正当な理由のない（イベントによって発生する以外の）スキーマ操作の間隔。

- この値は秒数で指定します。
- デフォルトは、14400（4 時間）です。
- 通常、SchemaPublisher は、この期間内に少なくとも 1 回公開します。
- 開発用のサーバーでは、迅速な公開を確保するために小さい値を使用します（たとえば、`SchemaPublishInterval=30`）。
- 本番サーバーでデフォルト値を小さくすると、パフォーマンスの問題が生じる場合があります。

場所

<*install_dir*>/config/config.cfg

例

この間隔を 1 週間に設定するには、次を使用します。

```
SchemaPublishInterval=604800
```

SCHEMAPUBLISHMAXIMUMINTERVAL

タイプおよび使用方法

- 構成変数（4-3 ページ）
- スキーマ（5-33 ページ）

説明

自動スキーマ公開サイクル間の最大時間。

- この値は秒数で指定します。
- デフォルトは、86400（1 日）です。
- 開発用のサーバーでは、迅速な公開を確保するために小さい値に設定します（たとえば、`SchemaPublishMaximumInterval=30`）。
- 本番サーバーでデフォルト値を小さくすると、パフォーマンスの問題が生じる場合があります。

場所

<*install_dir*>/config/config.cfg

例

この値を 4 時間として指定するには、次を使用します。

```
SchemaPublishMaximumInterval=14400
```

SCHEMAPUBLISHMINIMUMINTERVAL

タイプおよび使用方法

- 構成変数（4-3 ページ）
- スキーマ（5-33 ページ）

説明

自動スキーマ公開サイクル間の最小時間。

- この値は秒数で指定します。
- デフォルトは、4 秒です。

場所

<*install_dir*>/config/config.cfg

例

```
SchemaPublishMinimumInterval=8
```

SCHEMAPUBLISHRENAMERETRYINITIALINTERVAL

タイプおよび使用方法

- 構成変数（4-3 ページ）
- スキーマ（5-33 ページ）

説明

名前変更の操作を再試行するまでの遅延を指定します。

- 名前変更の操作には、*schema* ディレクトリから *schema.old* へ、または *schema.work* ディレクトリから *schema* への名前変更が含まれます。
- 再試行の失敗後は、遅延値が 2 倍になります。
- `SchemaPublishRenameRetryIterations` に到達すると、公開操作は失敗します。
- 値をミリ秒で指定します。
- デフォルト値は、100 ミリ秒です。
- これは Windows システムの動作にのみ影響を与えます。

場所

<*install_dir*>/config/config.cfg

例

```
SchemaPublishRenameRetryInitialInterval=50
```

関連項目

– SchemaPublishRenameRetryIterations（6-703 ページ）

SCHEMAPUBLISHRENAMERETRYITERATIONS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- スキーマ（5-33 ページ）

説明

名前変更の操作を試行する回数を指定します。

- これは Windows システムの動作にのみ影響を与えます。
- この操作には、*schema* ディレクトリから *schema.old* へ、または *schema.work* ディレクトリから *schema* への名前変更が含まれます。

場所

<*install_dir*>/config/config.cfg

例

```
SchemaPublishRenameRetryIterations=6
```

関連項目

– SchemaPublishRenameRetryInitialInterval（6-702 ページ）

SCHEMAPUBLISHRETRYINTERVAL

タイプおよび使用方法

- 構成変数（4-3 ページ）
- スキーマ（5-33 ページ）

説明

SchemaPublisher によるスキーマの公開の失敗後に設ける遅延時間を指定します。

- この遅延は、SchemaPublisher による開始と失敗の継続を防止します。
- 再試行の失敗後、この値は 2 倍になりますが、`SchemaPublishRetryIntervalMax` の値を超過することはありません。
- 公開に成功すると、値は元の値にリセットされます。
- デフォルト値は、3 秒です。

場所

<*install_dir*>/config/config.cfg

例

```
SchemaPublishRetryInterval=4
```

関連項目

– SchemaPublishRetryIntervalMax（6-705 ページ）

SCHEMAPUBLISHRETRYINTERVALMAX

タイプおよび使用方法

❖ 構成変数（4-3 ページ）

❖ スキーマ（5-33 ページ）

説明

SchemaPublishRetryInterval の最大時間を指定します。

❖ この値は秒数で指定します。

❖ デフォルト値は、300（5 分）です。

場所

<*install_dir*>/config/config.cfg

例

```
SchemaPublishRetryIntervalMax=120
```

関連項目

– SchemaPublishRetryInterval（6-704 ページ）

SCHEMAVERIFYPUBLISHING

タイプおよび使用方法

- 構成変数（4-3 ページ）
- スキーマ（5-33 ページ）

説明

TRUE に設定すると、最後の名前変更を行う前に、ダイジェストを使用して SchemaPublisher が作成するファイルを検証します。

- `SchemaVerifyPublishing=TRUE` に設定した場合、スキーマの公開のための所要時間が増加します。
- 本番用のサーバーでは、TRUE に設定します。

場所

<*install_dir*>/config/config.cfg

例

```
SchemaVerifyPublishing=true
```

関連項目

– SchemaBuildDigestFile（6-697 ページ）
– SchemaPublishCheckInterval（6-698 ページ）

SCHEMAWIN32RENAMEDELAYINTERVAL

タイプおよび使用方法

- 構成変数（4-3 ページ）
- スキーマ（5-33 ページ）

説明

ファイルをクローズして名前変更を試みる前の遅延サイズを指定します。

- これは Windows システムの動作にのみ影響を与えます。
- この数値は、可能なかぎり小さい値にチューニングします。
- 値はミリ秒で指定します。
- デフォルト値は、10 ミリ秒です。

場所

<*install_dir*>/config/config.cfg

例

```
SchemaWin32RenameDelayInterval=4
```

SCRIPT_NAME

タイプおよび使用方法

- Web サーバー変数（4-36 ページ）
- Web サーバー（5-42 ページ）

説明

CGI リンク・ファイルの相対パスを取得します。

- CGI リンク・ファイルは、Web サーバー・セキュリティ・フィルタの実行可能ファイルであり、オペレーティング・システムおよび Web サーバーごとに異なるファイルです。たとえば、Solaris の場合、ファイル名は *nph-idc_cgi.exe* です。

出力

文字列として相対パスを返します。

例

次はページまたはログに出力される情報の例です。

```
SCRIPT_NAME=/idcplg/idc_cgi_isapi-<instance>.dll
```

次は Idoc スクリプト文または評価の一部としての使用例です。

```
<$if SCRIPT_NAME$>
<!--statement-->
```

SCRIPTDEBUGTRACE

タイプおよび使用方法

- 設定可能変数（4-34 ページ）
- デバッグ（5-20 ページ）

説明

すべてのインクルードおよび Idoc スクリプト eval（6-205 ページ）関数のコールのトレースを有効にします。

- *eval* 関数の内容および動的に割り当てられたインクルードもトレースの一部として表示されます。トレースは、*include* または *eval* コールのネスト・レベルごとに 1 つの + 文字でインデントされます。トレースには、エラー・メッセージ（ネストされた場所の情報なし）および Idoc スクリプト関数 trace（6-814 ページ）のコールの出力も表示されます。
- 構成ファイルでの設定またはサービス・コールのパラメータとして使用できます。

出力

TRUE に設定すると、すべてのインクルードおよび Idoc スクリプト関数 *eval* のコールをトレースし、トレースの結果がページに表示されます。

例

スクリプトとして、構成設定の値を返します。

```
<$ScriptDebugTrace$>
```

コンテンツ・サーバーの URL に含めると、すべてのデバッグ・トレース情報が「New Checkin」ページの下部に追加されます。

```
http://<home>/idcplg/idc_cgi_isapi-<instance>.dll?IdcService=
CHECKIN_NEW_FORM&ScriptDebugTrace=true
```

関連項目

- eval（6-205 ページ）
- ScriptErrorTrace（6-710 ページ）
- setResourceInclude（6-737 ページ）
- trace（6-814 ページ）

SCRIPTERRORTRACE

タイプおよび使用方法

❖ 設定可能変数（4-34 ページ）

❖ デバッグ（5-20 ページ）

説明

スクリプト・エラーのトレースを有効にします。

❖ 構成ファイルでの設定またはサービス・コールのパラメータとして使用できます。

出力

TRUE に設定すると、スクリプトのエラー情報が、表示されたページの下部に追加されます。エラーが検出されなかった場合は、「No Errors」が表示されます。

例

スクリプトとして、構成設定の値を返します。

```
<$ScriptErrorTrace$>
```

コンテンツ・サーバーの URL に含めると、エラー・トレース情報が「New Checkin」ページの下部に追加されます。

```
http://<home>/idcplg/idc_cgi_isapi-<instance>.dll?IdcService=
CHECKIN_NEW_FORM&ScriptErrorTrace=true
```

関連項目

– ScriptDebugTrace（6-709 ページ）

– setResourceInclude（6-737 ページ）

– trace（6-814 ページ）

SEARCHCACHECLEANERINTERVAL

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）

説明

検索キャッシュのクリーンアップの試行間隔を秒数で設定します。

- デフォルトは、120（2 分）です。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
SearchCacheCleanerInterval=60
```

SEARCHCACHEHARDLIMIT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）

説明

検索リクエストに厳格な制限が含まれる場合、検索キャッシュに含める最大結果数を定義します。

- 最後の行が SearchCacheSoftLimit より大きいが、SearchCacheHardLimit 以下である場合、検索キャッシュには、1 ～ SearchCacheHardLimit のすべての行が含まれます。
- 最後の行が SearchCacheHardLimit より大きい場合、検索はキャッシュされません。
- 通常、この設定は、1 ページ当たりに表示される最大検索結果数に一致します。6-527 ページの「MaxResults」を参照してください。
- デフォルトは、200 です。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
SearchCacheHardLimit=300
```

関連項目

– MaxResults（6-527 ページ）
– SearchCacheSoftLimit（6-714 ページ）

SEARCHCACHEMAXITEMAGE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）

説明

検索キャッシュ内のアイテムの最長保存期間（分）を設定します。

- デフォルトは、60（1 時間）です。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
SearchCacheMaxItemAge=30
```

SEARCHCACHESOFTLIMIT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）

説明

検索リクエストに弱い（ソフト）制限が含まれる場合、検索キャッシュに含める最大結果数を定義します。

- 検索の最終行が SearchCacheSoftLimit 以下の場合、検索キャッシュには、1 ～ SearchCacheSoftLimit の行が含まれます。
- 最終行が SearchCacheSoftLimit より大きいが、SearchCacheHardLimit 以下である場合、検索キャッシュには、1 ～ SearchCacheHardLimit のすべての行が含まれます。
- 最後の行が SearchCacheHardLimit より大きい場合、検索はキャッシュされません。
- 通常、この設定は、1 ページ当たりに表示されるデフォルトの検索結果数に一致します。
- デフォルトは、25 です。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
SearchCacheSoftLimit=50
```

関連項目

– SearchCacheHardLimit（6-712 ページ）

SEARCHCACHETRACE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）
- デバッグ（5-20 ページ）

説明

コンテンツ・サーバー・コンソールに、検索キャッシュのクリーニング・アルゴリズムに関するトレース情報をダンプします。

- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
SearchCacheTrace=true
```

注意：この変数は、Content Server 7.0 リリース以降では不要になりました。この機能は、「Admin Applets」の「System Audit Information」ページで利用できるようになりました。

SEARCHCONNECTIONWAITTIMEOUT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）

説明

検索の接続待機時間のタイムアウトをミリ秒で設定します。

- これは、検索が拒否されるまでに検索リクエストがキュー内で待機する最大時間です。
- デフォルトは、60000（1 分）です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
SearchConnectionWaitTimeout=120000
```

SEARCHDEBUGLEVEL

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）
- デバッグ（5-20 ページ）

説明

インデクサのログ・ファイル・エントリの詳細レベルを定義します。

- ログ・ファイルは、*/weblayout/groups/secure/logs/* にあります。
- 次にデバッグ・レベルのリストを示します。

レベル	説明
none	ログを生成しない。デフォルト。
verbose	アクセスした各ファイルの情報を表示。索引付け済、無視または失敗を示し、完全なレポートを生成。
debug	機能レベルでの追加情報を生成。
trace	実行される各アクティビティの情報をログ。
all	デバッグ、トレースおよび冗長な情報を含む完全なレポートを生成。

技術ヒント：デバッグ・レベルを *none* 以外に設定した場合、各索引サイクルのログ・ファイルが生成されます。ログ・ファイル数を減らすには、定期的にシステム・メンテナンスを実行する必要があります。

場所

- 「Repository Manager」→「Indexer」タブ→「Configure」→「Indexer Debug Level」
- <*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
SearchDebugLevel=all
```

SEARCHDIR

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ディレクトリおよびパス（5-22 ページ）
- 検索（5-34 ページ）

説明

search ディレクトリのパスを定義します。

- デフォルト値はありません。

場所

<*install_dir*>/bin/intradoc.cfg

例

次は構成エントリの使用例です。

```
SearchDir=f:/search/
```

SEARCHINDEXERENGINENAME

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）

説明

この変数は、使用する検索エンジンの名前を指定します。次の値がサポートされます。

- DATABASE
- DATABASEFULLTEXT
- VERITY

デフォルトは VERITY です。DATABASE オプションに切り替える場合、別の検索索引を使用するために追加の手順を実行する必要はありません。

各検索エンジンは、問合せ文字列に独自の書式を使用します。デフォルトでは、汎用問合せ構文が使用されますが、データベース検索には SQL 構文が必要です。

Content Server 8.0 以上で Verity を使用する場合は、値をすべて大文字にします。

場所

<*install_dir*>/config/config.cfg

例

```
SearchIndexerEngineName=DATABASE
```

SEARCHQUERYFORMAT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）

説明

サポートされた検索エンジンとは無関係に認識される問合せ構文を使用して検索問合せページを取得します。

- 動的に使用する場合、URL に次のように追加されます。
  `SearchQueryFormat=Universal`
- 動的に使用する場合、汎用検索構文は、1 つの検索セッション全体を通じて維持されます。構文は、問合せページ、結果ページ、ナビゲーション・リンクおよびページ内での検索や問合せの保存などのアクション中も維持されます。ユーザーがメニュー項目をクリックして検索問合せページを再度ロードするまで維持されます。
- これを Content Server の config.cfg ファイルに追加して、汎用検索問合せページを常に使用することもできます。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
SearchQueryFormat=Universal
```

関連項目

– UniversalSearchAllowFullText（6-825 ページ）

SEARCHQUERYTRACE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）
- デバッグ（5-20 ページ）

説明

検索の問合せに関するトレース情報をコンテンツ・サーバー・コンソールにダンプします。

- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
SearchQueryTrace=true
```

注意：この変数は、Content Server 7.0 リリース以降では不要になりました。この機能は、「Admin Applets」の「System Audit Information」ページで利用できるようになりました。

SECONDARYCONVERSIONSTEPNAME

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Inbound Refinery（5-11 ページ）

説明

ネイティブ・ステップの変換ステップが失敗した場合に Inbound Refinery が使用するバックアップ用の変換ステップの名前を定義します。

- デフォルトは、OIXConverter（Outside In）です。
- AllowSecondaryConversion（6-37 ページ）を TRUE に設定する必要があります。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
SecondaryConversionStepName=OIXConverter
```

関連項目

– AllowSecondaryConversion（6-37 ページ）

SELECTACCESSMEMBERALLOWMANUALENTRY

タイプおよび使用方法

- Collaboration Manager の構成変数（4-4 ページ）
- Collaboration Manager（5-5 ページ）

説明

Mozilla を使用するときにこの変数を有効にすると、ユーザーをコラボレーションに追加するとき、検索するユーザーの名前を入力できます。この機能は、Internet Explorer または Netscape を使用しているときにデフォルトで有効化されます。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
SelectAccessMemberAllowManualEntry=true
```

関連項目

– SelectAccessMemberResultLimit（6-724 ページ）

SELECTACCESSMEMBERRESULTLIMIT

タイプおよび使用方法

- Collaboration Manager の構成変数（4-4 ページ）
- Collaboration Manager（5-5 ページ）

説明

この変数は、ユーザーをコラボレーション・プロジェクトに追加する際に表示される名前の数を制限します。デフォルトは、200 です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
SelectAccessMemberResultLimit=400
```

関連項目

– SelectAccessMemberAllowManualEntry（6-723 ページ）

SELFREGISTEREDACCOUNTS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 内部セキュリティ（5-36 ページ）

説明

自己登録ユーザーに与えられるデフォルトのアカウントおよび権限を定義します。

- これはカンマ区切りのリストです。
- アカウント名の後に、各アカウントの権限をカッコ内に指定する必要があります。
- #none エントリの場合は、アカウントが割り当てられていないコンテンツ・アイテムに対する権限が付与されます。#all エントリの場合は、すべてのアカウントに対する権限が付与されます。
- デフォルト値はありません。
- アカウントのリストを文字列で返します。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
SelfRegisteredAccounts=#none(RWDA),USERS/<$NewUser$>,BOS(R)
```

スクリプトとして、定義されたアカウント情報を文字列で返します。

```
<$SelfRegisteredAccounts$>
```

関連項目

- UseSelfRegistration（6-879 ページ）
- SelfRegisteredAccounts（6-725 ページ）

SELFREGISTEREDROLES

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 内部セキュリティ（5-36 ページ）

説明

自己登録ユーザーに与えられるデフォルトのロールを定義します。

- これはカンマ区切りのリストです。
- デフォルト値はありません。
- ロールを文字列で返します。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
SelfRegisteredRoles=guest,salesRole
```

スクリプトとして、定義されたロールを文字列で返します。

```
<$SelfRegisteredRoles$>
```

関連項目

- UseSelfRegistration（6-879 ページ）
- SelfRegisteredRoles（6-726 ページ）

SERVER_NAME

タイプおよび使用方法

- Web サーバー変数（4-36 ページ）
- Web サーバー（5-42 ページ）

説明

自己参照 URL に表示されるとおりに、Web サーバーのホスト名、DNS 別名または IP アドレスを取得します。

- この変数は、特定のリクエストではなく、すべてのリクエストに設定されます。

出力

サーバー情報を文字列で返します。

例

次はページまたはログに出力される情報の例です。

```
SERVER_NAME=centralserver
```

次は Idoc スクリプト文または評価の一部としての使用例です。

```
<$if SERVER_NAME$>
<!--statement-->
```

SERVER_PORT

タイプおよび使用方法

❖ Web サーバー変数（4-36 ページ）

❖ Web サーバー（5-42 ページ）

説明

リクエストの送信先の Web サーバー・ポート番号を取得します。

❖ この変数は、現在のゲートウェイ・プログラム・リクエストに固有のものです。

出力

文字列としてポート番号を返します。

例

次はページまたはログに出力される情報の例です。

```
SERVER_PORT=80
```

次は Idoc スクリプト文または評価の一部としての使用例です。

```
<$if SERVER_PORT$>
<!--statement-->
```

SERVER_PROTOCOL

タイプおよび使用方法

- Web サーバー変数（4-36 ページ）
- Web サーバー（5-42 ページ）

説明

受信リクエストのプロトコルおよびリビジョンを返します。

- この変数は、現在のゲートウェイ・プログラム・リクエストに固有のものです。

出力

`protocol/revision` の書式で文字列を返します。

例

次はログに出力される情報の例です。

```
SERVER_PROTOCOL=HTTP/1.0
```

次は Idoc スクリプト文または評価の一部としての使用例です。

```
<$if SERVER_PROTOCOL$>
<!--statement-->
```

SERVER_SOFTWARE

タイプおよび使用方法

- Web サーバー変数（4-36 ページ）
- Web サーバー（5-42 ページ）

説明

重要：この設定は、バージョン 7.0 以降では不要です。Web サーバー・フィルタで、この情報は送信されなくなりました。

リクエストに応答している Web サーバー・ソフトウェアの名前とバージョンを返します。

- これはゲートウェイを実行しているサーバーでもあります。
- この変数は、特定のリクエストではなく、すべてのリクエストに設定されます。

出力

name/version の書式で文字列を返します。

例

次はページまたはログに出力される情報の例です。

```
SERVER_SOFTWARE=Microsoft-IIS/4.0
```

次は Idoc スクリプト文または評価の一部としての使用例です。

```
<$if SERVER_SOFTWARE$>
<!--statement-->
```

SETCONTENTTYPE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

ブラウザで適切にレンダリングされるように、返されたページのコンテンツ・タイプを設定します。たとえば、"text/plain"、"application/excel" または "text/xml" のコンテンツ・タイプ（ファイル形式のタイプまたは MIME タイプ）があります。

パラメータ

1 つのパラメータを取ります。

- パラメータは、コンテンツ・タイプのみです。

出力

- 関数が成功した場合、TRUE を返します。
- 関数が失敗した場合、FALSE を返します。

例

```
<$setContentType("text/plain")$>
```

SETCOOKIE

タイプおよび使用方法

- 設定可能変数（4-34 ページ）
- Web サーバー（5-42 ページ）

説明

ブラウザに Cookie 値を設定します。

- セッションの情報や匿名ユーザーに関する情報を保存するために使用されます。
- 有効期限を渡して、指定した時間に Cookie を期限切れにすることもできます。

重要 : セキュア・データを Cookie に保存しないでください。悪質なユーザーによって、サイトに Cookie が設定される可能性があるため、Cookie の使用を制限する必要があります。

パラメータ

この関数は、3 つのパラメータを取ることができます。

- 最初のパラメータは、Cookie の名前です。
- 2 番目のパラメータは、設定する値です。
- 3 番目のオプションのパラメータは、Cookie の有効期限を指定する有効期限インジケータです。

例

次の例では、ユーザーのブラウザに Cookie `myCounter` を設定し、ユーザーがページにアクセスしたときに値を増加します。Cookie は 1 日で期限切れになります。

```
<$myCounter=getCookie("myCounter")$>
<$if not myCounter$>
    <$myCounter=1$>
<$endif$>
Num times loaded: <$myCounter$>
<$setCookie("myCounter", myCounter+1, dateCurrent(1))$>
```

関連項目

– getCookie（6-260 ページ）

SETEXPIRES

タイプおよび使用方法

❖ グローバル関数（4-21 ページ）

❖ Idoc スクリプト（5-27 ページ）

説明

この関数を使用して、ページの有効期限の絶対時間を設定できます。これは、頻繁にリクエストされる存続期間の長いページでの使用に最も適しています。クライアントとサーバー間での時間の同期に問題があるため、特に存続期間の短いキャッシュでは、完全に信頼できるものではありません。データは、受け入れ可能な HTTP 日付書式のいずれかに従う必要があります。

❖ 詳細は、W3 プロトコルを参照してください。
http://www.w3.org/Protocols/rfc2616/rfc2616-sec3.html#sec3.3

パラメータ

1 つのパラメータを取ります。

❖ パラメータは、HTTP に必要な方法で書式設定された絶対日付のみです。この日付後、ページをリフレッシュする必要があります。

出力

❖ 関数が成功した場合、TRUE を返します。

❖ 関数が失敗した場合、FALSE を返します。

例

指定された絶対時間にページが期限切れになります。すべてのタイムスタンプは、GMT タイムゾーンで指定する必要があります。

```
<$setExpires("Sat, 02 Aug 2003 24:00:00 GMT")$>
```

他の IdocScript 関数を使用して、有効期限を現在の日付から 14 日後に設定する方法は次のとおりです。

```
<$date=formatDateWithPattern(dateCurrent(14), "EEE, dd MMM yyyy")$>
<$setExpires(date & " 24:00:00 GMT"))$>
```

関連項目

– setHttpHeader（6-734 ページ）

SETHTTPHEADER

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- Idoc スクリプト（5-27 ページ）

説明

この関数を使用して、任意の HTTP ヘッダーを設定できます。値としては、これらに限定されませんが、Cache-Control、Content-Type、Last-Modified、またはその他の許容される HTTP ヘッダーがあります。

- 詳細は、W3 プロトコルを参照してください。
  `http://www.w3.org/Protocols/rfc2616/rfc2616-sec3.html#sec3.3`

パラメータ

パラメータを 2 つ取ります。次に例を示します。

```
setHttpHeader(headerName, headerValue)
```

パラメータ	説明
headerName	有効な HTTP ヘッダー名
headerValue	ヘッダーの値

出力

- 関数が成功した場合、TRUE を返します。
- 関数が失敗した場合、FALSE を返します。

例

渡された XSLT を使用してブラウザでレンダリングできるように、コンテンツ・タイプを XML に設定します。

```
<$setHttpHeader("Content-Type", "text/xml")$>
```

関連項目

– setMaxAge（6-735 ページ）

SETMAXAGE

タイプおよび使用方法

❖ グローバル関数（4-21 ページ）

❖ Idoc スクリプト（5-27 ページ）

説明

この関数を使用して、ページの有効期限の絶対時間を設定できます。これは、頻繁にリクエストされる存続期間の長いページでの使用に最も適しています。クライアントとサーバー間での時間の同期に問題があるため、特に存続期間の短いキャッシュでは、完全に信頼できるものではありません。データは、受け入れ可能な HTTP 日付書式のいずれかに従う必要があります。

❖ 詳細は、W3 プロトコルを参照してください。
`http://www.w3.org/Protocols/rfc2616/rfc2616-sec3.html#sec3.3`

パラメータ

1 つのパラメータを取ります。

❖ パラメータは、このページの存続期間（秒数）のみです。

出力

❖ 関数が成功した場合、TRUE を返します。

❖ 関数が失敗した場合、FALSE を返します。

例

ページのコンテンツは即時に期限切れになります。

```
<$setMaxAge(0)$>
```

このページのキャッシュの存続期間を 10 分に設定し、この時間が経過したら即時にページのリフレッシュを絶対に試行する必要があることをブラウザに指定します。

```
<$setMaxAge(600)$>
<$setHttpHeader("Cache-Control", "must-revalidate")$>
```

関連項目

– setHttpHeader（6-734 ページ）

SETPDFVERSION

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

PDF Converter によって生成される PDF ファイルのバージョンを指定できるようになります。

- TRUE に設定すると、PdfVersion（6-583 ページ）設定の値に従って、PDF Converter が PDF ファイルを生成します。
- FALSE に設定すると、PDF Converter によって、PDF バージョン 1.4（Acrobat 5.0 準拠）ファイルが生成されます。
- デフォルトは、FALSE です。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
SetPdfVersion=true
```

関連項目

– PdfVersion（6-583 ページ）

SETRESOURCEINCLUDE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- Idoc スクリプト（5-27 ページ）

説明

インクルードを定義します。

- この関数により、動的に作成されたスクリプトをインクルードに割り当てることができます（同様に、eval（6-205 ページ）関数により、このようなスクリプトを解析および評価できます）。
- 指定したインクルードがすでに存在する場合、新しいスクリプトで *super* キーワードを使用して、既存のインクルードを参照できます。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、インクルードの名前です。
- 2 番目のパラメータは、インクルードの内容を定義する文字列です。この文字列に Idoc スクリプトを含めることができます。

出力

インクルードを解析できなかった場合は 0、解析できる場合は 1 を返します。

この動作を抑制するには、`exec` を使用します。

例

文字列 "My name is *resource include*" を使用して、スクリプトを動的に作成します。

```
<$setResourceInclude("my_name","My name is <$my_name$>")$>
```

super キーワードを使用して、*std_display_field* インクルードを変更します。

```
<$myInclude="<span class='field'><$include super.std_display_field$></span>"
<$setResourceInclude("std_display_field",myInclude)$>
```

次の例では、返される結果を抑制します。

```
<$exec setResourceInclude("std_display_field",myInclude)$>
```

SETVALUE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- ResultSet（5-32 ページ）

説明

ターゲットは、#local または ResultSet の名前です。ターゲットが #local の場合、パラメータ name はローカル・データの変数名を識別し、パラメータ value によって値が設定されます。ターゲット領域が空ではなく、#local でもない場合は、パラメータ target によって指定された ResultSet 内のフィールドが設定の対象となります。ResultSet が存在し、現在の行が有効であれば、その行にあるその特定の列の値（パラメータ name によって指定された列）がパラメータ value の値に設定されます。現在、そのフィールドが ResultSet に含まれていない場合、フィールドは新しい列として ResultSet に追加され、値が設定されます。target で指定された ResultSet の行が有効でない場合、この関数は無効です（ただし、サーバーの出力に内部例外が報告されます）。この関数は、getValue (...) と対照的であることに注意してください。

パラメータ

パラメータを 3 つ取ります。

- 最初のパラメータは、設定されるターゲット領域です（#local または ResultSet の名前）。
- 2 番目のパラメータは、設定される値を保持するキーの名前です（ローカル変数または ResultSet フィールド）。
- 3 番目のパラメータは、設定する値です。

出力

- 関数が成功した場合、TRUE を返します。
- 関数が失敗した場合、FALSE を返します。

例

```
<$setValue(target, name, value)$>
```

SHAREDDIR

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ディレクトリおよびパス（5-22 ページ）
- Inbound Refinery（5-11 ページ）
- クラスタリング（5-5 ページ）

説明

shared ディレクトリのパスを定義します。

- *shared* ディレクトリには、コンテンツ・サーバーの共有ファイルが含まれます。
- Inbound Refinery がインストールされている場合、*shared* ディレクトリに変換エンジンが含まれます。Inbound Refinery のすべての一時的作業は、このディレクトリおよびサブディレクトリで実行されます。
- クラスタ Windows コンテンツ・サーバー構成では、この設定により、コンテンツ・サーバーは、コンテンツ・サーバーがインストールされたファイル・システムのかわりに、ローカル・ディレクトリ（スタブ・ディレクトリ）のコードおよびリソース・ファイルを参照できます。使用される書式は `SharedDir=<stub_dir>/shared` で、*<stub_dir>* は、スタブ・ディレクトリのパスです（`f:/stellclust/` など）。詳細は、クラスタリングのドキュメントを参照してください。
- デフォルト値は、*<install_dir>*`/shared` です。デフォルトを使用すると、構成ファイルに変数は表示されません。

場所

- 「System Properties」→「Paths」タブ→「Shared Directory Path」
- *<install_dir>*/bin/intradoc.cfg

例

次は構成エントリの使用例です。

```
SharedDir=c:/stellent/shared/
```

関連項目

– HttpSharedRoot（6-323 ページ）

SHAREDWEBLAYOUTDIR

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

プロキシ・コンテンツ・サーバーの weblayout ディレクトリのパスを指定します。

- プロキシ・コンテンツ・サーバーでの /weblayout/help/ および /weblayout/images/ ディレクトリの重複を避けたい場合、通常、この設定を使用してマスター・コンテンツ・サーバーを指します。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
SharedWeblayoutDir=c:/stellent/weblayout/
```

SHOWCREATEEXTERNALMENU

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Record Manager および Retention Manager の構成変数（4-20 ページ）

説明

この Content Server 構成変数は、URM に固有のものです。この変数を true に設定すると、コンテンツ・サーバー・インタフェースに、次の外部ソースの作成機能が表示されます。

- 「External Content」→「<external_source_name>」サブメニューの「**Create** <external_source_name> **Item**」オプション「External Content」メニューは、外部ソースが作成されている場合にのみ表示されます。
- 「Exploring Retention Category」ページ→「Action」メニューの「**Create** <external_source_name> **Record**」オプション（レコード・カテゴリ）または「Check in New <external_source_name> Item」オプション（非レコード・カテゴリ）
- 「Exploring Records Folder」ページ→「Action」メニューの「**Create** <external_source_name> **Record**」オプション（レコード・カテゴリのみ）
- 任意のリンク・ページの「Action」メニューの「**Link New** <external_source_name> **Item**」オプション

false に設定すると、これらの機能は表示されません。

デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
ShowCreateExternalMenu=true
```

SHOWDOCACTIONS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）
- ページ表示（5-30 ページ）
- クライアント（5-4 ページ）

説明

検索結果ページに、チェックアウト・リンクを表示するかどうかを定義します。

- この変数は、Oracle Desktop などの ODMA アプリケーション用の追加のコードを評価するためにも使用されます。ODMA クライアントからアクセスされないコンテンツ・サーバーのチェックアウト・リンクを表示するには、showSearchResultsCheckoutColumn（6-746 ページ）変数を使用します。
- TRUE に設定すると、チェックアウト・リンクを含む列が検索結果ページに表示されます。
- FALSE に設定すると、検索結果ページにチェックアウト・リンクは表示されません。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
showDocActions=1
```

関連項目

- showSearchResultsCheckoutColumn（6-746 ページ）
- noInfoColumn（6-543 ページ）
- noRevisionsColumn（6-546 ページ）

SHOWHYPERLINKBOX

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

PDF ファイルのハイパーリンクをボックスで囲むかどうかを定義します。

- TRUE に設定すると、PDF ファイルのハイパーリンクを囲むボックスが設定されます。
- FALSE に設定すると、PDF ファイルのハイパーリンクを囲むボックスは設定されません。
- デフォルトは、FALSE です。

場所

<*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
ShowHyperlinkBox=true
```

SHOWONLYKNOWNACCOUNTS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 内部セキュリティ（5-36 ページ）
- フィールド表示（5-24 ページ）

説明

チェックインおよび検索のページに、事前定義済のアカウントのみを表示するかどうかを定義します。

- TRUE に設定すると、チェックイン・ページおよび検索ページの「Accounts」オプションに、事前定義済のアカウントのみが表示されます。
- FALSE に設定すると、ユーザーが所属するユーザー定義のアカウントもオプション・リストに表示されます。
- デフォルトは、空の文字列です。

場所

- 「System Properties」→「Content Security」タブ→「Show only known accounts」
- 「Admin Server」→「Content Security」→「Show only known accounts」
- <*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
ShowOnlyKnownAccounts=true
```

SHOWREVISIONSCOLUMN

タイプおよび使用方法

- 構成変数（4-3 ページ）
- フィールド表示（5-24 ページ）

説明

Content Server のより新しいバージョンでは、「Classic」表示の使用中に、ドキュメントのリビジョン番号が「Search Results」ページに表示されません。この構成変数を使用して、リビジョン番号を表示します。

- TRUE に設定すると、「Classic」表示の使用中に、ドキュメントのリビジョン番号が「Search Results」ページに表示されます。
- この設定は、Classic 検索モードでのみ機能します。
- デフォルトは、TRUE です。

場所

- <*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
ShowRevisionsColumn=false
```

SHOWSEARCHRESULTSCHECKOUTCOLUMN

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）
- ページ表示（5-30 ページ）

説明

標準検索結果ページに、チェックアウト・リンクを表示するかどうかを定義します。

- TRUE に設定すると、チェックアウト・リンクを含む列が標準検索結果ページに表示されます。
- FALSE に設定すると、標準検索結果ページに、チェックアウト・リンクが表示されません。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
showSearchResultsCheckoutColumn=1
```

関連項目

- showDocActions（6-742 ページ）
- noInfoColumn（6-543 ページ）
- noRevisionsColumn（6-546 ページ）

SHOWWEBDAVURLINBROWSERREQUEST

タイプおよび使用方法

- Collaboration Manager の構成変数（4-4 ページ）
- Collaboration Manager（5-5 ページ）

説明

WebDAV の場所へのドキュメント・リンクを有効にします。

- 通常、コンテンツまたはフォルダの場所に対する URL は、Content Server のコンテンツ・アイテムまたはフォルダへのリンクです。この変数を 1 に設定すると、リンク先が WebDAV の場所になります。
- Collaboration Manager のインストール中に追加されます。

場所

<*install_dir*>/<*instancename*>/data/components/CollaborationManager/config.cfg

例

```
showWebDavUrlInBrowserRequest=1
```

SINGLEGROUP

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- コンテンツ・アイテム（5-6 ページ）
- ワークフロー（5-44 ページ）

説明

現在のリビジョンが、基本ワークフローのコントリビュータ・ステップにあるかどうかを評価します。

出力

- リビジョンがコントリビュータ・ステップにある場合は、TRUE を返します。
- リビジョンがコントリビュータ・ステップにない場合は、FALSE を返します。

例

リビジョンがコントリビュータ・ステップにあるかどうかを評価します。

```
<$if not SingleGroup$>
```

SKIPMOZILLAMAILTOWORKAROUND

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

Mozilla メール・アプリケーションによる「Search Result」および「Content Info」ページからの正確な電子メール処理を可能にする電子メール・ソリューションを無効にします。メーラーから、正しく書式設定された電子メールが送信されない場合は、この変数を TRUE に設定します。

場所

<*install_dir*>/config/config.cfg

例

特殊な電子メール処理を無効にします。

```
SkipMozillaMailToWorkaround=true
```

SMTPPORT

タイプおよび使用方法

❖ 構成変数（4-3 ページ）

❖ コンテンツ・サーバー（5-8 ページ）

説明

Simple Mail Transfer Protocol（SMTP）のポート番号を定義します。

❖ デフォルトは、25 です。

場所

❖ 「System Properties」→「Internet」タブ→「SMTP Port」

❖ <*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
SmtpPort=25
```

関連項目

– SysAdminAddress（6-787 ページ）

– MailServer（6-512 ページ）

SOCKETHOSTADDRESSSECURITYFILTER

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Web サーバー（5-42 ページ）
- 外部セキュリティ（5-37 ページ）

説明

コンテンツ・サーバーへのコールを許可される IP アドレスを定義します。

- コンテンツ・サーバーへのリモート IdcCommandX コールを実行できるようにするには、この設定または SocketHostNameSecurityFilter（6-752 ページ）のどちらかが必要です。（通常、両方の設定を同時に使用しないでください。）詳細は、『IdcCommand—Java Command Utility Reference Guide』を参照してください。
- 複数のアドレスは、パイプ文字（|）を使用して区切ります。パイプの前後に空白が挿入されていないことを確認してください。
- アドレス・グループを示すには、ワイルドカードを使用します（* は 0 個以上の文字を表し、? は任意の 1 文字を表します）。
- デフォルトは、ローカル・ホストの IP アドレスです。

場所

- 「System Properties」→「Server」タブ→「IP Address Filter」
- <*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
SocketHostAddressSecurityFilter=10.10.3.*|127.0.0.1
```

関連項目

– SocketHostNameSecurityFilter（6-752 ページ）

SOCKETHOSTNAMESECURITYFILTER

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Web サーバー（5-42 ページ）
- 外部セキュリティ（5-37 ページ）

説明

コンテンツ・サーバーへのコールを許可されるサーバー名を定義します。

- コンテンツ・サーバーへのリモート IdcCommandX コールを実行できるようにするには、この設定または SocketHostAddressSecurityFilter（6-751 ページ）のどちらかが必要です。（通常、両方の設定を同時に使用しないでください。）詳細は、『Idc コマンド・リファレンス・ガイド』を参照してください。
- 複数のサーバー名は、パイプ文字（|）を使用して区切ります。パイプの前後に空白が挿入されていないことを確認してください。

場所

- 「System Properties」→「Server」タブ→「Hostname Filter」
- <*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
SocketHostNameSecurityFilter=olympus|localhost|zeus
```

関連項目

– SocketHostAddressSecurityFilter（6-751 ページ）

SOURCEID

タイプおよび使用方法

- コンテンツ読取り専用変数（4-33 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

現在の動的サーバー・ページのコンテンツ ID を提供します。

- この変数によって返される値は、`ref:dID` と同じです。3-36 ページの「動的サーバー・ページでのメタデータの参照」を参照してください。

出力

コンテンツ ID を文字列で返します。

例

dID 変数に、現在の HCSF ページの値を設定します。

```
<input type=hidden name="dID" value="<$SourceID$>">
```

SOURCEPATH

タイプおよび使用方法

❖ 構成変数（4-3 ページ）

❖ 外部セキュリティ（5-37 ページ）

説明

LDAP プロバイダを識別する一意の文字列を指定します。ユーザーが初めてプロバイダを通じて資格証明を要求したとき、この文字列がユーザー情報とともに格納され、次回ユーザーが資格証明を要求したときにユーザーをプロバイダと照合するために使用されます。

❖ LDAP プロバイダ名を Source Path として使用することをお薦めします。

❖ デフォルトはありません。

場所

❖ 「Add/Edit LDAP Provider」ページ→「Source Path」

❖ <*install_dir*>/data/providers/*provider_name*/provider.hda

例

次は構成エントリの使用例です。

```
SourcePath=LdapProvider1
```

関連項目

– Priority（6-596 ページ）

SPECIALAUTHGROUPS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 内部セキュリティ（5-36 ページ）

説明

特殊な権限を持つセキュリティ・グループを定義します。

- この設定は、Stellent Collaboration Server および Need to Know コンポーネント（Stellent Content Server DVD の extras/NtkDocDisclosure/ ディレクトリ）で使用されます。これらの製品を使用する場合、詳細は関連ドキュメントを参照してください。
- 文字列を返します。
- デフォルト値はありません。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
SpecialAuthGroups=executive,subadmin
```

StatusCode

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

最後に行った関数コールが成功または失敗したかを示します。

- 一部の関数は、成功すると、この変数を 0 に設定します。
- 一部の関数は、失敗すると、この変数を負の数値（-1）に設定します。通常では、abortToErrorPage（6-2 ページ）関数をコールして、最新の関数コールからの StatusMessage（6-757 ページ）とともにエラー・ページを表示します。

出力

- 最後に行った関数コールが成功した場合は、0 を返します。
- 最後に行った関数コールが失敗した場合は、-1 を返します。

例

現在のステータス・コードを返します。

```
<$StatusCode$>
```

関連項目

- abortToErrorPage（6-2 ページ）
- getUserValue（6-274 ページ）
- hasAppRights（6-281 ページ）
- StatusMessage（6-757 ページ）

STATUSMESSAGE

タイプおよび使用方法

- 値変数（4-34 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

最後に行った関数コールのエラー・メッセージを定義します。

- 通常、この変数は、abortToErrorPage（6-2 ページ）関数をコールしたときにエラー・ページに表示されます。

出力

- 最後に行った関数コールが成功した場合は、OK を返します。
- 最後に行った関数コールが失敗した場合は、ローカライズされたエラー・メッセージを返します。

例

現在のステータス・メッセージを返します。

```
<$StatusMessage$>
```

関連項目

– abortToErrorPage（6-2 ページ）
– StatusCode（6-756 ページ）

STDPAGEWIDTH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ページ表示（5-30 ページ）

説明

ページ幅をピクセル単位で設定します。

- デフォルトは、550 です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
StdPageWidth=1000
```

ページ幅を文字列で返します。

```
<$StdPageWidth$>
```

stdSecurityCheck

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 内部セキュリティ（5-36 ページ）

説明

標準セキュリティ・モデルに基づいて、現在のユーザーがリクエストされたコンテンツ・アイテムにアクセスできるかどうかをチェックします。

- アクティブなデータをチェックして、標準（またはデフォルト）のセキュリティ・モデルで、コンテンツ・アイテムへのユーザーのアクセスが許可されるかどうかを判断します。これにより、カスタムなセキュリティを実装した場合でも、標準セキュリティ・モデルをベースラインとして実行することが可能になります。

パラメータ

なし。

出力

- 現在のユーザーがコンテンツ・アイテムにアクセスできる場合は、TRUE を返します。
- 現在のユーザーがコンテンツ・アイテムにアクセスできない場合は、FALSE を返します。

例

ユーザーの権限レベルをリクエストされたコンテンツ・アイテムと比較します。

```
<$stdSecurityCheck()$>
```

STRCENTERPAD

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 文字列（5-39 ページ）

説明

文字列の両側に同等の空白を埋め込みます。文字列の両側に同数の空白を追加して、指定された長さ以上の文字列にします。必要に応じて、文字列の長さに 1 文字追加されます。

注意：文字列連結、文字列論理和および簡単な文字列比較には、特殊な文字列演算子（3-23 ページ）を使用します。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、文字列です。
- 2 番目のパラメータは、文字列の長さです。

出力

両側に空白を含む文字列を返します。

例

<space><space>inf<space><space> の形式を使用して、両側に同等の空白を埋め込み、長さが 7 文字の文字列を作成します。

```
<$strCenterPad("inf", 7)$>
```

<space><space><space>inf<space><space><space> の形式を使用して、両側に同等の空白を埋め込み、長さが 9 文字の文字列を作成します。

```
<$strCenterPad("inf", 8)$>
```

<space><space><space>inf<space><space><space> の形式を使用して、両側に同等の空白を埋め込み、長さが 9 文字の文字列を作成します。

```
<$strCenterPad("inf", 9)$>
```

RPTDISPLAYMAPVALUE

タイプおよび使用方法

- グローバル関数
- ローカライゼーション

説明

この関数は、内部使用のみです。

この関数は、内部キー・コードのローカライズされた文字列表現を返します。これらのキー・コードは、コンテンツ・アイテムのステータスおよび状態フラグの保存に使用されます。この関数は、内部状態の説明を表示するために「Content Information」ページおよびワークフロー・ページで使用されます。

パラメータ

パラメータを 2 つ取ります。

- キーが格納されている表の名前
- ローカライズの対象となるキー

STRCOMMAAPPENDNODUPLICATES

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 文字列（5-39 ページ）

説明

この関数は、既存の文字列に新しいトークンを追加します。この関数は、文字列とカンマと新しいトークンを返します。トークンが文字列にすでに存在する場合、トークンは追加されません。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、文字列です。
- 2 番目のパラメータは、トークンです。

例

```
<$myString = strCommaAppendNoDuplicates(myString, "A")$>
<$myString = strCommaAppendNoDuplicates(myString, "B")$>
<$myString = strCommaAppendNoDuplicates(myString, "C")$>
<$myString = strCommaAppendNoDuplicates(myString, "A")$>
<$myString = strCommaAppendNoDuplicates(myString, "B")$>
```

STRCONFINE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 文字列（5-39 ページ）

説明

文字列を最大長に制限し、必要に応じてパディングを追加します。

- 文字列が指定された長さ以下の場合、文字列は影響を受けません。
- 文字列が指定された長さより長い場合、文字列は短縮され、指定された長さと等しくなるように 3 つの埋込み文字が追加されます。StrConfineOverflowChars（6-765 ページ）変数を変更して、埋込みに使用する文字を指定できます。

注意：文字列連結、文字列論理和および簡単な文字列比較には、特殊な文字列演算子（3-23 ページ）を使用します。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、文字列です。
- 2 番目のパラメータは、文字列の最大長です。

出力

文字列を返します。

例

in<dot><dot><dot> の形式を使用して、文字列を制限し、長さが 5 文字の文字列になるように 3 つの埋込み文字（ドット）を追加します。

```
<$strConfine("inform", 5)$>
```

i<dot><dot><dot><dot> の形式を使用して、文字列を制限し、長さが 5 文字の文字列になるように埋込み文字を追加します。

```
<$strConfine("i", 5)$>
```

長さが最大 10 文字の文字列に制限されるため、この 6 文字の文字列は影響を受けません。

```
<$strConfine("inform", 10)$>
```

長さが最大 6 文字の文字列に制限されるため、この 6 文字の文字列は影響を受けません。

```
<$strConfine("inform", 6)$>
```

関連項目

– StrConfineOverflowChars（6-765 ページ）

STRCONFINEOVERFLOWCHARS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 文字列（5-39 ページ）

説明

文字列の埋込み文字を定義します。

- strConfine（6-763 ページ）関数によって埋込みに使用される文字を定義します。
- デフォルトは、ドットです。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
StrConfineOverflowChars=.
```

関連項目

– strConfine（6-763 ページ）

STREQUALS

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 文字列（5-39 ページ）

説明

2 つの文字列が、大 / 小文字の区別も含めて等しいかどうかをチェックします。

注意 : 文字列連結、文字列論理和および簡単な文字列比較には、特殊な文字列演算子（3-23 ページ）を使用します。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、文字列です。
- 2 番目のパラメータは、最初のパラメータと比較する文字列です。

出力

- 文字列が等しい場合は、TRUE を返します。
- 文字列が等しくない場合は、FALSE を返します。

例

文字列が等しいかどうかを評価し、TRUE (1) を返します。

```
<$strEquals("Home","Home")$>
```

文字列が等しいかどうかを評価し、FALSE (0) を返します。

```
<$strEquals("home","Home")$>
```

関連項目

- regexMatches（6-638 ページ）
- regexReplaceAll（6-639 ページ）
- regexReplaceFirst（6-640 ページ）
- strEqualsIgnoreCase（6-767 ページ）

STREQUALSIGNORECASE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 文字列（5-39 ページ）

説明

大 / 小文字の区別は含めずに、2 つの文字列が等しいかどうかをチェックします。

注意 : 文字列連結、文字列論理和および簡単な文字列比較には、特殊な文字列演算子（3-23 ページ）を使用します。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、文字列です。
- 2 番目のパラメータは、最初のパラメータと比較する文字列です。

出力

- 文字列が等しい場合は、TRUE を返します。
- 文字列が等しくない場合は、FALSE を返します。

例

文字列が等しいかどうかを評価し、TRUE (1) を返します。

```
<$strEqualsIgnoreCase("home","Home")$>
```

文字列が等しいかどうかを評価し、FALSE (0) を返します。

```
<$strEqualsIgnoreCase("home","page")$>
```

関連項目

– strEquals（6-766 ページ）

STRGENERATERANDOM

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 文字列（5-39 ページ）

説明

この関数は、ランダムな 16 進文字列を生成します。`length` パラメータによる指定と同数の文字を出力します。長さを指定しない場合、デフォルトは 16 文字です。

パラメータ

1 つのパラメータを取ります。

- 文字列の長さ

STRINDEXOF

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 文字列（5-39 ページ）

説明

ある文字列が、別の文字列の部分文字列かどうかを判断します。

注意：文字列連結、文字列論理和および簡単な文字列比較には、特殊な文字列演算子（3-23 ページ）を使用します。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、文字列です。
- 2 番目のパラメータは、可能な部分文字列です。

出力

- 2 番目の文字列が最初の文字列の部分文字列である場合、最初の文字列のどこから部分文字列が開始するかを示す索引値を返します。最初の文字の索引値は 0 です。
- 2 番目の文字列が最初の文字列の部分文字列でない場合は、-1 の値を返します。

例

"xy" が "xyz" の部分文字列かどうかを評価し、索引値 0 を返します。

```
<$if strIndexOf("xyz","xy") >=0$> check for substring <$endif$>
```

"yz" が "xyz" の部分文字列かどうかを評価し、索引値 1 を返します。

```
<$if strIndexOf("xyz","yz") >=0$> check for substring <$endif$>
```

"ab" が "xyz" の部分文字列かどうかを評価し、索引値 -1 を返して部分文字列でないことを示します。

```
<$if strIndexOf("xyz","ab") >=0$> check for substring <$endif$>
```

STRLEFTFILL

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 文字列（5-39 ページ）

説明

文字列の左側に文字を埋めて指定された長さにします。

注意：文字列連結、文字列論理和および簡単な文字列比較には、特殊な文字列演算子（3-23 ページ）を使用します。

パラメータ

パラメータを 3 つ取ります。

- 最初のパラメータは、文字列です。
- 2 番目のパラメータは、補てん文字です。
- 3 番目のパラメータは、長さです。

出力

必要に応じて指定された文字が左側に埋め込まれた文字列を返します。

例

文字 Z を左側に埋め込んで文字列 *sleep* を 10 スペースにフォーマットします。これは、文字列 *ZZZZZsleep* を返します。

```
<$strLeftFill("sleep",'Z',10)$>
```

文字列 *sleep* を返す例を次に示します。

```
<$strLeftFill("sleep",'Z',4)$>
```

STRLEFTPAD

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 文字列（5-39 ページ）

説明

文字列の左側に追加の空白を埋め込んで指定された長さにします。

注意：文字列連結、文字列論理和および簡単な文字列比較には、特殊な文字列演算子（3-23 ページ）を使用します。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、文字列です。
- 2 番目のパラメータは、長さです。

出力

必要に応じて左側に空白が埋め込まれた文字列を返します。

例

`<space><space>inf` の形式を使用して、左側に追加の空白を埋め込んで長さが 5 文字の文字列にします。

```
<$strLeftPad("inf", 5)$>
```

文字列 information を返します。

```
<$strLeftPad("information", 5)$>
```

STRLENGTH

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 文字列（5-39 ページ）

説明

文字列の長さを評価します。

注意：文字列連結、文字列論理和および簡単な文字列比較には、特殊な文字列演算子（3-23 ページ）を使用します。

パラメータ

1 つのパラメータを取ります。

- パラメータは、文字列のみです。

出力

整数値を返します。

例

文字列 home の長さを評価し、整数の 4 を返します。

```
<$strLength("home")$>
```

STRLOWER

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 文字列（5-39 ページ）

説明

文字列をすべて小文字にフォーマットします。

注意：文字列連結、文字列論理和および簡単な文字列比較には、特殊な文字列演算子（3-23 ページ）を使用します。

パラメータ

1 つのパラメータを取ります。

- パラメータは、文字列のみです。

出力

文字列をすべて小文字で返します。

例

文字列 Home を評価し、*home* を返します。

```
<$strLower("Home")$>
```

STRREMOVEWS

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 文字列（5-39 ページ）

説明

文字列から空白を削除します。

注意：文字列連結、文字列論理和および簡単な文字列比較には、特殊な文字列演算子（3-23 ページ）を使用します。

パラメータ

1 つのパラメータを取ります。

- パラメータは、文字列のみです。

出力

空白のない文字列を返します。

例

文字列 h o m e を文字列 *home* にフォーマットします。

```
<$strRemoveWs("h o m e")$>
```

STRREPLACE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 文字列（5-39 ページ）

説明

既存の部分文字列を別の部分文字列に置換します。

- 置換する文字列が複数の場合、それらはすべて新しい部分文字列に置換されます。

注意：文字列連結、文字列論理和および簡単な文字列比較には、特殊な文字列演算子（3-23 ページ）を使用します。

パラメータ

パラメータを 3 つ取ります。

- 最初のパラメータは、置換を実行する文字列です。
- 2 番目のパラメータは、置換される部分文字列です。
- 3 番目のパラメータは、既存の部分文字列を置き換える部分文字列です。

出力

文字列を返します。

例

単語 *classified* を *restricted* に置き換え、"This document is restricted" という文字列にします。

```
<$strReplace("This document is classified.","classified","restricted")$>
```

日付のスラッシュをピリオドに置き換えて、"6.20.2001" という形式の日付にします。

```
<$strReplace(formatDateOnly(dateCurrent()),"/",".")$>
```

STRREPLACEIGNORECASE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 文字列（5-39 ページ）

説明

大 / 小文字の区別を使用せずに検索を実行して、既存の部分文字列を別の部分文字列に置換します。

- 置換する文字列が複数の場合、それらはすべて新しい部分文字列に置換されます。

パラメータ

パラメータを 3 つ取ります。

- 最初のパラメータは、置換を実行する文字列です。
- 2 番目のパラメータは、置換される部分文字列です。
- 3 番目のパラメータは、既存の部分文字列を置き換える部分文字列です。

出力

文字列を返します。

例

```
<$hello = "Hello world!I love the world!"$>
<$strReplaceIgnoreCase(hello, "WORLD", "universe")$>
```

STRRIGHTFILL

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 文字列（5-39 ページ）

説明

文字列の右側に文字を埋めて指定された長さにします。

注意：文字列連結、文字列論理和および簡単な文字列比較には、特殊な文字列演算子（3-23 ページ）を使用します。

パラメータ

パラメータを 3 つ取ります。

- 最初のパラメータは、文字列です。
- 2 番目のパラメータは、補てん文字です。
- 3 番目のパラメータは、長さです。

出力

必要に応じて指定された文字が右側に埋め込まれた文字列を返します。

例

文字 Z を右側に埋め込んで文字列 *sleep* を 10 スペースにフォーマットします。これは、文字列 *sleep*ZZZZZ を返します。

```
<$strRightFill("sleep",'Z',10)$>
```

文字列 sleep を返します。

```
<$strRightFill("sleep",'Z',4)$>
```

STRRIGHTPAD

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 文字列（5-39 ページ）

説明

文字列の右側に追加の空白を埋め込んで指定された長さにします。

注意：文字列連結、文字列論理和および簡単な文字列比較には、特殊な文字列演算子（3-23 ページ）を使用します。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、文字列です。
- 2 番目のパラメータは、長さです。

出力

必要に応じて右側に空白が埋め込まれた文字列を返します。

例

`inf<space><space>` の形式を使用して、右側に追加の空白を埋め込んで長さが 5 文字の文字列にします。

```
<$strRightPad("inf", 5)$>
```

文字列 information を返します。

```
<$strRightPad("information", 5)$>
```

STRSUBSTRING

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 文字列（5-39 ページ）

説明

文字列から部分文字列を取得します。

- 最初の文字の索引値は 0 です。

注意：文字列連結、文字列論理和および簡単な文字列比較には、特殊な文字列演算子（3-23 ページ）を使用します。

パラメータ

パラメータを 2 つまたは 3 つ取ります。

- 最初のパラメータは、文字列です。
- 2 番目のパラメータは、開始索引を表す数値です。
- 3 番目のオプション・パラメータは、終了索引を表す数値です。

出力

部分文字列を返します。

- 終了索引パラメータを指定した場合、開始索引の文字で始まり、終了索引の 1 文字前で終わる部分文字列が返されます。
- 終了索引を指定しない場合は、開始索引から文字列の最後までの部分文字列が返されます。

例

文字列 *my* の最初の 2 文字を取得します。

```
<$strSubstring("mystring",0,2)$>
```

string の 2 番目の文字の後ろにある文字列を取得します。

```
<$strSubstring("mystring",2)$>
```

関連項目

– regexReplaceAll（6-639 ページ）
– regexReplaceFirst（6-640 ページ）

STRTRIMWS

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 文字列（5-39 ページ）

説明

文字列の先頭と末尾の空白を削除します。

注意：文字列連結、文字列論理和および簡単な文字列比較には、特殊な文字列演算子（3-23 ページ）を使用します。

パラメータ

1 つのパラメータを取ります。

- パラメータは、文字列のみです。

出力

先頭と末尾の空白がない文字列を返します。

例

文字列 " homepage " を文字列 *homepage* としてフォーマットします。

```
<$strTrimWs(" homepage ")$>
```

STRUPPER

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 文字列（5-39 ページ）

説明

文字列をすべて大文字にフォーマットします。

注意：文字列連結、文字列論理和および簡単な文字列比較には、特殊な文字列演算子（3-23 ページ）を使用します。

パラメータ

1 つのパラメータを取ります。

- パラメータは、文字列のみです。

出力

文字列をすべて大文字で返します。

例

文字列 Home を評価し、*HOME* を返します。

```
<$strUpper("Home")$>
```

StyleForRequiredFields

タイプおよび使用方法

- その他のフィールド表示変数（4-29 ページ）
- フィールド表示（5-24 ページ）

説明

必須メタデータ・フィールドの表示方法を設定します。

- 設定すると、Cascading Style Sheet に定義されたスタイルで必須フィールドが表示されます。
- スタイルの値は、Content Server で定義された任意のフィールド・キャプション・スタイルです。

パラメータ

1 つのパラメータを取ります。

- パラメータは、スタイルの指定のみです。

場所

<*install_dir*>/config/config.cfg

例

次の例では、必須フィールドが赤く表示されます。

```
StyleForRequiredFields=requiredField
```

関連項目

- fieldCaptionStyle（6-218 ページ）
- NotationForRequiredFields（6-548 ページ）

SUBJECTSDIR

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

subjects ディレクトリのパスを定義します。

- サブジェクトは、コンテンツ・サーバー・サービスによって、オプション・リストの変更や新規ユーザーの追加など、システムに変更が加えられたときに通知されるサブシステムです。
- デフォルトは、<*install_dir*>*/data/subjects/* です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
SubjectsDir=c:/stellent_temp/data/subjects
```

SUBSCRIPTIONCOLLATEDFIELDLIST

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

サブスクリプションに表示されるフィールドのリストを定義します。

- 起動時に定義されるデフォルトのフィールドは、dID および dDocName です。
- 追加のフィールドは、`AdditionalSubscriptionCollatedFieldList` を使用して定義できます。
- `SubscriptionCollatedFieldList` を変更するかわりに、`AdditionalSubscriptionCollatedFieldList` を使用してリストにサブスクリプション・フィールドを追加することをお薦めします。

場所

<*install_dir*>/config/config.cfg

関連項目

– AdditionalSubscriptionCollatedFieldList（6-14 ページ）

SUBSCRIPTIONMETHOD

タイプおよび使用方法

❖ 構成変数（4-3 ページ）

❖ コンテンツ・サーバー（5-8 ページ）

説明

サブスクリプションの処理方法を指定します。

❖ 次の値があります。

- **SubscriptionOriented:** コンテンツ・サーバーは、各サブスクリプションを索引付けされたコンテンツ・アイテムと比較します。

  この値は、コンテンツ・アイテム数と比べてサブスクリプション数が少ない場合に使用します（たとえば、100,000 個のコンテンツ・アイテムと 1000 個のサブスクリプション）。

- **DocumentOriented:** コンテンツ・サーバーは、索引付けされた各コンテンツ・アイテムをアクティブなサブスクリプションと比較します。

  この値は、コンテンツ・アイテム数と比べてサブスクリプション数が多い場合に使用します（たとえば、1000 個のコンテンツ・アイテムと 1000 個のサブスクリプション）。

❖ デフォルトは、*SubscriptionOriented* です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
SubscriptionMethod=DocumentOriented
```

SysAdminAddress

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ユーザー（5-40 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

システム管理者の電子メール・アドレスを定義します。

- これは、コンテンツ・サーバーから受信するワークフローおよびサブスクリプションの電子メール通知に使用されるアドレスです。
- 文字列を返します。
- デフォルト値はありません。

場所

- 「System Properties」→「Internet」タブ→「Administrator Mail Address」
- 「Admin Server」→「Internet Configuration」→「Administrator Mail Address」
- <*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
SysAdminAddress=admin@companyname.com
```

スクリプトとして、構成設定の値を返します。

```
<$SysAdminAddress$>
```

関連項目

- MailServer（6-512 ページ）
- SmtpPort（6-750 ページ）

SYSTEMDATEFORMAT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 日付と時間（5-19 ページ）
- ローカライゼーション（5-29 ページ）

説明

コンテンツ・サーバーの日付 / 時間の書式を設定します。

日付には 2 つのタイプの書式があります。秒数を保存する書式と、秒数を切り捨て、日付を分まで保存する書式です。秒数を使用する場合は、2037 年までのみ日付を保存できます。

- SystemLocale（6-789 ページ）を構成設定として指定しない場合、*SystemDateFormat* は、コンテンツ・サーバーの構成表の *SystemLocale* 設定ではなく、Java VM の OS 設定から直接に推測されます。*SystemLocale* を明示的に定義すると、そのロケールのコンテンツ・サーバー構成表から、日付 / 時間の書式が使用されます。
- この書式の構文は、標準 Java クラス・ライブラリで提供されている日付書式の拡張機能です。

注意：日付と時間の書式の詳細は、『Using Content Server in International Environments』を参照してください。

場所

<*install_dir*>/config/config.cfg

例

4 桁の年の書式を使用して、米国の標準日時を設定する構成エントリとしての使用例です。

```
SystemDateFormat=M/d{/yyyy} {h:mm[:ss] {aa}[zzz]}
```

関連項目

- SystemLocale（6-789 ページ）
- SystemTimeZone（6-794 ページ）

SYSTEMLOCALE

タイプおよび使用方法

❖ 構成変数（4-3 ページ）

❖ ローカライゼーション（5-29 ページ）

説明

コンテンツ・サーバーのシステム・ロケールを設定します。

❖ この設定によって制御される内容は次のとおりです。

- コンテンツ・サーバーのインタフェースで使用されるローカライズされた文字列
- Web ページの文字エンコーディング
- 日付 / 時間の書式
- タイムゾーン
- フルテキスト索引付けで使用されるパラメータ

これらの機能は、個別の構成エントリまたは *SystemLocale* 設定の変更により、個別に制御できます。

❖ この設定は、System Properties ユーティリティの「Server」タブで変更できます。特定のロケールのデフォルトは、System Properties ユーティリティの「Localization」タブで変更できます。

❖ *SystemLocale* を構成設定として指定しない場合、SystemDateFormat（6-788 ページ）は、コンテンツ・サーバーの構成表の *SystemLocale* 設定ではなく、Java VM の OS 設定から直接に推測されます。*SystemLocale* を明示的に定義すると、そのロケールのコンテンツ・サーバー構成表から、日付 / 時間の書式が使用されます。

❖ 構成エントリ VerityLocale（6-905 ページ）および VerityEncoding（6-903 ページ）によって、*SystemLocale* からの設定は上書きされますが、互換性のない *VerityLocale* および *VerityEncoding* の選択肢が多数あるため、直接に *SystemLocale* を構成するのが優先されるソリューションです。

注意：ローカライゼーションの詳細は、『Using Content Server in International Environments』を参照してください。

場所

❖ 「System Properties」→「Server」タブ→「System Locale」

❖ <*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
SystemLocale=Deutsch
```

関連項目

– SystemDateFormat（6-788 ページ）
– SystemTimeZone（6-794 ページ）

SYSTEMREPORTSDIR

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ディレクトリおよびパス（5-22 ページ）
- テンプレート（5-40 ページ）

説明

Web レイアウト・エディタのレポート用のテンプレートが含まれる *reports* ディレクトリのパスを定義します。

- デフォルトは、*<install_dir>/shared/config/reports/* です。

場所

<install_dir>/config/config.cfg

例

次は構成エントリの使用例です。

```
SystemReportsDir=c:/myreports/
```

SYSTEMRESOURCESDIR

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

標準コンテンツ・サーバー・リソースが含まれる *resources* ディレクトリのパスを定義します。

- デフォルトは、*<install_dir>/shared/config/resources/* です。

場所

<install_dir>/config/config.cfg

例

次は構成エントリの使用例です。

```
SystemResourcesDir=c:/std_resources/
```

SYSTEMTEMPLATESDIR

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ディレクトリおよびパス（5-22 ページ）
- テンプレート（5-40 ページ）

説明

標準コンテンツ・サーバー・ページ・テンプレートが含まれる *templates* ディレクトリのパスを定義します。

- デフォルトは、<*install_dir*>*/shared/config/templates/* です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
SystemTemplatesDir=c:/std_templates
```

SYSTEMTIMEZONE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 日付と時間（5-19 ページ）
- ローカライゼーション（5-29 ページ）

説明

コンテンツ・サーバーのデフォルトのタイムゾーンを設定します。

- <*install_dir*>/shared/config/resources/std_locale.htm ファイルの *SystemTimeZones* 表に、指定可能なタイムゾーンのリストがあります。
- デフォルトは、ローカル・コンピュータのオペレーティング・システムのタイムゾーン設定です。デフォルトを使用すると、構成ファイルに変数は表示されません。

注意：タイムゾーンの書式の詳細は、『Using Content Server in International Environments』を参照してください。

場所

- 「System Properties」→「Server」タブ→「System Timezone」
- <*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
SystemTimeZone=Europe/Berlin
```

関連項目

- SystemDateFormat（6-788 ページ）
- SystemLocale（6-789 ページ）

TEMPDIR

タイプおよび使用方法

- 構成変数（4-3 ページ）
- デバッグ（5-20 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

Content Server Analyzer ユーティリティがファイルを複製およびテストする場所を指定します。

- デフォルトは、IdcAnalyzeLogDir（6-330 ページ）設定で指定されたログ・ディレクトリ内の *temp* サブディレクトリです。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
TempDir=f:/stellent/analyzer_logs/temp/
```

関連項目

– IdcAnalyzeLogDir（6-330 ページ）

TEMPLATECLASS

タイプおよび使用方法

- テンプレート読取り専用変数（4-31 ページ）
- ページ表示（5-30 ページ）
- テンプレート（5-40 ページ）

説明

現在のページのテンプレートの分類を取得します。

- この変数を使用すると、テンプレートのアイデンティティに基づいてテンプレート内に条件付きコンテンツを作成できます。サーバーから配信されるページのソースを調べるには、テンプレート・ページ中のこの変数を使用します。
- 標準テンプレートの場合、この変数は、*templates.hda* ファイルの *IntradocTemplates* 表の *class* 列に定義されます。
- 結果テンプレートの場合、この変数は `Results` に評価されます。
- レポートテンプレートの場合、この変数は `Reports` に評価されます。
- 動的サーバー・ページでは、この変数は通常、`IdcDynamicFile` です。

出力

文字列を返します。

例

このマークアップは、テンプレート情報の表をページに表示します。

```
<TABLE>
    <TR><TD>Template Name</TD>
    <TD><$TemplateName$></TD></TR>
    <TR><TD>Template Class</TD>
    <TD><$TemplateClass$></TD></TR>
    <TR><TD>Template Type</TD>
    <TD><$TemplateType$></TD></TR>
    <TR><TD>Template File Path</TD>
    <TD><$TemplateFilePath$></TD></TR>
</TABLE>
```

関連項目

- TemplateFilePath（6-798 ページ）
- TemplateName（6-799 ページ）
- TemplateType（6-800 ページ）

TEMPLATEFILEPATH

タイプおよび使用方法

- テンプレート読取り専用変数（4-31 ページ）
- ページ表示（5-30 ページ）
- テンプレート（5-40 ページ）

説明

テンプレートが実際にロードされたパスを取得します。

- この変数を使用すると、テンプレートのアイデンティティに基づいてテンプレート内に条件付きコンテンツを作成できます。サーバーから配信されるページのソースを調べるには、テンプレート・ページ中のこの変数を使用します。

出力

文字列としてパスを返します。

例

このマークアップは、テンプレート情報の表をページに表示します。

```
<TABLE>
    <TR><TD>Template Name</TD>
    <TD><$TemplateName$></TD></TR>
    <TR><TD>Template Class</TD>
    <TD><$TemplateClass$></TD></TR>
    <TR><TD>Template Type</TD>
    <TD><$TemplateType$></TD></TR>
    <TR><TD>Template File Path</TD>
    <TD><$TemplateFilePath$></TD></TR>
</TABLE>
```

関連項目

- TemplateClass（6-796 ページ）
- TemplateName（6-799 ページ）
- TemplateType（6-800 ページ）

TEMPLATENAME

タイプおよび使用方法

- テンプレート読取り専用変数（4-31 ページ）
- ページ表示（5-30 ページ）
- テンプレート（5-40 ページ）

説明

現在のページのテンプレートの内部名を取得します。たとえば、DOC_INFO または CHECKIN_NEW_FORM など。

- この変数を使用すると、テンプレートのアイデンティティに基づいてテンプレート内に条件付きコンテンツを作成できます。サーバーから配信されるページのソースを調べるには、テンプレート・ページ中のこの変数を使用します。

出力

テンプレート名を文字列で返します。

例

このマークアップは、テンプレート情報の表をページに表示します。

```
<TABLE>
    <TR><TD>Template Name</TD>
    <TD><$TemplateName$></TD></TR>
    <TR><TD>Template Class</TD>
    <TD><$TemplateClass$></TD></TR>
    <TR><TD>Template Type</TD>
    <TD><$TemplateType$></TD></TR>
    <TR><TD>Template File Path</TD>
    <TD><$TemplateFilePath$></TD></TR>
</TABLE>
```

関連項目

- TemplateClass（6-796 ページ）
- TemplateFilePath（6-798 ページ）
- TemplateType（6-800 ページ）

TEMPLATETYPE

タイプおよび使用方法

- テンプレート読取り専用変数（4-31 ページ）
- ページ表示（5-30 ページ）
- テンプレート（5-40 ページ）

説明

現在のページのテンプレート・タイプを提供します。

- この変数を使用すると、テンプレートのアイデンティティに基づいてテンプレート内に条件付きコンテンツを作成できます。サーバーから配信されるページのソースを調べるには、テンプレート・ページ中のこの変数を使用します。
- 標準テンプレートの場合、この変数は、*templates.hda* ファイルの *IntradocTemplates* 表の *formtype* 列に定義されます。
- 検索結果テンプレートの場合、この変数は、*templates.hda* ファイルの *SearchResultTemplates* 表の *formtype* 列に定義されます。
- レポート・テンプレートの場合、この変数は、*reports.hda* ファイルの *IntradocReports* 表の *datasource* 列に定義されます。
- 動的サーバー・ページでは、この変数は、`hcsp` または `hcsf` のいずれかです。

出力

テンプレート・タイプを文字列で返します。

例

このマークアップは、テンプレート情報の表をページに表示します。

```
<TABLE>
    <TR><TD>Template Name</TD>
    <TD><$TemplateName$></TD></TR>
    <TR><TD>Template Class</TD>
    <TD><$TemplateClass$></TD></TR>
    <TR><TD>Template Type</TD>
    <TD><$TemplateType$></TD></TR>
    <TR><TD>Template File Path</TD>
    <TD><$TemplateFilePath$></TD></TR>
</TABLE>
```

関連項目

- TemplateClass（6-796 ページ）
- TemplateFilePath（6-798 ページ）
- TemplateName（6-799 ページ）

TEXTINDEXERFILTERVALIDATEFORMATS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）

説明

この変数は、読み取れない文字がないかを検証されるファイル・タイプをカンマ区切りのリストで指定します。この変数を使用すると、指定されたファイル形式は、読み取れない文字がないかをチェックされ、それらの文字は、Verity Integration での索引付けのためにテキスト・ファイルを渡す前に修正されます。Verity Integration は、Content Server とともに使用できるアドオン製品であり、購入することができます。

場所

<*install_dir*>/custom/TextIndexerFilter/textindexerfilter_einviornment.cfg

例

```
TextIndexerFilterValidateFormats=application/pdf
```

THUMBNAILDPI

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

サムネイルの解像度を 1 インチ当たりのドット数（DPI）で設定します。

- ピクセルは、ターゲット・デバイスごとにサイズが異なるため、サムネイルが予定よりも低品質で表示される場合があります。この設定を使用して、特定の解像度を設定すると、ターゲット・デバイスに適切なサイズおよび解像度のサムネイルを確保できます。
- 有効な範囲は、1 ～ 2400 です。
- デフォルトは、34 です。

場所

- 「Local Inbound Refinery Configuration」画面→「Thumbnail Options」タブ→「Thumbnail DPI」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

構成設定として、サムネイルの解像度を定義します。

```
ThumbnailDPI=96
```

関連項目

- ThumbnailHeight（6-804 ページ）
- ThumbnailWidth（6-809 ページ）

THUMBNAILHEIGHT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

サムネイルの高さをピクセル単位で設定します。

- デフォルトは、80 です。

場所

<*install_dir*>/config/config.cfg

注意：この変数は、Inbound Refinery のローカルな構成ファイル（<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg）に保存できますが、Inbound Refinery が読み取るためにコンテンツ・サーバーの構成ファイルにも保存する必要があります。

例

構成設定として、サムネイルの高さをピクセル数で定義します。

```
ThumbnailHeight=100
```

スクリプトとして、構成設定の値を返します。

```
<$ThumbnailHeight$>
```

THUMBNAILJPEGQUALITY

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

サムネイルを作成するための JPEG 圧縮の損失の程度を設定します。

- 有効な範囲は、1 ～ 100 です。100 の場合、画質は最高ですが、圧縮率は最低です。1 にすると、画質は最低ですが、圧縮率は高くなります。
- デフォルトは、100 です。

場所

- 「Local Inbound Refinery Configuration」画面→「Thumbnail Options」タブ→「Thumbnail quality」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

構成設定として、サムネイルの損失の程度を定義します。

```
ThumbnailJPEGQuality=50
```

THUMBNAILSIZELIMIT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

サムネイル・イメージの最大サイズをピクセル数で設定します。

- この設定により、大きいグラフィックが必要以上に大きいサムネイルに変換されるのを防止できます。
- この設定は、サムネイル・イメージのサイズに影響のある他のすべての設定を上書きします。
- デフォルトは、0（無制限）です。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成設定の例です。

```
ThumbnailSizeLimit=100
```

THUMBNAILSIZINGMETHOD

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Inbound Refinery（5-11 ページ）

説明

サムネイルを作成する際にグラフィック・サイズを設定するための方法を決定します。

- PDF Converter では、Inbound Refinery とは異なるサムネイル・エンジンが使用されるため、PDF Converter がインストールされている場合、この設定は無効です。
- *QuickSizing* に設定すると、最も速いサイズ設定方法が使用されます。
- *SmoothSizing* に設定すると、最も正確なサイズ設定方法が使用されます。
- *GraySizing* に設定すると、グレースケール・グラフィックには最も正確なサイズ設定方法が使用され、カラー・グラフィックには最も速いサイズ設定方法が使用されます。
- デフォルトは、*SmoothSizing* です。

場所

- 「Local Inbound Refinery Configuration」画面→「Thumbnail Options」タブ→「Sizing options」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成設定の例です。

```
ThumbnailSizingMethod=QuickSizing
```

THUMBNAILTEMPLATE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

サムネイルを作成する際に HTML Export によって使用されるテンプレートのパスを定義します。

- 文字列としてパスを返します。
- デフォルトは、*<SharedDir>*/config/resources/std_thumbnail.htm です。

場所

<*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成設定の例です。

```
ThumbnailTemplate=c:/stellent/IdcRefinery/shared/config/resources/
custom_thumbnail.htm
```

THUMBNAILWIDTH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

サムネイルの幅をピクセル単位で設定します。

- デフォルトは、80 です。

場所

<*install_dir*>/config/config.cfg

注意：この変数は、Inbound Refinery のローカルな構成ファイル（<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg）に保存できますが、Inbound Refinery が読み取るためにコンテンツ・サーバーの構成ファイルにも保存する必要があります。

例

構成設定として、サムネイルの幅をピクセル数で定義します。

```
ThumbnailWidth=100
```

スクリプトとして、構成設定の値を返します。

```
<$ThumbnailWidth$>
```

TIMEOUTCHANGEDTEMPLATEFORCEDCONVERSIONSINMINS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Dynamic Converter（5-17 ページ）

説明

テンプレートまたはレイアウトが変更されたとき、Dynamic Converter がファイルを再変換するまでの最小の間隔（分）を定義します。

- 正の値に設定すると、テンプレートまたはレイアウトの変更による強制変換または事前変換の再変換は、この設定で指定した時間が経過した後にのみ実行されます。
- 別のテンプレートまたはレイアウトを使用するように変換ルールが変更された場合、通常の評価ロジックが使用されます。
- NeverTimeoutForcedConversionsTemplateChange（6-541 ページ）の設定が TRUE の場合、この設定は無視されます。
- 0 に設定されている場合は、TimeoutMismatchedForcedConversionsInMins（6-811 ページ）の設定が使用されます。
- デフォルトは、0 です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
TimeoutChangedTemplateForcedConversionsInMins=1
```

関連項目

- NeverTimeoutForcedConversionsTemplateChange（6-541 ページ）
- TimeoutMismatchedForcedConversionsInMins（6-811 ページ）

TIMEOUTMISMATCHEDFORCEDCONVERSIONSINMINS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Dynamic Converter（5-17 ページ）

説明

ソース・ドキュメント、テンプレートまたはレイアウトが変更されたとき、Dynamic Converter がファイルを再変換するまでの最小の間隔（分）を定義します。

- 正の値に設定すると、ソース・ドキュメント、テンプレートまたはレイアウトの変更による強制変換または事前変換の再変換は、この設定で指定した時間が経過した後にのみ実行されます。
- 別のテンプレートまたはレイアウトを使用するように変換ルールが変更された場合、通常の評価ロジックが使用されます。
- TimeoutChangedTemplateForcedConversionsInMins（6-810 ページ）が正の値に設定されている場合、この設定は、ソース・ドキュメントの変更のみに適用されます。
- 0 に設定すると、強制変換または事前変換は、ソース・ドキュメントの変更のために再変換されません。
- デフォルトは、15 分です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
TimeoutMismatchedForcedConversionsInMins=10
```

関連項目

- NeverTimeoutForcedConversionsTemplateChange（6-541 ページ）
- TimeoutChangedTemplateForcedConversionsInMins（6-810 ページ）

TIMEOUTPERONEMEGINSEC

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）

説明

索引コレクションへのファイルの索引付けのタイムアウトを設定します。

- 索引付けされるバルクロード・ファイルのサイズを MB 単位に切り上げ、この設定を掛け合せると、タイムアウト値（秒）が決定します。
- ファイルに多数の数値データが含まれる場合は、タイムアウトを 3000 ～ 6000 に増加します。4 ～ 5 回のバルクロードごとに発生する索引付けのクリーンアップ・サイクルもタイムアウトに加算され、数値データの場合は時間がかかる可能性があります。
- デフォルトは、1MB 当たり 600 秒です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
TimeoutPerOneMegInSec=3000
```

TOINTEGER

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 文字列（5-39 ページ）

説明

文字列を整数に変換します。

注意：文字列連結、文字列論理和および簡単な文字列比較には、特殊な文字列演算子（3-23 ページ）を使用します。

パラメータ

1 つのパラメータを取ります。

- パラメータは、文字列のみです。

出力

整数値を返します。文字列が数値に評価されないか、整数以外の数値に評価された場合、エラーがスローされます。

例

文字列 "4" を整数に変換し、4 の値を返します。

```
<$toInteger("4")$>
```

TRACE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- デバッグ（5-20 ページ）

説明

デバッグまたはトレース・メッセージの ScriptDebugTrace 出力へのロギングを有効にします。メッセージは、コンソールまたはシステム・ログに出力することもできます。

パラメータ

1 つの必須パラメータと 2 つのオプションのパラメータを取ります。

- 最初のパラメータは、出力されるメッセージです。値は次のいずれかです。
  - 文字列。文字列のみをパラメータとして渡す場合、この関数を実行するには ScriptDebugTrace（6-709 ページ）変数が設定されている必要があります。文字列は、累積されたデバッグ・トレースに追加されます。
  - `#local`。現在のすべてのローカル変数のダンプを準備します。
  - `#all`。すべてのローカル・データ、結果セットおよび環境データの完全な HDA ダンプを準備します。
- 2 番目（オプション）のパラメータは、メッセージの中継場所です。値は次のとおりです。
  - `#console`。コンソールに表示します。
  - `#log`。HTML ログ・ファイルにメッセージをログします。
  - 変数名（StatusMessage など）。この場合、メッセージは現在の値に追加されます。
- 3 番目（オプション）のパラメータは、適切なトレース・セクションです。これは、場所パラメータを `#console` に設定した場合にのみ関連があります。

例

次の例では、文字列 message は、（常にログされる）システム・コンソールにログされます。

```
<$trace("message", "#console")$>
```

次の例では、文字列 message をシステム・コンソールの pagecreation トレース・セクションにログします。

```
<$trace("message", "#console", "pagecreation")$>
```

次の例では、文字列 message を HTML コンテンツ・サーバー・ログ・ファイルにログします。

```
<$trace("message", "#log")$>
```

次の例では、すべてのローカル変数とその値をシステム・コンソールにダンプします。

```
<$trace("#local", "#console")$>
```

次の例では、すべてのローカル変数、結果セットおよび環境変数をシステム・コンソールにダンプします。

```
<$trace("#all", "#console")$>
```

次の例では、すべてのデータを変数 MyTraceDump にダンプします。このデータは、ページに表示できます。これはコンソール・ログを表示する適切なアクセス権のない HCSP 開発者に役立ちます。

```
<$trace("#all", "MyTraceDump")$>
<$MyTraceDump$>
```

関連項目

– ScriptDebugTrace（6-709 ページ）

TRACECONVERSIONCACHECLEANUP

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Dynamic Converter（5-17 ページ）

説明

Dynamic Converter のキャッシュ・クリーンアップ・サイクル中にレポートをデバッグ・トレースに出力します。

- TRUE に設定すると、削除されたドキュメント、削除されたドキュメントの変換タイプ（強制か否か）、キャッシュ・サイズの制約のためにドキュメントが削除されたかどうかの報告がデバッグ・トレースに含まれます。
- Dynamic Converter のキャッシュ・クリーンアップ・サイクルの詳細は、6-105 ページの「ConversionCacheExpirationPeriodInDays」、6-122 ページの「DatedCacheIntervalDays」および 6-516 ページの「MaxConversionCacheSizeInMegs」を参照してください。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
TraceConversionCacheCleanup=true
```

関連項目

- ConversionCacheExpirationPeriodInDays（6-105 ページ）
- DatedCacheIntervalDays（6-122 ページ）
- MaxConversionCacheSizeInMegs（6-516 ページ）

TRACEISVERBOSE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- デバッグ（5-20 ページ）

説明

この変数を TRUE に設定すると、TraceSectionsList（6-821 ページ）構成オプションによって指定されたセクションでの冗長レポートが有効になります。冗長出力をサポートしないセクションもあることに注意してください。この構成設定では、指定されたセクションが冗長出力をサポートする場合、そのセクションの冗長出力を有効にします。

- デフォルトでは、レポートは FALSE に設定されます。

場所

<*install_dir*>/config/config.cfg

例

```
TraceIsVerbose=true
```

関連項目

– TraceSectionsList（6-821 ページ）

TRACERESOURCECONFLICT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- デバッグ（5-20 ページ）

説明

コマンドラインからコンテンツ・サーバーを起動するとき、コンポーネント・リソースによって 2 回上書きされるシステム・リソースがコマンドラインに出力されます。

- TRUE に設定すると、2 つ以上のコンポーネント・リソースによって上書きされるシステム・リソースが出力されます。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

```
TraceResourceConflict=true
```

関連項目

- TraceResourceLoad（6-819 ページ）
- TraceResourceOverride（6-820 ページ）

TRACERESOURCELOAD

タイプおよび使用方法

- 構成変数（4-3 ページ）
- デバッグ（5-20 ページ）

説明

コマンドラインからコンテンツ・サーバーを起動するとき、すべてのリソース・ロード・アクティビティがコマンドラインに出力されます。

- TRUE に設定すると、ロードされたすべてのリソース、リソースの上書き、リソースの競合およびリソースのマージが出力されます。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

```
TraceResourceLoad=true
```

関連項目

– TraceResourceConflict（6-818 ページ）
– TraceResourceOverride（6-820 ページ）

TRACERESOURCEOVERRIDE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- デバッグ（5-20 ページ）

説明

コマンドラインからコンテンツ・サーバーを起動するとき、コンポーネント・リソースによって上書きされるシステム・リソースおよび非システム・コンポーネント・リソースがコマンドラインに出力されます。

- TRUE に設定すると、1 つ以上のコンポーネント・リソースによって上書きされるシステム・リソースおよび非システム・リソースが出力されます。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

```
TraceResourceOverride=true
```

関連項目

- TraceResourceConflict（6-818 ページ）
- TraceResourceLoad（6-819 ページ）

TRACESECTIONSLIST

タイプおよび使用方法

- 構成変数（4-3 ページ）
- デバッグ（5-20 ページ）

説明

この変数は、Content Server の起動時にアクティブなトレース・オプションを設定します。

- トレースは、セクション・ベースでアクティブ化できます。system セクションは、デフォルトで設定されています。
- 他のセクションは、カンマ区切りのリストで追加できます。
- コンポーネントは、独自のトレース・セクションを追加できます。
- 一部のトレース・セクションは、冗長レポートをサポートします。冗長レポートを有効にするには、TraceIsVerbose（6-817 ページ）構成オプションを使用します。

場所

<*install_dir*>/config/config.cfg

セクション

次の表は、トレースの可能なセクションのリストです。

セクション	説明
all	すべてのセクションをトレースします。
applet	一般的なアプレットのトレースを有効にします。
archiver	アーカイバ・データ・ファイルの読取りおよび書込みを含む、標準アーカイブ・アクティビティをトレースします。
archiverlocks	長いアーカイブ・アクティビティでファイルおよびアーカイブに設定されるロックをトレースします。

セクション	説明
chunkedrequest	大きいリクエストを複数の小さいリクエストに分割するプロトコルの実装のために作成されたメッセージおよびヘッダーをトレースします。
filelock	ディレクトリに対する短期間のロックをトレースします。衝突およびタイムアウトに重点がおかれます。
filelonglock	長期間のロックの作成、削除およびメンテナンスをトレースします。
filequeue	ファイル・キューへのアクセスをトレースします。
indexermonitor	索引の更新および再構築アクティビティの開始と終了をトレースします。
localization	ローカライゼーション構成アクティビティをトレースします。
mail	コンテンツ・サーバーの内部 API によって送信されたメールをトレースします。
pagecreation	Idoc スクリプト・ページの作成時のパフォーマンスおよび監査データを作成します。
scheduledevents	毎時または毎日のスケジュール済バックグラウンド・アクティビティをトレースします。
schema	カスタム・メタデータ・スキーマ情報をトレースします。
searchquery	検索コレクションに対して実行される問合せのデータおよびパフォーマンスをトレースします。
socketrequests	受信リクエストの低レベルのデータをトレースします。
system	内部メッセージをトレースします。
systemdatabase	実行された問合せを含む、データベース・アクティビティに関するデータを作成します。
transfermonitor	アーカイバのバッチファイル転送アクティビティをトレースします。

セクション	説明
userstorage	外部ユーザー・リポジトリへのアクセスでのパラメータおよび結果をトレースします。
workflow	ワークフロー内を移動するコンテンツ・アイテムのデータをトレースします。

例

```
TraceSectionsList=system, mail, pagecreation
```

関連項目

– TraceIsVerbose（6-817 ページ）

UIEHIDESEARCHCHECKBOXES

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Record Manager および Retention Manager（5-32 ページ）

説明

Records Manager 専用。

検索ページにメタデータ・フィールドのチェック・ボックスを表示するかどうかを制御します。

TRUE に設定すると、ページにメタデータ・フィールドのチェック・ボックスは表示されません。FALSE に設定すると、メタデータ・フィールドのチェック・ボックスが表示されます。

デフォルトは、TRUE です。

場所

- <*install_dir*>/custom/RecordsManagement/records_management_environments.cfg
- <*install_dir*>/custom/RetentionManager/retentionmanager_environments.cfg

例

次は構成エントリの使用例です。

```
UieHideSearchCheckboxes=false
```

UNIVERSALSEARCHALLOWFULLTEXT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）

説明

SearchQueryFormat=Universal を使用する場合、表示されるフィールドを変更します。UniversalSearchAllowFullText が FALSE に設定されている場合、「Search」ページに「Full Text Search」フィールドは表示されません。

デフォルトは、TRUE です。

場所

<*install_dir*>/config/config.cfg

例

```
UniversalSearchAllowFullText=false
```

関連項目

– SearchQueryFormat（6-720 ページ）

UPLOADAPPLET

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）
- クライアント（5-4 ページ）

説明

複数ファイルの Upload アプレットを有効にします。

重要：この設定は、Upload アプレットのカスタム・レガシー・バージョンでのみ使用されます。標準の MultiUpload（6-539 ページ）変数が有効な場合は、この設定を有効にしないでください。

- TRUE に設定すると、Upload アプレットが有効になり、複数のファイルを ZIP して単一のコンテンツ・アイテムとしてチェックインできます。
- FALSE に設定した場合、Upload アプレットは無効です。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

構成設定として、Upload アプレットを有効にします。

```
UploadApplet=true
```

スクリプトとして、Upload アプレットの状態を評価します。

```
<$UploadApplet$>
```

関連項目

- DownloadApplet（6-181 ページ）
- MultiUpload（6-539 ページ）

URL

タイプおよび使用方法

❖ グローバル関数（4-21 ページ）

❖ 文字列（5-39 ページ）

説明

URL で使用する文字列を書式設定します。

❖ この関数は、空白および予約文字をエスケープ・シーケンスに変換します。

パラメータ

1 つのパラメータを取ります。

❖ パラメータは、文字列のみです。

出力

URL で使用する書式に設定された文字列を返します。

例

文字列 "home page" を *home%20page* として書式設定します。

```
<$url("home page")$>
```

文字列 "home/page" を *home%2fpage* として書式設定します。

```
<$url("home/page")$>
```

文字列 "home?page" を *home%3fpage* として書式設定します。

```
<$url("home?page")$>
```

URLESCAPE7BIT

タイプおよび使用方法

❖ グローバル関数（4-21 ページ）

❖ 文字列（5-39 ページ）

説明

この関数は、文字列パラメータの URL エンコード・バージョンを返します。url 関数に類似していますが、この関数は、7 ビット・クリーンでない文字のみをエンコードします（ASCII）。したがって、この関数は、同じ文字列に対して繰り返しコールできます。

url 関数を使用して文字列のエンコードを繰り返し行うと、すべての % 文字が %25 にエンコードされます。

パラメータ

1 つのパラメータを取ります。

❖ パラメータは、文字列のみです。

出力

URL で使用する書式に設定された文字列を返します。

例

```
<$path = "mÿ f浜 '.txt"$>
20f浜 '.txt = <$url(path)$>
mÿ%2520f浜 '.txt = <$url(url(path))$>
m%ff%20f%efl%eb.txt = <$urlEscape7Bit(url(path))$>
m%ff%20f%efl%eb.txt= <$urlEscape7Bit(urlEscape7Bit(url(path)))$>
```

関連項目

– url（6-827 ページ）

USEACCOUNTS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 内部セキュリティ（5-36 ページ）

説明

アカウントの使用を有効にします。

- TRUE に設定すると、アカウントは有効になります。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
UseAccounts=true
```

USEADOBEDISTILLER

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

PDF Converter が Jaws PDF Library を使用せずに、Adobe Acrobat Distiller を使用するように指定します。

- TRUE に設定すると、PDF Converter は Adobe Acrobat Distiller を使用します。
- FALSE に設定すると、PDF Converter は Jaws PDF Library を使用します。
- PDF Converter のインストール時に Adobe Acrobat Distiller が検出された場合、デフォルトは TRUE です。
- PDF Converterのインストール時にAdobe Acrobat Distillerが検出されなかった場合、デフォルトはありません。

場所

<*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
UseAdobeDistiller=true
```

関連項目

– DistillerPath（6-169 ページ）

USEADSI

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

コンテンツ・サーバーの Active Directory（ADSI）認証を有効にします。

- この設定は、NtlmSecurityEnabled（6-555 ページ）が TRUE の場合にのみ有効です。
- このオプションは、IIS Web サーバーを使用する場合にのみ利用できます。
- TRUE に設定すると、Active Directory で定義されたユーザーは、ネットワーク資格証明を使用してコンテンツ・サーバーにログインできます。
- デフォルト値はありません。

場所

- 「System Properties」→「Internet」タブ→「Active Directory Security (ADSI)」オプション
- 「Admin Server」→「Internet Configuration」→「Active Directory Security」オプション
- <*install_dir*>/config/config.cfg

例

ADSI セキュリティを有効にします。

```
UseAdsi=true
```

関連項目

- NtlmSecurityEnabled（6-555 ページ）
- UseNtlm（6-848 ページ）

USEALCHEMY111

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

Image Alchemy のどのバージョンのパラメータを Inbound Refinery で構成するかを定義します（PDF ファイルのサムネイルの作成に使用）。

- TRUE に設定すると、Inbound Refinery では、Image Alchemy 1.11 のパラメータが構成されます。
- FALSE に設定すると、Inbound Refinery では、Image Alchemy の最新バージョンのパラメータが構成されます。
- デフォルトは、FALSE です。

場所

<*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
UseAlchemy111=true
```

USEAUTOCAD2000

タイプおよび使用方法

- 構成変数（4-3 ページ）
- AutoCAD（5-14 ページ）

説明

AutoCAD ファイルの変換に AutoCAD 2000 または AutoCAD 14 のどちらを使用するかを定義します。

- FALSE に設定すると、AutoCAD 14 を使用して変換します。
- TRUE に設定すると、AutoCAD 2000 を使用して変換します。
- デフォルトは、FALSE です。

場所

- 「Shared Inbound Refinery Configuration」画面→「AutoCAD Options」タブ→「Use AutoCAD 2000」
- <*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
UseAutoCad2000=true
```

USEAUTOCADMODELSPACE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- AutoCAD（5-14 ページ）

説明

AutoCAD DWF ファイルを「モデル空間」または「ペーパー空間」のどちらに描画するかを定義します。

- TRUE に設定すると、モデル空間が使用されます。
- FALSE に設定すると、ペーパー空間が使用されます。
- デフォルトは、FALSE です。

場所

- 「Shared Inbound Refinery Configuration」画面→「AutoCAD Options」タブ→「View space options」
- <*install_dir*>/IdcRefinery/shared/idcrefinery.cfg

例

次は構成エントリの使用例です。

```
UseAutocadModelSpace=true
```

USEBELLEVUELOOK

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ページ表示（5-30 ページ）

説明

コンテンツ・サーバー・インタフェースの代替ルック・アンド・フィールを有効にします。

- TRUE に設定すると、Bellevue ルック・アンド・フィール（緑と赤のカラー・スキーム）が使用されます。
- UseXpedioLook（6-892 ページ）が TRUE の場合、この設定は上書きされます。
- デフォルト値はありません。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
UseBellevueLook=true
```

関連項目

– UseXpedioLook（6-892 ページ）

USECASCADINGSTYLESHEETS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Dynamic Converter（5-17 ページ）

説明

Dynamic Converter の Cascading Style Sheet（CSS）の使用を有効にします。

- TRUE に設定すると、Cascading Style Sheet は有効になります。
- FALSE に設定すると、Cascading Style Sheet は無効になります。このオプションは、CSS をサポートしない古いブラウザでのみ有用です。
- デフォルトは、TRUE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
UseCascadingStyleSheets=false
```

USECUSTOMMODALING

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

この変数を FALSE に設定すると、モーダル・ダイアログで、全フレームをロックするかわりに、ダイアログの下のフレームのみをロックできます。この変数は、Red Hat Linux ES 3 を使用しているシステムにのみ効果があります。

たとえば、Repository Manager とアーカイバがどちらもアクティブで、Repository Manager からダイアログをオープンした場合、アーカイバは機能しません。ただし、この変数を FALSE に設定した場合、アーカイバは、他のダイアログがオープンした状態で機能し続けることができます。

- デフォルトは、TRUE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
UseCustomModaling=false
```

USEDATABASESHORTINDEXNAME

タイプおよび使用方法

- 構成変数（4-3 ページ）
- データベース（5-18 ページ）

説明

使用するデータベースが 18 文字を超える索引名をサポートしない場合に必要です。

- TRUE に設定すると、短い索引名が使用されます。
- FALSE に設定すると、短い索引名は使用されません。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
UseDatabaseShortIndexName=false
```

USEDCLAPPLET

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）
- フィールド表示（5-24 ページ）

説明

Dependent Choice List アプレットを有効にします。

- この設定は、Dependent Choice List コンポーネント（<*install_dir*>/samples/AdvancedDependentList.zip ファイル）とともに使用します。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
UseDCLApplet=true
```

USEFOURDIGITYEAR

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 日付と時間（5-19 ページ）

説明

重要： この設定は、バージョン 5.1 以降では不要になりました。SystemDateFormat（6-788 ページ）を使用します。

4 桁または 2 桁の年の表示形式を指定します。

- TRUE に設定すると、コンテンツ・サーバーで 4 桁の年が表示されます。
- FALSE に設定すると、コンテンツ・サーバーで 2 桁の年が表示されます。
- コンテンツ・サーバーは、データ・エントリとして 2 桁および 4 桁の年を両方ともサポートします。
- デフォルトは、TRUE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
UseFourDigitYear=false
```

関連項目

– SystemDateFormat（6-788 ページ）

USEFULLGROUPNAME

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

コンテンツ・サーバーのロールまたはアカウントへのマッピングに、Active Directory または LDAP グループの階層全体を含めるかどうかを指定します。

- TRUE に設定すると、グループ階層全体（指定された接頭辞またはネーミング・コンテキストまで）が、ロールまたはアカウント名に含まれます。
- FALSE に設定すると、グループ階層はロールまたはアカウント名に含まれません。
- 詳細な例は、「Active Directory Configuration」ページまたは「Add/Edit LDAP Provider」ページを参照してください。
- デフォルトは、FALSE です。

場所

Active Directory セキュリティ

- 「Active Directory Configuration」ページ→「Use Full Group Names」
- <*install_dir*>/data/users/config/filter.hda

LDAP セキュリティ

- 「Add/Edit LDAP Provider」ページ→「Use Full Group Names」
- <*install_dir*>/data/providers/*provider_name*/provider.hda

例

次は構成エントリの使用例です。

```
UseFullGroupName=true
```

USEGROUPFILTER

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

コンテンツ・サーバーのロールおよびアカウントにマッピングされる Active Directory または LDAP グループの選択に、フィルタを使用できます。

- Active Directory または LDAP セキュリティで使用されます。TRUE に設定すると、コンテンツ・サーバーのロールおよびアカウントへの Active Directory または LDAP グループのマッピングに、RolePrefix（6-657 ページ）および AcctPrefix（6-7 ページ）の設定が使用されます。
- デフォルトは、FALSE です。

場所

Active Directory セキュリティ

- 「Active Directory Configuration」ページ→「Group Filtering」
- <*install_dir*>/data/users/config/filter.hda

LDAP セキュリティ

- 「Add/Edit LDAP Provider」ページ→「Group Filtering」
- <*install_dir*>/data/providers/*ldap_provider_name*/provider.hda

例

次は構成エントリの使用例です。

```
UseGroupFilter=true
```

関連項目

- AcctPrefix（6-7 ページ）
- RolePrefix（6-657 ページ）

USEHTMLORTEXTHIGHLIGHTINFO

タイプおよび使用方法

- 値変数（4-34 ページ）
- 検索（5-34 ページ）

説明

フルテキスト検索のキーワードの強調表示が有効かどうか、およびファイル形式が強調表示をサポートするかどうか（PDF、HTM、TXT など）をチェックします。

- EnableDocumentHighlight（6-193 ページ）が TRUE の場合、キーワードの強調表示は有効です。

出力

- 強調表示が有効な場合は、TRUE を返します。
- 強調表示が有効でない場合は、FALSE を返します。

例

検索キーワードの強調表示ステータスを返します。

```
<$UseHtmlOrTextHighlightInfo$>
```

関連項目

- UseXmlUrl（6-891 ページ）
- EnableDocumentHighlight（6-193 ページ）

USELIGHTWEIGHTTOPOSTSCRIPTSTEP

タイプおよび使用方法

- 構成変数（4-3 ページ）
- PDF Converter（5-12 ページ）

説明

Inbound Refinery で、PostScript への変換のために、Outside In または Adobe Acrobat Distiller のどちらを使用するかを定義します。

- TRUE に設定すると、Inbound Refinery は、PostScript への変換に Outside In を使用します。
- TRUE に設定すると、Inbound Refinery は、PostScript への変換に Adobe Acrobat Distiller を使用します。
- デフォルトは、FALSE です。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
UseLightweightToPostscriptStep=true
```

USELOCALGROUPS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

ローカル・グループの Windows ネットワーク・ユーザーに、コンテンツ・サーバー権限を付与するかどうかを定義します。

- NTLM セキュリティで使用します。TRUE の場合、LocalGroupServer（6-505 ページ）設定で指定された Windows Server 上のローカル・グループは、デフォルト・マスター・ドメインのグローバル（ドメイン）グループであるかのようにロールおよびアカウントにマップされます。
- デフォルトは、FALSE です。

場所

- 「Configure NT Domain Security」ページ→「Use Local Groups」
- <*install_dir*>/data/users/config/filter.hda

例

次は構成エントリの使用例です。

```
UseLocalGroups=true
```

関連項目

– LocalGroupServer（6-505 ページ）

USEMICROSOFTVM

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

Microsoft の Java 仮想マシン（JVM）の使用を指定します。

- TRUE に設定すると、Microsoft の JVM が使用されます。
- FALSE に設定すると、Sun の JVM が使用されます。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
UseMicrosoftVM=true
```

USENETSCAPE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

プロバイダを通じて LDAP サーバーへの接続に使用する方法を指定します。

- TRUE に設定すると、Netscape SDK が使用されます。これは優れたパフォーマンスのために推奨される設定です。
- FALSE に設定すると、JNDI 接続が使用されます。
- デフォルトは、TRUE です。

場所

- 「Add/Edit LDAP Provider」ページ→「Use Netscape SDK」
- <*install_dir*>/data/providers/*provider_name*/provider.hda

例

次は構成エントリの使用例です。

```
UseNetscape=0
```

USENTLM

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

コンテンツ・サーバーの NT LAN Manager（NTLM）認証を有効にします。

- この設定は、NtlmSecurityEnabled（6-555 ページ）が TRUE の場合にのみ有効です。
- このオプションは、IIS Web サーバーを使用する場合にのみ利用できます。
- TRUE に設定すると、Windows ネットワーク（NTLM）で定義されたユーザーは、ネットワーク資格証明を使用してコンテンツ・サーバーにログインできます。
- デフォルト値はありません。

場所

- 「System Properties」→「Internet」タブ→「NT Domain Security (NTLM)」オプション
- 「Admin Server」→「Internet Configuration」→「NT Domain Security」オプション
- <*install_dir*>/config/config.cfg

例

NTLM セキュリティを有効にします。

```
UseNtlm=true
```

関連項目

- NtlmSecurityEnabled（6-555 ページ）
- UseAdsi（6-831 ページ）

USERACCOUNTS

タイプおよび使用方法

- ユーザー読取り専用変数（4-33 ページ）
- 内部セキュリティ（5-36 ページ）
- ユーザー（5-40 ページ）

説明

現在のユーザーが割り当てられているアカウントのカンマ区切りのリストを取得します。

- #none エントリは、アカウントが割り当てられていないコンテンツ・アイテムに対する権限を示します。
- #all エントリは、すべてのアカウントに対する権限を示します。

出力

アカウント・リストを文字列で返します。

例

次に例を示します。

```
UserAccounts=BOS,SEA,MSP/Gen
```

このマークアップは、ページにユーザー変数を表示します。

```
<$if UserName$>
    Logon Name: <$UserName$><BR>
    User Name: <$UserFullName$><BR>
    E-mail Address: <$UserAddress$><BR>
    Accounts: <$UserAccounts$><BR>
<$endif$>
```

USERADDRESS

タイプおよび使用方法

- ユーザー読取り専用変数（4-33 ページ）
- ユーザー（5-40 ページ）

説明

現在のユーザーの電子メール・アドレスを取得します。

出力

電子メール・アドレスを文字列で返します。

例

このマークアップは、ページにユーザー変数を表示します。

```
<$if UserName$>
    Logon Name: <$UserName$><BR>
    User Name: <$UserFullName$><BR>
    E-mail Address: <$UserAddress$><BR>
    Default Account: <$UserDefaultAccount$><BR>
<$endif$>
```

USERAPPRIGHTS

タイプおよび使用方法

- ユーザー読取り専用変数（4-33 ページ）
- 内部セキュリティ（5-36 ページ）
- ユーザー（5-40 ページ）

説明

現在のユーザーのアプリケーション権限をチェックします。

出力

- ユーザーの権限を指定するビット・フラグ数値を返します。

管理アプリケーション	ビット・フラグ
ユーザー管理	1
Web レイアウト・エディタ	2
Repository Manager	4
ワークフロー管理	8
Configuration Manager	16
アーカイバ	32

例

現在のユーザーのアプリケーション権限を表示します。

```
<$UserAppRights$>
```

たとえば、次のように組み合せると、次の数値になります。

- ユーザー管理 + Web レイアウト・エディタ = 3
- Repository Manager + ワークフロー管理 + アーカイバ = 44
- 全 6 個のアプリケーション = 63

関連項目

– AdminAtLeastOneGroup（6-16 ページ）
– UserIsAdmin（6-862 ページ）

USERCACHE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

users ディレクトリのパスを定義します。これは、ユーザーのパーソナライズ情報の格納場所です。

- デフォルトは、<*install_dir*>*/data/users/* です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
UserCache=c:/myuserdata/
```

USERCACHETIMEOUT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）
- ユーザー（5-40 ページ）

説明

グローバルおよび外部ユーザー情報の一時キャッシュのタイムアウトをミリ秒で設定します。

- デフォルトは、60000 ミリ秒（1 分）です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
UserCacheTimeout=120000
```

USERDEFAULTACCOUNT

タイプおよび使用方法

- ユーザー読取り専用変数（4-33 ページ）
- 内部セキュリティ（5-36 ページ）
- ユーザー（5-40 ページ）

説明

現在のユーザーのデフォルト・アカウントを取得します。

出力

アカウント名を文字列で返します。

例

このマークアップは、ページにユーザー変数を表示します。

```
<$if UserName$>
   Logon Name: <$UserName$><BR>
   User Name: <$UserFullName$><BR>
   E-mail Address: <$UserAddress$><BR>
   Default Account: <$UserDefaultAccount$><BR>
<$endif$>
```

USEREDIRECTEDOUTPUT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- デバッグ（5-20 ページ）

説明

Java 仮想マシンからの出力を取得できます。これには、有効なトレース機能からのログ

注意 : Windows システムで、コンテンツ・サーバーがサービスとして実行されている場合にのみ使用します。

出力および VM クラッシュから生じたスタック・ダンプが含まれます。

場所

<*install_dir*>/config/config.cfg

例

```
UseRedirectedOutput=true
```

USEREVISIONCREATEDATE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

アーカイブをインポートするシステムにこの変数を設定すると、アーカイバは、dCreateDate を使用して比較を行います。

- これにより、Content Server は、dInDate の割当てではなく、最初の作成日によってリビジョンを区別できます。
- これはデフォルトでは設定されません。

重要：この構成設定は、Content Server の古いバージョン（7.5 未満）からのレガシー・コンテンツがインポートされていない場合に使用します。

場所

<*install_dir*>/config/config.cfg

例

```
UseRevisionCreateDate=true
```

USERFILTEREDSUBJECTS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 内部セキュリティ（5-36 ページ）

説明

現在のユーザーのセキュリティ資格証明が変更された場合にリフレッシュする必要があるサブジェクトを指定します。

- サブジェクトは、コンテンツ・サーバー・サービスによって、オプション・リストの変更や新規ユーザーの追加など、システムに変更が加えられたときに通知されるサブシステムです。
- この設定に、任意のカスタム・セキュリティ・サブジェクトを追加します。通常、カスタム・サブジェクトは、非常に複雑なセキュリティ統合でのみ作成されます。
- これはカンマ区切りのリストです。
- デフォルトは、*users,accounts* です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
UserFilteredSubjects=users,accounts,licenses
```

USERFULLNAME

タイプおよび使用方法

- ユーザー読取り専用変数（4-33 ページ）
- ユーザー（5-40 ページ）

説明

現在のユーザーの完全名。

出力

- ユーザーの完全名を文字列で返します。
- ユーザーがログインしていない場合は、文字列 anonymous を返します。

例

このマークアップは、ページにユーザー変数を表示します。

```
<$if UserName$>
   Logon Name: <$UserName$><BR>
   User Name: <$UserFullName$><BR>
   E-mail Address: <$UserAddress$><BR>
   Default Account: <$UserDefaultAccount$><BR>
<$endif$>
```

USERHASACCESSTOACCOUNT

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 内部セキュリティ（5-36 ページ）
- ユーザー（5-40 ページ）

説明

この関数は、ユーザーが特定の権限レベルで指定されたアカウントにアクセスできる場合は TRUE を返します。権限は、アクセス・レベルを 1 文字で表現したものです。次に例を示します。

- R: 読取り。R または 1 で指定します。
- W: 書込み。W または 2 で指定します。
- D: 削除。D または 4 で指定します。
- A: 管理。A または 8 で指定します。

アルファベットで指定する場合は、引用符で囲む必要があります。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、アカウント名です。
- 2 番目のパラメータは、チェックするアクセス・レベルです。

出力

- 現在のユーザーが、指定したアクセス権を持つユーザーの場合は、TRUE を返します。
- 指定したアクセス権のないユーザーの場合は、FALSE を返します。

例

指定されたアカウントに対する読取りアクセス権のあるユーザーかどうかを評価します。次の 2 つの例は、同じ結果を返します。

```
<$userHasAccessToAccount("profile_account", "R")$>
<$userHasAccessToAccount("profile_account", 1)$>
```

USERHASGROUPPRIVILEGE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 内部セキュリティ（5-36 ページ）
- ユーザー（5-40 ページ）

説明

この関数は、指定されたグループに対する指定された権限を持つユーザーの場合は TRUE を返します。権限は、アクセス・レベルを 1 文字で表現したものです。次に例を示します。

- R: 読取り
- W: 書込み
- D: 削除
- A: 管理

パラメータ

パラメータを 2 つ取ります。

- チェックするセキュリティ・グループ
- チェックするアクセス・レベル

出力

- 現在のユーザーが、指定したアクセス権を持つユーザーの場合は、TRUE を返します。
- 指定したアクセス権のないユーザーの場合は、FALSE を返します。

例

指定されたロールを持つユーザーかどうかを評価します。

```
<$userHasGroupPrivilege("Public", "R")$>
<$userHasGroupPrivilege("Secure", "A")$>
```

USERHASROLE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 内部セキュリティ（5-36 ページ）
- ユーザー（5-40 ページ）

説明

特定のロールを現在のユーザーが持つかどうかをチェックします。

パラメータ

1 つのパラメータを取ります。

- パラメータは、ロールの名前のみです。

出力

- 現在のユーザーが指定したロールを持たない場合は、TRUE を返します。
- ユーザーが指定したロールを持たない場合は、FALSE を返します。

例

指定されたロールを持つユーザーかどうかを評価します。

```
<$userHasRole("admin")$>
```

USERISADMIN

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- 内部セキュリティ（5-36 ページ）
- ユーザー（5-40 ページ）

説明

現在のユーザーが、管理アプレットに対する権限を持つかどうかを確認します。

出力

- ユーザーに管理権限がある場合は、TRUE を返します。
- ユーザーに管理権限がない場合は、FALSE を返します。

例

ユーザーの管理権限を評価します。

```
<$UserIsAdmin$>
```

関連項目

- AdminAtLeastOneGroup（6-16 ページ）
- UserAppRights（6-851 ページ）

USERLANGUAGEID

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- ローカライゼーション（5-29 ページ）

説明

ユーザーの優先言語を表す 2 文字のコードを返します。英語の場合は `en`、フランス語は `fr`、日本語は `ja` などです。

ローカライズされたコンテンツの URL を構成する場合に役立ちます。

出力

優先言語のコードを返します。

USERLOCALEID

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- ローカライゼーション（5-29 ページ）

説明

ユーザーのロケールの完全名を返します。English-US、English-UK、Japanese などです。

ロケールには、言語、日付および数字の書式に関する情報が含まれます。

出力

優先言語の名前を返します。

USERNAME

タイプおよび使用方法

- ユーザー読取り専用変数（4-33 ページ）
- ユーザー（5-40 ページ）

説明

現在のユーザーのユーザー名を取得します。

出力

- ユーザー名を文字列で返します。
- ユーザーがログインしていない場合は、文字列 anonymous になります。

例

このマークアップは、ページにユーザー変数を表示します。

```
<$if UserName$> Logon Name: <$UserName$><BR>
   User Name: <$UserFullName$><BR>
   E-mail Address: <$UserAddress$><BR>
   Default Account: <$UserDefaultAccount$><BR>
<$endif$>
```

USERPUBLISHCACHEDIR

タイプおよび使用方法

- 構成変数（4-3 ページ）
- クラスタリング（5-5 ページ）

説明

Web サーバー・プラグインが、コンテンツ・サーバーのインストール先のファイル・システムではなく、ローカル・ディレクトリ（スタブ・ディレクトリ）内のデータを参照できるようにします。

- この設定は、クラスタ Windows コンテンツ・サーバー構成で使用されます。詳細は、クラスタリングのドキュメントを参照してください。
- 使用する書式は、`UserPublishCacheDir=<stub_dir>/data/users` です。ここで、*<stub_dir>* は、スタブ・ディレクトリのパスです（`f:/stellent/` など）。

場所

<install_dir>/config/config.cfg

例

次は構成エントリの使用例です。

```
UserPublishCacheDir=f:/stellent/data/users
```

USERROLES

タイプおよび使用方法

- ユーザー読取り専用変数（4-33 ページ）
- 内部セキュリティ（5-36 ページ）
- ユーザー（5-40 ページ）

説明

現在のユーザーが所属するロールのカンマ区切りのリストを取得します。

出力

ユーザーのロール・リストを文字列で返します。

例

次に例を示します。

```
PublicContributor,ClassifiedConsumer
```

ユーザー・ロールのリストを参照します。

```
<$if UserRoles$>
    <$include optional_field$>
<$endif$>
```

USERSELECTAPPLETCOLUMNOPTION

タイプおよび使用方法

❖ Collaboration Manager の構成変数（4-4 ページ）

❖ Collaboration Manager（5-5 ページ）

説明

Select User アプレットのユーザーおよびグループ列の表示およびソート順序を指定します。

❖ Collaboration Manager のインストール中に追加されます。

場所

<*install_dir*>/<*instancename*>/data/components/CollaborationManager/config.cfg

例

```
UserSelectAppletColumnOption=a
```

関連項目

–『Stellent Collaboration Manager System Administration Guide』

USERSELECTAPPLETDATABASECASESENSITIVE

タイプおよび使用方法

- Collaboration Manager の構成変数（4-4 ページ）
- Collaboration Manager（5-5 ページ）

説明

設定すると、データベースの大文字・小文字の区別を有効にします。

- 設定した場合、Select User アプレットのマークアップに、次のように追加されます。
  `IsDatabaseCaseSentivite = true`
- これは、システム・データベース構成エントリ `DatabasePreserveCase` が事前に設定されている場合に追加されます。
- Collaboration Manager のインストール中に追加されます。

場所

<*install_dir*>/<*instancename*>/data/components/CollaborationManager/config.cfg

例

```
UserSelectAppletDatabaseCaseSensitive=true
```

USERSELECTAPPLETDATABASEISO8859DICTIONARYORDER

タイプおよび使用方法

- Collaboration Manager の構成変数（4-4 ページ）
- Collaboration Manager（5-5 ページ）

説明

設定すると、ISO 8859 キャラクタ・セットの照合順序が有効になります。

- 設定した場合、Select User アプレットのマークアップに、次のように追加されます。
  `IsDatabaseIso8859DictionaryOrder = 1`
- この変数は、`IsLowerCaseCompare` アプレット・パラメータが true に設定され、`IsDatabaseCaseSensitive` アプレット・パラメータが false または未設定の場合、デフォルトで 1 に設定されます。
- SQL Server を使用し、ISO 8859（または CP1252）キャラクタ・セットを使用する場合で、照合順序がデフォルトから変更されていなければ、この変数を設定します。
- Collaboration Manager のインストール中に追加されます。

場所

<install_dir>/*<instancename>*/data/components/CollaborationManager/config.cfg

例

```
UserSelectAppletDatabaseIso8859DictionaryOrder=1
```

USERSELECTAPPLETEXTERNALALIASDEFAULTPERMISSIONS

タイプおよび使用方法

❖ Collaboration Manager の構成変数（4-4 ページ）

❖ Collaboration Manager（5-5 ページ）

説明

外部の別名（グループ）に対するデフォルトの権限を指定します。

❖ 次の数値を設定できます。

構成値	権限
1	読取り
3	読取り、書込み
7	読取り、書込み、削除
15	読取り、書込み、削除、管理

❖ 明示的に設定されていない場合、デフォルトは 1 です。

場所

<install_dir>/*<instancename>*/data/components/CollaborationManager/config.cfg

例

```
UserSelectAppletExternalAliasDefaultPermissions=3
```

関連項目

– UserSelectAppletExternalUserDefaultPermissions（6-872 ページ）

– UserSelectAppletInternalAliasDefaultPermissions（6-874 ページ）

– UserSelectAppletInternalUserDefaultPermissions（6-875 ページ）

USERSELECTAPPLETEXTERNALUSERDEFAULTPERMISSIONS

タイプおよび使用方法

❖ Collaboration Manager の構成変数（4-4 ページ）

❖ Collaboration Manager（5-5 ページ）

説明

外部ユーザーに対するデフォルトの権限を指定します。

❖ 次の数値を設定できます。

構成値	権限
1	読取り
3	読取り、書込み
7	読取り、書込み、削除
15	読取り、書込み、削除、管理

❖ 明示的に設定されていない場合、デフォルトは 1 です。

場所

<install_dir>/*<instancename>*/data/components/CollaborationManager/config.cfg

例

```
UserSelectAppletExternalUserDefaultPermissions=3
```

関連項目

– UserSelectAppletExternalAliasDefaultPermissions（6-871 ページ）

– UserSelectAppletInternalAliasDefaultPermissions（6-874 ページ）

– UserSelectAppletInternalUserDefaultPermissions（6-875 ページ）

USERSELECTAPPLETFORCEENTRIESTOLOWERCASE

タイプおよび使用方法

- Collaboration Manager の構成変数（4-4 ページ）
- Collaboration Manager（5-5 ページ）

説明

設定すると、Select User アプレットのマークアップで、エントリを小文字にします。

- 設定した場合、Select User アプレットのマークアップに、次のように追加されます。
  `ForceEntriesToLowerCase = 1`
- この変数は、データベースのセキュリティで `where` 句のすべての ACL 部分がすべて小文字を使用するように Content Server または Collaboration Manager が変更されている場合にのみ設定します。
- 設定した場合、内部 ACL チェックは、大文字・小文字を区別する方法で実行されることを確認します。
- Collaboration Manager のインストール中に追加されます。

場所

<*install_dir*>/<*instancename*>/data/components/CollaborationManager/config.cfg

例

```
UserSelectAppletForceEntriesToLowerCase=1
```

USERSELECTAPPLETINTERNALALIASDEFAULTPERMISSIONS

タイプおよび使用方法

❖ Collaboration Manager の構成変数（4-4 ページ）

❖ Collaboration Manager（5-5 ページ）

説明

外部ユーザーに対するデフォルトの権限を指定します。

❖ 次の数値を設定できます。

構成値	権限
1	読取り
3	読取り、書込み
7	読取り、書込み、削除
15	読取り、書込み、削除、管理

❖ 明示的に設定されていない場合、デフォルトは 3 です。

場所

<*install_dir*>/<*instancename*>/data/components/CollaborationManager/config.cfg

例

```
UserSelectAppletExternalUserDefaultPermissions=15
```

関連項目

– UserSelectAppletExternalAliasDefaultPermissions（6-871 ページ）
– UserSelectAppletExternalUserDefaultPermissions（6-872 ページ）
– UserSelectAppletInternalUserDefaultPermissions（6-875 ページ）

USERSELECTAPPLETINTERNALUSERDEFAULTPERMISSIONS

タイプおよび使用方法

❖ Collaboration Manager の構成変数（4-4 ページ）

❖ Collaboration Manager（5-5 ページ）

説明

外部ユーザーに対するデフォルトの権限を指定します。

❖ 次の数値を設定できます。

構成値	権限
1	読取り
3	読取り、書込み
7	読取り、書込み、削除
15	読取り、書込み、削除、管理

❖ 明示的に設定されていない場合、デフォルトは 3 です。

場所

<install_dir>/*<instancename>*/data/components/CollaborationManager/config.cfg

例

```
UserSelectAppletExternalUserDefaultPermissions=15
```

関連項目

– UserSelectAppletExternalAliasDefaultPermissions（6-871 ページ）
– UserSelectAppletExternalUserDefaultPermissions（6-872 ページ）
– UserSelectAppletInternalAliasDefaultPermissions（6-874 ページ）

USESEARCHCACHE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）

説明

TRUE に設定すると、検索キャッシュが有効になります。

- 検索キャッシュでは、最近の検索エンジンの問合せをキャッシュして、パフォーマンスが向上します。
- デフォルトは、TRUE です。

場所

<*install_dir*>/config/config.cfg

例

構成変数として使用します。

```
UseSearchCache=true
```

この変数を FALSE に設定し、CGI の問合せを行うブラウザ URL リクエストの行で使用すると、毎回問合せが実行されます。これにより、実行される問合せの処理速度の計算、およびキャッシュ結果の検証が可能になります（新しい検索結果が毎回生成されるため）。

```
http://...?IdcService=GET_SEARCH_RESULTS&...UseSearchCache=false
```

USESEARCHML

タイプおよび使用方法

- 構成変数（4-3 ページ）
- XML Converter（5-16 ページ）

説明

XML 変換に SearchML XML スキーマを使用します。

- TRUE に設定すると、XML への変換に SearchML XML スキーマが使用されます。UseXmlSchema（6-889 ページ）設定は、FALSE である必要があります。
- FALSE に設定し、UseXmlSchema（6-889 ページ）設定が TRUE の場合、XML への変換に FlexionDoc スキーマが使用されます。
- デフォルトは、FALSE です。

場所

- 「Local Inbound Refinery Configuration」画面→「XML Options」タブ→「Use SearchML Schema」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
UseSearchML=true
```

関連項目

- UseXmlSchema（6-889 ページ）
- UseXmlEmbeddings（6-884 ページ）
- UseXmlExtensibleDTD（6-885 ページ）
- UseXmlNoDTD（6-886 ページ）
- UseXmlNoReference（6-887 ページ）
- UseXmlPStyleNames（6-888 ページ）
- UseXmlSchemaDTD（6-890 ページ）

USESECURELDAP

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

LDAP プロバイダで SSL を使用するかどうかを指定します。

- TRUE に設定すると、SSL が使用されます。LDAP サーバーに適切な資格証明がインストールされている必要があります。SSL が起動されると、資格証明により、LDAP サーバーとコンテンツ・サーバー間の通信が保護されます。
- FALSE に設定すると、SSL は使用されません。
- デフォルトは、FALSE です。

場所

- 「Add/Edit LDAP Provider」ページ→「Use SSL」
- <*install_dir*>/data/providers/*provider_name*/provider.hda

例

次は構成エントリの使用例です。

```
UseSecureLdap=true
```

関連項目

– LdapPort（6-477 ページ）

USESELFREGISTRATION

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 内部セキュリティ（5-36 ページ）
- ユーザー（5-40 ページ）

説明

ユーザーが独自のユーザー・プロファイルを作成できるようにします。

- 自己登録ユーザーは、グローバル・ユーザーです。
- SelfRegisteredRoles（6-726 ページ）および SelfRegisteredAccounts（6-725 ページ）を使用して、自己登録ユーザーのデフォルトのロールおよびアカウントを設定できます。
- TRUE に設定すると、ポータル・ナビゲーション・バーの「Login」リンクの下に、「Self-Registration」リンクが表示されます。（リンクを表示するには、Web レイアウト・エディタでポータル・ページを更新する必要があります。）
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成設定の例です。

```
UseSelfRegistration=true
```

スクリプトとして、構成設定の値を返します。

```
<$UseSelfRegistration$>
```

関連項目

- SelfRegisteredAccounts（6-725 ページ）
- SelfRegisteredRoles（6-726 ページ）

USESHORTNAMESALWAYS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）

説明

Web サーバー・フィルタにより、Active Directory のユーザー名から DOMAIN_NAME¥ 接頭辞を削除するかどうかを指定します。

- TRUE に設定すると、デフォルトのマスター・ドメインがドメイン名と同じであると推測して、Active Directory のユーザー名から DOMAIN_NAME¥ 接頭辞を削除します。
- FALSE に設定すると、ユーザー名に DOMAIN_NAME¥ 接頭辞が含まれます。
- デフォルトは、FALSE です。

場所

- 「Active Directory Configuration」ページ→「Use Short Names」
- <*install_dir*>/data/users/config/filter.hda

例

次は構成エントリの使用例です。

```
UseShortNamesAlways=true
```

UseSSL

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 外部セキュリティ（5-37 ページ）
- Web サーバー（5-42 ページ）

説明

Secure Sockets Layer（SSL）を有効にします。

- この設定は、変数 HttpWebRoot（6-325 ページ）および HttpCgiPath（6-312 ページ）に影響を与えます。
- Secure Sockets Layer は、SSL に対応した Web サーバーを使用する場合にのみ使用します。
- TRUE に設定されている場合、SSL が使用されます（*http* ではなく、*https* を使用）。
- デフォルトは、FALSE です。

場所

- 「System Properties」→「Internet」タブ→「Use Secure Sockets Layer」
- <*install_dir*>/config/config.cfg

例

構成設定として、SSL を有効にします。

```
UseSSL=true
```

スクリプトとして、構成設定の値を返します。

```
<$UseSSL$>
```

関連項目

- HttpCgiPath（6-312 ページ）
- HttpWebRoot（6-325 ページ）

USEUSERSELECTAPPLET

タイプおよび使用方法

- Collaboration Manager の構成変数（4-4 ページ）
- Collaboration Manager（5-5 ページ）

説明

Select User アプレットを有効にするかどうかを指定します。

- Collaboration Manager のインストール中に追加されます。
- TRUE に設定すると、Select User アプレットをアクティブにすることができます。
- デフォルトは、TRUE です。

場所

<*install_dir*>/<*instancename*>/custom/CollaborationManager/environment.cfg

例

次は構成エントリの使用例です。

```
UseUserSelectApplet=true
```

関連項目

–『Stellent Collaboration Server System Administration Guide』

USEVDK4CUSTOMSTYLE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）

説明

デフォルトの <*install_dir*>/shared/search/style/basic/ ディレクトリのかわりに、<*install_dir*>/search/vdk4_custom_style/ ディレクトリの Verity Integration スタイル・ファイルを使用します。Verity Integration は、Content Server とともに使用できるアドオン製品であり、購入することができます。

- Verity スタイル・ファイルをカスタマイズする前に、デフォルトのファイルを <*install_dir*>/shared/search/style/basic/ ディレクトリから <*install_dir*>/search/vdk4_custom_style/ ディレクトリにコピーします。すべてのカスタマイズは、/vdk4_custom_style/ ディレクトリで行います。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
UseVdk4CustomStyle=true
```

USEXMLEMBEDDINGS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- XML Converter（5-16 ページ）

説明

SearchML スキーマを使用する場合、XML への変換処理に埋込みを含めます。

- この設定は、UseSearchML（6-877 ページ）を TRUE に設定した場合にのみ有効です。
- TRUE に設定すると、XML への変換中に埋込みが含まれます。
- FALSE に設定すると、XML への変換中に埋込みは含まれません。
- デフォルトは、FALSE です。

場所

- 「Local Inbound Refinery Configuration」画面→「XML Options」タブ→「Include embeddings in the SearchML processing」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
UseXmlEmbeddings=true
```

関連項目

– UseSearchML（6-877 ページ）

USEXMLEXTENSIBLEDTD

タイプおよび使用方法

- 構成変数（4-3 ページ）
- XML Converter（5-16 ページ）

説明

SearchML スキーマを使用する場合、XML 出力に XML スキーマ定義（XSD）への参照を作成します。

- この設定は、UseSearchML（6-877 ページ）を TRUE に設定した場合にのみ有効です。
- TRUE に設定すると、*searchml.xsd* を参照するタグが XML 出力に含まれます。
- FALSE に設定すると、XML スキーマ定義の参照は XML 出力に含まれません。
- デフォルトは、FALSE です。

場所

- 「Local Inbound Refinery Configuration」画面→「XML Options」タブ→「Use a reference to an XML Schema in SearchML output」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
UseXmlExtensibleDTD=true
```

関連項目

- UseSearchML（6-877 ページ）
- UseXmlSchemaDTD（6-890 ページ）
- UseXmlNoDTD（6-886 ページ）

UseXmlNoDTD

タイプおよび使用方法

- 構成変数（4-3 ページ）
- XML Converter（5-16 ページ）

説明

SearchML スキーマを使用する場合、Document Type Definition（DTD）参照を XML 出力から削除します。

- この設定は、UseSearchML（6-877 ページ）を TRUE に設定した場合にのみ有効です。
- TRUE に設定すると、定義の参照は XML 出力に含まれません。
- FALSE に設定すると、定義の参照は XML 出力に残されます。
- デフォルトは、FALSE です。

場所

- 「Local Inbound Refinery Configuration」画面→「XML Options」タブ→「Remove any definition reference in SearchML output」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
UseXmlNoDTD=true
```

関連項目

- UseSearchML（6-877 ページ）
- UseXmlSchemaDTD（6-890 ページ）
- UseXmlExtensibleDTD（6-885 ページ）

USEXMLNOREFERENCE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- XML Converter（5-16 ページ）

説明

SearchML スキーマを使用する場合、XML 宣言の作成を抑止します。

- この設定を使用して、別の XML ファイルに含めることができる XML セグメントを生成できます。
- この設定は、UseSearchML（6-877 ページ）を TRUE に設定した場合にのみ有効です。
- TRUE に設定すると、XML 宣言が XML 出力に含まれます。
- FALSE に設定すると、XML 宣言は XML 出力に含まれません。
- デフォルトは、FALSE です。

場所

- 「Local Inbound Refinery Configuration」画面→「XML Options」タブ→「Suppress the creation of the XML Declaration」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
UseXmlNoReference=true
```

関連項目

– UseSearchML（6-877 ページ）

USEXMLPSTYLENAMES

タイプおよび使用方法

- 構成変数（4-3 ページ）
- XML Converter（5-16 ページ）

説明

SearchML を使用する場合、XML 出力に段落スタイル情報が含まれます。

- この設定は、UseSearchML（6-877 ページ）を TRUE に設定した場合にのみ有効です。
- TRUE に設定すると、段落スタイル情報が XML 出力に含まれます。
- FALSE に設定すると、段落スタイル情報は XML 出力に含まれません。
- デフォルトは、FALSE です。

場所

- 「Local Inbound Refinery Configuration」画面→「XML Options」タブ→「Include paragraph style information in generated SearchML output」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
UseXmlPStyleNames=true
```

関連項目

– UseSearchML（6-877 ページ）

USEXMLSCHEMA

タイプおよび使用方法

- 構成変数（4-3 ページ）
- XML Converter（5-16 ページ）

説明

XML への変換に FlexionDoc XML スキーマを使用します。

- TRUE に設定すると、XML への変換に FlexionDoc XML スキーマが使用されます。UseSearchML（6-877 ページ）設定は、FALSE である必要があります。
- FALSE に設定し、UseSearchML（6-877 ページ）設定が TRUE の場合、XML への変換に SearchML スキーマが使用されます。
- デフォルトは、TRUE です。

場所

- 「Local Inbound Refinery Configuration」画面→「XML Options」タブ→「Use FlexionDoc XML Schema」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
UseXMLSchema=false
```

関連項目

- UseSearchML（6-877 ページ）
- IncludeXmlFlexionDocType（6-349 ページ）

USEXMLSCHEMADTD

タイプおよび使用方法

- 構成変数（4-3 ページ）
- XML Converter（5-16 ページ）

説明

SearchML スキーマを使用する場合、Document Type Definition（DTD）を参照する DOCTYPE タグが XML 出力に作成されます。

- この設定は、UseSearchML（6-877 ページ）を TRUE に設定した場合にのみ有効です。
- TRUE に設定すると、*searchml.dtd* を参照する DOCTYPE タグが XML 出力に含まれます。
- FALSE に設定すると、XML 出力に DOCTYPE タグは含まれません。
- デフォルトは FALSE です。

場所

- 「Local Inbound Refinery Configuration」画面→「XML Options」タブ→「Include a DOCTYPE reference in SearchML output」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
UseXmlSchemaDTD=false
```

関連項目

- UseSearchML（6-877 ページ）
- UseXmlExtensibleDTD（6-885 ページ）
- UseXmlNoDTD（6-886 ページ）

USEXMLURL

タイプおよび使用方法

- 条件付き動的変数（4-2 ページ）
- 検索（5-34 ページ）

説明

XML ドキュメントの検索キーワードの強調表示が有効かどうかをチェックします。

- TRUE に設定すると、キーワード検索を実行するときに XML URL が作成されます。

出力

- XML キーワードの強調表示が有効な場合は、TRUE を返します。
- XML キーワードの強調表示が有効でない場合は、FALSE を返します。

例

XML ドキュメントの検索キーワードの強調表示のステータスを返します。

```
<$UseXmlUrl$>
```

関連項目

- UseHtmlOrTextHighlightInfo（6-843 ページ）
- EnableDocumentHighlight（6-193 ページ）

USEXPEDIOLOOK

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ページ表示（5-30 ページ）

説明

コンテンツ・サーバー・インタフェースの代替ルック・アンド・フィールを有効にします。

- TRUE に設定すると、Xpedio ルック・アンド・フィール（紫と青のカラー・スキーム）が使用されます。
- この設定が TRUE の場合、UseBellevueLook（6-835 ページ）設定は上書きされます。
- デフォルト値はありません。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
UseXpedioLook=true
```

関連項目

– UseBellevueLook（6-835 ページ）

UTGETVALUE

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- パーソナライズ（5-31 ページ）

説明

指定されたパーソナライズ（ユーザー・トピック）ファイルからユーザー・プロパティ値を返します。

- ユーザー・トピック・ファイルは、<*install_dir*>/data/users/profiles/*us*/*username*/ディレクトリにある HDA ファイルです。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、ユーザー・トピック HDA ファイルの名前です。
- 2 番目のパラメータは、ユーザー・トピック・ファイルに定義されたプロパティの名前です。

出力

指定されたユーザー・トピック・ファイル内のプロパティ値を返します。

例

現在のユーザーの *pne_portal.hda* ファイルの *emailFormat* プロパティの値を返します。

```
<$utGetValue("pne_portal","emailFormat")$>
```

関連項目

- utLoad（6-894 ページ）
- utLoadResultSet（6-896 ページ）

UTLOAD

タイプおよび使用方法

❖ グローバル関数（4-21 ページ）

❖ パーソナライズ（5-31 ページ）

説明

utGetValue（6-893 ページ）および utLoadResultSet（6-896 ページ）関数によって使用できるように、パーソナライズ（ユーザー・トピック）ファイルをロードします。

❖ ユーザー・トピック・ファイルは、<*install_dir*>/data/users/profiles/*us*/*username*/ ディレクトリにある HDA ファイルです。

パラメータ

1 つのパラメータを取ります。

❖ パラメータは、ユーザー・トピック HDA ファイルの名前のみです。

出力

なし。

例

現在のユーザーの *wf_in_queue.hda* ファイルをロードし、他のユーザー・トピック関数が使用できるようにします。

```
<$utLoad("wf_in_queue")$>
```

関連項目

– utGetValue（6-893 ページ）

– utLoadResultSet（6-896 ページ）

UTLOADDOCUMENTPROFILES

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- コンテンツ・プロファイル（5-7 ページ）

説明

ユーザーの現在の「My Check In」および「My Search」リンクのコンテンツ・プロファイルの情報を取得するために使用されます。

パラメータ

パラメータを 2 つ取ります。

- ユーザー・トピック名
- パーソナライズ・リンクでユーザーが利用できるコンテンツ・プロファイルを含む結果セットの名前

出力

なし。

例

```
<$utLoadDocumentProfiles("pne_portal", "PneDocumentProfiles")
```

関連項目

– loadDocumentProfile（6-497 ページ）

UTLOADRESULTSET

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- パーソナライズ（5-31 ページ）
- ResultSet（5-32 ページ）

説明

パーソナライズ（ユーザー・トピック）ファイルに指定された ResultSet を DataBinder にロードします。

- ユーザー・トピック・ファイルは、<*install_dir*>/data/users/profiles/*us*/*username*/ ディレクトリにある HDA ファイルです。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、ユーザー・トピック HDA ファイルの名前です。
- 2 番目のパラメータは、ユーザー・トピック・ファイルに定義された ResultSet の名前です。

出力

- ResultSet が DataBinder に正常にロードされた場合は、TRUE を返します。
- 関数が失敗した場合、FALSE を返します。

例

現在のユーザーの *pne_portal.hda* ファイルから、*SavedQueries* ResultSet を DataBinder にロードします。

```
<$utLoadResultSet("pne_portal","SavedQueries")$>
```

関連項目

- utGetValue（6-893 ページ）
- utLoad（6-894 ページ）

VALUESTYLE

タイプおよび使用方法

❖ フィールド情報変数（4-26 ページ）

❖ フィールド表示（5-24 ページ）

説明

メタデータ・フィールドの値のスタイルを指定します。

❖ 値で一般的に使用される SPAN スタイルを次に示します。これらは *<install_dir>*/shared/config/resources/std_page.htm リソース・ファイルの *std_style_declaration* インクルードに定義されています。

- tableEntry
- xxsmall
- strongHighlight

出力

値のスタイル名を返します。

例

値のスタイルを設定します。

```
<$if isFieldInfoOnly$>
    <$if valueStyle$>
        <$fieldValueStyle = valueStyle$>
    <$else$>
        <$fieldValueStyle = "tableEntry"$>
    <$endif$>
<$endif$>
```

Download アプレットを使用する場合、フィールド値に *xxsmall* スタイルを使用します。

```
<$if useDownloadApplet$>
    <$valueStyle="xxsmall", fieldValue =
    strTrimWs(inc("download_file_by_applet_form_content"))$>
<$else$>
    <$fieldValue = strTrimWs(inc("doc_file_get_copy"))$>
<$endif$>
```

関連項目

– fieldValueStyle（6-230 ページ）

VAULTDIR

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ディレクトリおよびパス（5-22 ページ）
- Inbound Refinery（5-11 ページ）

説明

vault ディレクトリのパスを定義します。

- これはコンテンツ・サーバーのネイティブ・ファイル・リポジトリのルート・ディレクトリです。
- この設定は、コンテンツ・サーバーのルート・ディレクトリに ***vault*** ディレクトリがない場合にのみ必要です。
- デフォルト値はありません。

場所

- Content Server: <*install_dir*>/bin/intradoc.cfg
- Inbound Refinery: <*install_dir*>/IdcRefinery/connections/main/intradoc.cfg

例

次は構成エントリの使用例です。

```
VaultDir=f:/vault
```

VDKSUMMARY

タイプおよび使用方法

- 値変数（4-34 ページ）
- 索引付け（5-28 ページ）
- コンテンツ・アイテム（5-6 ページ）

説明

コンテンツ・アイテムの Verity Integration フルテキスト索引のサマリーを取得します。Verity Integration は、Content Server とともに使用できるアドオン製品であり、購入することができます。

出力

文字列を返します。

例

生成されたコンテンツ・アイテムのサマリーを表示します。

```
<$VDKSUMMARY$>
```

VERBOSEMODE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Inbound Refinery（5-11 ページ）
- デバッグ（5-20 ページ）

説明

Inbound Refinery ログのログ・レベルを指定します。

- TRUE に設定すると、すべてのメッセージおよび構成情報が Inbound Refinery ログに記録されます（PDF が最適化されているかどうか、カスタム・コンバータ・パスの場所、カスタム変換のために許容される時間など）。
- FALSE に設定すると、エラー・メッセージのみが Inbound Refinery ログに記録されます。
- デフォルトは、FALSE です。

場所

- 「Local Inbound Refinery Configuration」画面→「General」タブ→「Verbose logging」
- <*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
VerboseMode=true
```

VERITYAPPNAME

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）

説明

Verity ライセンス・キー。Verity Integration は、Content Server とともに使用できるアドオン製品であり、購入することができます。

警告 : サポート・サービスまたはコンサルティング・サービスから特定の Verity ライセンス・キーを指定された場合以外は、この設定を変更しないでください。

場所

<*install_dir*>/search/search.cfg

例

次は構成エントリの使用例です。

```
VerityAppName=INTRANET SOLUTIONS
```

VERITYAPPSIGNATURE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）

説明

Verity ライセンス・キー。Verity Integration は、Content Server とともに使用できるアドオン製品であり、購入することができます。

警告 : サポート・サービスまたはコンサルティング・サービスから特定の Verity ライセンス・キーを指定された場合以外は、この設定を変更しないでください。

場所

<*install_dir*>/search/search.cfg

例

次は構成エントリの使用例です。

```
VerityAppSignature=S62D-4MFB-623K-EB6D-9KUX
```

VERITYENCODING

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）
- ローカライゼーション（5-29 ページ）

説明

Verity コレクションの文字コードを設定します。Verity Integration は、Content Server とともに使用できるアドオン製品であり、購入することができます。

- この値は、SystemLocale（6-789 ページ）で指定された設定を上書きしますが、互換性のない VerityLocale（6-905 ページ）および *VerityEncoding* の選択肢が多数あるため、直接に *SystemLocale* を構成するのが優先されるソリューションです。
- <*install_dir*>/shared/config/resources/std_locale.htm ファイルの *LocaleConfig* 表に、サポートされるシステム・ロケールのコード体系のリストがあります。サポートされるコード体系の詳細は、http://java.sun.com/products/jdk/1.2/docs/guide/internat/encoding.doc.html を参照してください。
- デフォルトは、*SystemLocale* のコードです。

注意：詳細は、『Using Content Server in International Environments』を参照してください。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
VerityEncoding=EUC_KR
```

VERITYINSTALLDIR

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）

説明

Verity の索引付けおよび検索エンジンのインストール・ディレクトリを定義します。Verity Integration は、Content Server とともに使用できるアドオン製品であり、購入することができます。

- デフォルトは、*<install_dir>/shared/vdk/common* または *<install_dir>/shared/vdk4/common* です。

警告：サポート・サービスまたはコンサルティング・サービスから提案された場合以外は、この設定を変更しないでください。

場所

<install_dir>/search/search.cfg

例

次は構成エントリの使用例です。

```
VerityInstallDir=c:/search97/common
```

VERITYLOCALE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 索引付け（5-28 ページ）
- ローカライゼーション（5-29 ページ）

説明

Verity ロケールの言語を定義します。Verity Integration は、Content Server とともに使用できるアドオン製品であり、購入することができます。

- この値は、SystemLocale（6-789 ページ）で指定された設定を上書きしますが、互換性のない *VerityLocale* および VerityEncoding（6-903 ページ）の選択肢が多数あるため、直接に *SystemLocale* を構成するのが優先されるソリューションです。
- <*install_dir*>/shared/config/resources/std_locale.htm ファイルの *VerityLocaleConfig* 表に、システム・ロケールに対応する Verity ロケールのリストがあります。
- デフォルトは、*SystemLocale* のコードです。

注意：ローカライゼーションの詳細は、『Using Content Server in International Environments』を参照してください。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
VerityLocale=germanx
```

WARNABOUTTIMEZONE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 日付と時間（5-19 ページ）
- ローカライゼーション（5-29 ページ）

説明

コンピュータのタイムゾーンを判別できない場合、コンテンツ・サーバーのインストーラが警告を表示するかどうかを指定します。

- TRUE に設定すると、デフォルトのタイムゾーンが使用され、警告が表示されます。
- FALSE に設定すると、デフォルトのタイムゾーンが使用されますが、警告は表示されません。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
WarnAboutTimeZone=true
```

WATCHDOGINTERVAL

タイプおよび使用方法

- Collaboration Manager の構成変数（4-4 ページ）
- Collaboration Manager（5-5 ページ）

説明

ブラウザが Content Server からのレスポンスをチェックするまでの待機時間をミリ秒単位で指定します。指定された時間までにレスポンスを受信しなかった場合は、サーバーが応答していないことを示す警告メッセージが発行されます。

- デフォルトは、5000 です。
- Collaboration Manager のインストール中に追加されます。

場所

<*install_dir*>/<*instancename*>/data/components/CollaborationManager/config.cfg

例

```
watchdogInterval=2000
```

関連項目

- actionItemInterval（6-9 ページ）
- deleteActionItemInterval（6-145 ページ）
- deleteWatchdogInterval（6-146 ページ）

WEBBROWSERPATH

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ディレクトリおよびパス（5-22 ページ）

説明

スタンドアロン・アプリケーションから Content Server のヘルプ・ファイルを表示する Web ブラウザのパスを定義します。

- デフォルト値はありません。

場所

- 「System Properties」→「Paths」タブ→「Browser Executable Path」
- <*install_dir*>/bin/intradoc.cfg

例

次は構成エントリの使用例です。

```
WebBrowserPath=c:/Program Files/Internet Explorer/Iexplore.exe
```

WEBLAYOUTDIR

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ディレクトリおよびパス（5-22 ページ）
- Inbound Refinery（5-11 ページ）

説明

weblayout ディレクトリのパスを定義します。

- これはコンテンツ・サーバー Web サイトのルート・ディレクトリです。
- この設定は、コンテンツ・サーバーのルート・ディレクトリに weblayout ディレクトリがない場合にのみ必要です。
- デフォルト値はありません。

場所

- Content Server: <*install_dir*>/bin/intradoc.cfg
- 管理サーバー : <*install_dir*>/admin/bin/intradoc.cfg
- Inbound Refinery: <*install_dir*>/IdcRefinery/connections/main/intradoc.cfg

例

次は構成エントリの使用例です。

```
WeblayoutDir=f:/weblayout
```

WEBPROXYADMINSERVER

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

管理サーバーがコンテンツ・サーバー・インスタンスを制御するかどうかを定義します。

- TRUE に設定すると、コンテンツ・サーバーは、管理サーバーによって直接制御されます。
- FALSE に設定すると、コンテンツ・サーバーは、管理サーバーによって直接制御されません。（ただし、コンテンツ・サーバーは、別のコンテンツ・サーバー・インスタンスの管理サーバーによって制御される場合があります。）
- マスター・コンテンツ・サーバーでは、デフォルトは TRUE です。
- プロキシ・コンテンツ・サーバーでは、デフォルトは FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
WebProxyAdminServer=false
```

WEBSERVERAUTHONLY

タイプおよび使用方法

- 構成変数（4-3 ページ）
- Web サーバー（5-42 ページ）
- 外部セキュリティ（5-37 ページ）

説明

Web サーバー・フィルタによって、コンテンツ・サーバー・データベースに定義されたユーザーを認証するかどうかを指定します。

- TRUE に設定すると、Web サーバー・フィルタは、コンテンツ・サーバー・データベースからの内部ユーザー名およびパスワードに対するユーザーの検証を行いません。
- IIS の Basic 認証が有効な場合は、この設定を TRUE にする必要があります。これにより、Microsoft Network にログインするとき、ブラウザは Basic 認証を使用できます。ユーザーは、Netscape ブラウザを使用して、Windows ユーザー名とパスワードを使用してログインできます。
- このオプションを有効にすると、コンテンツ・サーバーのユーザー・データベースに対するユーザーの認証が行われないため、Windows ネットワーク・ドメインへの追加のみで新規ユーザーを追加できます。ただし、Windows ネットワーク・ユーザーと同じ名前のユーザーがコンテンツ・サーバー・データベースに存在する場合は、Windows ネットワーク・プロファイルのかわりに、コンテンツ・サーバーのセキュリティ・プロファイルが使用されます。
- デフォルトは、FALSE です。

場所

<*install_dir*>/config/config.cfg

例

次は Web サーバー認証のみを有効にする構成エントリとしての使用例です。

```
WebServerAuthOnly=true
```

WFACTION

タイプおよび使用方法

- ワークフロー変数（4-39 ページ）
- ワークフロー（5-44 ページ）

説明

リビジョンに対して現在実行されているアクションを取得します。

- この変数は、ワークフロー・ステップの Exit イベントの**後で**設定されるため、通常では、次のステップの Entry イベント中に評価されます。
- 次のアクションがあります。
  - APPROVE
  - REJECT
  - CHECKIN
  - CONVERSION
  - META_UPDATE
  - TIMED_UPDATE
  - RESUBMIT

出力

アクションの名前を文字列として返します。

例

リビジョンが却下される過程にある場合、元の作成者に通知します。

```
<$if wfAction like "REJECT"$>
    <$wfSet("wfJumpName", "notifyAuthor")$>
    <$wfSet("wfJumpEntryNotifyOff", "0")$>
    <$wfNotify(dDocAuthor, "user")$>
<$endif$>
```

現在、リビジョンが承認されているが、*xDept* のメタデータ値が "HR" ではない場合、承認を終了してエラー・ページを表示します。

```
<$if wfAction like "APPROVE" and not(xDept like "HR")$>
    <$abortToErrorPage("The revision is not in HR.")$>
<$endif$>
```

WFADDACTIONHISTORYEVENT

タイプおよび使用方法

- ワークフロー関数（4-38 ページ）
- ワークフロー（5-44 ページ）

説明

ワークフローのコンパニオン・データの WorkflowActionHistory 表にイベントを挿入します。この関数の 3 つのパラメータは必須です。その他の行の値は、計算されるか、ローカル・データから継承されます。

パラメータ

パラメータを 3 つ取ります。

- 最初のパラメータは、ワークフロー・アクションです。
- 2 番目のパラメータは、アクションに関連付けられたメッセージです。
- 3 番目のパラメータは、現在のワークフロー・ステップのユーザーをカンマで区切ってリストします。現在のリストを計算する関数の詳細は、`wfComputeStepUserList` を参照してください。

出力

なし。

関連項目

– wfComputeStepUserList（6-916 ページ）

WFADDITIONALEXITCONDITION

タイプおよび使用方法

- ワークフロー変数（4-39 ページ）
- ワークフロー（5-44 ページ）

説明

現在のステップに定義された終了条件を取得します。

出力

終了条件を文字列で返します。

例

<*install_dir*>/shared/config/templates/workflow_info.htm テンプレート・ページにある次のコードを使用して、「Workflow Step Information」ページに終了条件を表示します。

```
<$wfDisplayCondition(dWfName, currentStepName,
"wfAdditionalExitCondition")$>
```

この変数に対する一般的な終了条件の出力は、次のようになります。

```
dSecurityGroup like "Secure"
```

WFADDUSER

タイプおよび使用方法

- ワークフロー関数（4-38 ページ）
- ワークフロー（5-44 ページ）
- ユーザー（5-40 ページ）

説明

ワークフロー・ステップのレビューアのリストにユーザー、別名またはワークフロー・トークンを追加します。この関数は、トークン内でのみ使用できます。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、ユーザー名、別名またはトークン名を示します。
- 2 番目のパラメータは、user または alias のいずれかのタイプを示します。

出力

なし。

例

ユーザー mjones をレビューアとして追加します。

```
<$wfAddUser("mjones", "user")$>
```

元の作成者のトークンおよび hr 別名をレビューアとして追加します。

```
<$wfAddUser(dDocAuthor, "user")$>
<$wfAddUser("hr", "alias")$>
```

WFCOMPUTESTEPUSERLIST

タイプおよび使用方法

❖ ワークフロー関数（4-38 ページ）

❖ ワークフロー（5-44 ページ）

説明

ワークフロー内の現在のステップからユーザーのリストを計算します。

パラメータ

なし。

出力

カンマで区切られたユーザーのリストを返します。

関連項目

– wfAddActionHistoryEvent（6-913 ページ）

WFCURRENTGET

タイプおよび使用方法

❖ ワークフロー関数（4-38 ページ）

❖ ワークフロー（5-44 ページ）

説明

コンパニオン・ファイルからローカル状態値を取得します。

パラメータ

1 つのパラメータを取ります。

❖ パラメータは、キーのみです。

出力

コンパニオン・ファイルからローカル状態値を返します。

例

ローカル・キー <step_name>@<workflow_name>.myKey の値を返します。

```
<$wfCurrentGet("myKey")$>
```

関連項目

– wfGet（6-922 ページ）

WFCURRENTSET

タイプおよび使用方法

- ワークフロー関数（4-38 ページ）
- ワークフロー（5-44 ページ）

説明

コンパニオン・ファイル内のキーのローカル状態値を設定します。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、キーです。
- 2 番目のパラメータは、値です。

出力

なし。

例

キー <step_name>@<workflow_name>.myKey を myValue に設定します。

```
<$wfCurrentSet("myKey", "myValue")$>
```

関連項目

– wfSet（6-939 ページ）

WFCURRENTSTEP

タイプおよび使用方法

- ワークフロー関数（4-38 ページ）
- ワークフロー（5-44 ページ）

説明

現在のステップに関連するステップの名前を取得します。

パラメータ

1 つのパラメータを取ります。

- パラメータは、現在のステップに関連するステップ数を表す整数のみです。

出力

- ステップ名を返します。
- ワークフローの始まりまたは終わりに到達すると、最初のステップまたは最後のステップの名前を返します。

例

現在のステップ名を返します。

```
<$wfCurrentStep(0)$>
```

前のステップ名を返します。

```
<$wfCurrentStep(-1)$>
```

次のステップ名を返します。

```
<$wfCurrentStep(1)$>
```

実際に次のステップに移動するには、ジャンプを設定する必要があります。次に例を示します。

```
<$wfSet("wfJumpName", "move to next step")$>
<$wfSet("wfJumpTargetStep", wfCurrentStep(1))$>
<$wfSet("wfJumpEntryNotifyOff", "0")$>
```

WFDISPLAYCONDITION

タイプおよび使用方法

- ワークフロー関数（4-38 ページ）
- ワークフロー（5-44 ページ）

説明

ワークフロー・ステップの終了条件を取得します。

パラメータ

パラメータを 3 つ取ります。

- 最初のパラメータは、ワークフロー名です。
- 2 番目のパラメータは、ステップ名です。
- 3 番目のパラメータは、表示される終了条件です。

出力

終了条件の式を返します。

例

<install_dir>/shared/config/templates/workflow_info.htm テンプレート・ページにある次のコードを使用して、「Workflow Step Information」ページに終了条件を表示します。

```
<$wfDisplayCondition(dWfName, currentStepName, "wfAdditionalExitCondition")$>
```

各パラメータの値は次のとおりです。

- dWfName（6-188 ページ）は、ワークフロー名の内部変数です。
- *currentStepName* は、ステップ名の内部変数である dWfStepName（6-189 ページ）に等しくなるように、テンプレート・ページに先に設定されます。
- wfAdditionalExitCondition（6-914 ページ）は、終了条件式の内部変数です。一般的な終了条件の出力は、次のようになります。

  ```
  dSecurityGroup like "Secure"
  ```

WFEXIT

タイプおよび使用方法

- ワークフロー関数（4-38 ページ）
- ワークフロー（5-44 ページ）

説明

ワークフロー・ステップを終了します。この関数は、関数のパラメータに従ってリビジョンを特定のステップに移動し、親リストの情報をリセットします。

技術ヒント：ワークフローを完全に終了するには、`wfExit(100,100)` を使用します。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、巻き戻すジャンプ数を示します。
- 2 番目のパラメータは、最初のパラメータによって決定されたステップに関連するターゲット・ステップを示します。

技術ヒント：ワークフローを完全に終了するには、`wfExit(100,100)` を使用するか、任意のパラメータを使用して、リビジョンが親ワークフローに戻り、そのワークフローの最後のステップを確実に通過するようにします。

出力

なし。

例

終了してワークフローの親ステップに戻ります。

```
<$wfExit(0,0)$>
```

前のジャンプ・ステップに戻ります。

```
<$wfExit(1,0)$>
```

前のジャンプ・ステップに戻り、そのワークフローの次のステップに移動します。

```
<$wfExit(1,1)$>
```

WFGET

タイプおよび使用方法

❖ ワークフロー関数（4-38 ページ）

❖ ワークフロー（5-44 ページ）

説明

コンパニオン・ファイルから状態値を取得します。

パラメータ

1 つのパラメータを取ります。

❖ パラメータは、状態キーのみです。

出力

コンパニオン・ファイルから状態値を返します。

例

現在のジャンプ名を返します。

```
<$wfGet("wfJumpName")$>
```

関連項目

– wfCurrentGet（6-917 ページ）

WFGETSTEPTYPELABEL

タイプおよび使用方法

- ワークフロー変数（4-39 ページ）
- ワークフロー（5-44 ページ）

説明

この関数は、内部ワークフロー・ステップ・タイプの値を取得し、判読可能なラベルに変換します。たとえば、:C:CA:CE: は、:C:（コントリビューション）、:CA:（自動コントリビューションまたは元の作成者ステップ）および :CE:（リビジョンの編集）の 3 つの状態です。Idoc スクリプト関数を適用すると、「Auto Contributor, Edit Revision」のようなラベルになります。

現在の状態には、次の値があります。

状態	説明
:CA:	自動投稿ステップまたは管理者が定義したすべてのワークフロー・ステップの前に起こるステップ（「元の作成者がドキュメントの却下を処理」）。
:CE:	現在のリビジョンの編集。すべての編集が現在のリビジョンを置換。
:CN:	新規リビジョンの作成。すべての編集が新規リビジョンを作成。
:R:	レビュー。ドキュメントを承認または却下できる。
:C:	投稿。ドキュメントを編集できる。

ワークフロー・ドキュメント・ステップの状態は、そのステップに該当するすべての状態が単一のフィールドとして結合され、複数の値がコロン（:）で区切られています（余分なコロンは削除されます）。したがって、新規リビジョンを作成するコントリビュータ / レビューア・ステップの状態は、:R:C:CE: になります。順序は問題ではありません。ただし、:C: は、:CE: の存在によって推測可能な場合でも必ず存在する必要があります（:CE: は、ステップがコントリビュータ・ステップであることを意味します）。値 :C: は、現在のワークフロー・ステップに関する 1 つの機能を正確に示し、コンテンツ・サーバーは含意的なルール（:CE: -> :C:）を追跡しません。したがって、コンテンツ・サーバーでは、:C: がない場合（:CE: が存在する場合でも）、そのワークフロー・ステップでは主要なコントリビュータ・タイプのアクティビティ（チェックアウトやチェックアウトの取消しなど）が許可されないものと解釈します。

パラメータ

1 つのパラメータを取ります。

❖ パラメータは、内部ワークフロー・ステップ・タイプの値のみです。

例

```
<$wfGetStepTypeLabel(stepType)$>
```

WFISFINISHEDDOCCONVERSION

タイプおよび使用方法

- ワークフロー関数（4-38 ページ）
- ワークフロー（5-44 ページ）

説明

現在のアクションの終了後に、ドキュメントが GENWWW にならないことを示します。

この Idoc スクリプト関数が true を返した場合、変換処理は終了しており、コンテンツ・アイテムは GENWWW ではありません。

パラメータ

なし。

出力

- ドキュメントの変換が終了している場合は、TRUE を返します。
- コンテンツ・アイテムが GENWWW 状態にあり、変換が終了していない場合は、FALSE を返します。

例

```
<$wfIsFinishedDocConversion()$>
```

WFISNOTIFYINGUSERS

タイプおよび使用方法

- ワークフロー関数（4-38 ページ）
- ワークフロー（5-44 ページ）

説明

このワークフローで、現在、この特定のワークフロー・イベントに関するすべてのワークフロー通知が抑止されている場合は TRUE を返します。抑止は、現在のアクションが終了するか、`wfSetIsNotifyingUsers` 関数によって通知が再度有効化されるまで、キュー内のワークフローおよび電子メールの通知に適用されます。

通知の抑止は一時的です。通知が有効であるか、後のワークフロー・アクションまで維持できる場合は、現在のステップで送信されなかったすべての通知が送信され、キュー内のワークフローは適切に更新されます。

パラメータ

なし。

出力

- 通知が抑止されている場合は、TRUE を返します。
- 通知が抑止されていない場合は、FALSE を返します。

例

```
<$wfIsNofifyingUsers()$>
```

関連項目

– wfSetIsNotifyingUsers（6-940 ページ）

WFISRELEASABLE

タイプおよび使用方法

- ワークフロー関数（4-38 ページ）
- ワークフロー（5-44 ページ）

説明

ワークフローに関するかぎり、ドキュメントがリリースされているかどうかを示します。ドキュメントが GENWW または DONE の状態であっても、この Idoc スクリプト関数から true が返された場合、ワークフローはドキュメントがリリースされるのを阻止しません。false が返された場合は、ワークフローでリリースが許可されるまで、ドキュメントはリリースされません。この Idoc スクリプト関数は、パラメータを取りません。この関数は、存在する任意のドキュメント情報のアクティブなリリース状態値を評価します（Idoc スクリプトの観点から言うと、この関数は、#active.dReleaseState が値 E でないかどうかをチェックします）。

パラメータ

なし。

出力

- ドキュメントがリリース可能な場合は、TRUE を返します。
- コンテンツ・アイテムがリリース可能でない場合は、FALSE を返します。たとえば、コンテンツ・アイテムのチェックインや変換処理が完了していない場合などです。

例

```
<$wfIsReleasable()$>
```

WFJUMPENTRYNOTIFYOFF

タイプおよび使用方法

- ワークフロー変数（4-39 ページ）
- ワークフロー（5-44 ページ）

説明

ジャンプ・エントリ通知のオンとオフを切り替えます。

- この変数が TRUE の場合、レビューアはジャンプ・エントリについて通知されません。
- この変数が FALSE の場合、レビューアはジャンプ・エントリについて通知されます。

出力

TRUE または FALSE を返します。

例

次のコードは、ステップ・エントリ通知を 0 に設定します。これにより、ジャンプ・エントリ時にレビューアは通知を受けます。

```
<$wfSet("wfJumpEntryNotifyOff","0")$>
```

WFJUMPMESSAGE

タイプおよび使用方法

- ワークフロー変数（4-39 ページ）
- ワークフロー（5-44 ページ）

説明

ジャンプ・エントリ時にユーザーに送信される電子メール通知に含めるメッセージを定義します。

- メッセージを指定しない場合、電子メール・メッセージには、電子メール・テンプレートの情報のみが含まれます。
- ジャンプ・メッセージに、Idoc スクリプトを含めることができます。これは、eval（6-205 ページ）関数を使用して実行する必要があります。次に例を示します。

```
<$eval(dDocName)$> is ready for your review.
```

出力

メッセージを文字列で返します。

例

次のコードでは、ジャンプ・エントリ時にユーザーに通知する電子メール・メッセージに、「This is the message text」というメッセージを含めます。

```
<$wfSet("wfJumpMessage","This is the message text")$>
```

関連項目

– eval（6-205 ページ）

WFJUMPNAME

タイプおよび使用方法

- ワークフロー変数（4-39 ページ）
- ワークフロー（5-44 ページ）

説明

現在のジャンプ名を取得します。

出力

ジャンプ名を文字列で返します。

例

次のコードは、現在のジャンプの名前を MyJump に設定します。

```
<$wfSet("wfJumpName","MyJump")$>
```

WFJUMPRETURNSTEP

タイプおよび使用方法

- ワークフロー変数（4-39 ページ）
- ワークフロー（5-44 ページ）

説明

現在のジャンプの後、ワークフローの終了時にリビジョンの戻り先となる親ワークフロー内のステップの名前を取得します。

- 戻りステップは、リビジョンがジャンプ先のワークフロー内の最後のステップを正しく完了し、正常にワークフローを終了した場合にのみ適用されます。したがって、リビジョンが別のワークフローにジャンプした場合、戻りステップは適用されません。
- コンパニオン・ファイルで、戻りステップはローカル・キーに保存されます。
  `<step_name>@<workflow_name>.returnStep`

出力

ステップ名を文字列で返します。

例

次のコードは、戻りステップが、親ワークフロー内の次のステップとなるように設定します。

```
<$wfSet("wfJumpReturnStep",wfCurrentStep(1))$>
```

WFJUMPTARGETSTEP

タイプおよび使用方法

- ワークフロー変数（4-39 ページ）
- ワークフロー（5-44 ページ）

説明

条件が満たされた場合、リビジョンのジャンプ先となるステップの名前を取得します。

- ターゲット・ステップは、シンボリック（*wfCurrentStep(1)* など）または明示的に（*MyStep@MyWorkflow* など）参照できます。

注意 : スクリプトの変更と再利用を容易にするため、ステップ・イベントのスクリプトでは、シンボリック参照を使用することを強くお薦めします。

出力

ステップ名を文字列で返します。

例

次のコードは、ターゲット・ステップを明示的に設定します。ジャンプ基準が満たされると、リビジョンは、Marketing ワークフローの step_1 というステップにジャンプします。

```
<$wfSet("wfJumpTargetStep","step_1@Marketing")$>
```

次のコードは、ターゲット・ステップをシンボリックに設定します。ジャンプ基準が満たされると、リビジョンは、現在のワークフローの最初のステップにジャンプします。

```
<$wfSet("wfJumpTargetStep",WfStart)$>
```

WFLOADDESIGN

タイプおよび使用方法

- ワークフロー関数（4-38 ページ）
- ワークフロー（5-44 ページ）

説明

この関数は、ワークフローの既存ステップまたはワークフローでの終了条件に関する情報を取得するために使用します。

この関数は、<*install_dir*>/data/workflow/design/*workflowname*.hda ファイルをロードし、ワークフローの設計情報を含む結果セットを返します。*workflowname* は、通常、ワークフロー・ページ、電子メール・テンプレートおよびジャンプ・スクリプトで利用可能な `dWfName` 変数の値に対応しています。

- 1 つのパラメータ（ワークフロー名）を取ります。

出力

情報を結果セットとして返します。

例

```
<$wfLoadDesign(WorkflowName)$>
```

WFMAILSUBJECT

タイプおよび使用方法

- ワークフロー変数（4-39 ページ）
- ワークフロー（5-44 ページ）

説明

ワークフローの電子メール通知の件名行を定義します。

- 件名行を指定しない場合、通知のタイプ（レビュー、却下またはワークフローの開始）に対するデフォルトの件名が使用されます。
- 件名文字列に、Idoc スクリプトを含めることができます。

出力

件名行を文字列として返します。

例

「*Content ID* has been released」という件名行を使用して、元の作成者に通知します。

```
<$wfMailSubject="File <$dDocName$> has been released"$>
<$wfNotify(dDocAuthor, "user")$>
```

関連項目

- wfNotify（6-936 ページ）
- wfMessage（6-935 ページ）

WFMESSAGE

タイプおよび使用方法

- ワークフロー変数（4-39 ページ）
- ワークフロー（5-44 ページ）

説明

ワークフローの電子メール通知に含めるメッセージを定義します。

- メッセージを指定しない場合、通知のタイプ（レビュー、却下またはワークフローの開始）に対するデフォルトのメッセージが使用されます。
- メッセージ文字列に、Idoc スクリプトを含めることができます。

出力

メッセージを文字列で返します。

例

「*Content ID* has been released」というメッセージを使用して、元の作成者に通知します。

```
<$wfMessage="File <$dDocName$> has been released"$>
<$wfNotify(dDocAuthor, "user")$>
```

関連項目

- wfNotify（6-936 ページ）
- wfMailSubject（6-934 ページ）

WFNOTIFY

タイプおよび使用方法

- ワークフロー関数（4-38 ページ）
- ワークフロー（5-44 ページ）

説明

指定されたユーザー、別名またはワークフロー・トークンに電子メール・メッセージを送信します。

- wfMailSubject（6-934 ページ）および wfMessage（6-935 ページ）変数を設定して、通知メッセージをカスタマイズできます。

パラメータ

2 つのパラメータと、3 番目のオプションのパラメータを取ります。

- 最初のパラメータは、通知先のユーザー名、別名またはトークンを指定します。
- 2 番目のパラメータは、user、alias または token のいずれかのタイプを示します。
- 3 番目のオプションのパラメータは、メッセージの作成に使用する電子メール・テンプレート名を指定します。（テンプレートの定義は、<*install_dir*>/shared/config/templates/templates.hda ファイルを参照してください。）

出力

なし。

例

元の作成者に通知します。

```
<$wfNotify(dDocAuthor, "user")$>
```

<*install_dir*>/shared/config/templates/reject_mail.htm ファイルをテンプレートとして使用して、myAlias 別名のすべてのユーザーに通知します。

```
<$wfNotify("myAlias", "alias", "WF_REJECT_MAIL")$>
```

関連項目

- wfMailSubject（6-934 ページ）
- wfMessage（6-935 ページ）

WFPARENTLIST

タイプおよび使用方法

- ワークフロー変数（4-39 ページ）
- ワークフロー（5-44 ページ）

説明

リビジョンがアクセスしたワークフロー・ステップのリスト（親リスト）を取得します。このリストは、ジャンプが発生した場所、およびエラー、却下または終了の際にステップを巻き戻す方法をシステムに対して示します。

- この変数は、条件文の作成に使用できますが、ハードコードまたは変更することはできません。
- エラー、却下または終了によって親リストが巻き戻された場合、リストからステップが削除されるため、親リストに完全なステップ履歴が反映されない場合があります。
- 親リストはグローバルであり、ステップ名によってローカライズされません。
- 親リスト内のステップは、最新のステップが最初にリストされます。各ステップは、番号記号（#）によって区切られています。ステップ名の前のアスタリスクは、ジャンプ・ステップであることを示します。次に例を示します。

```
Step_B@Workflow_2#*Step_A@Workflow_2#Step_1@Workflow_1
```

出力

親リストを文字列で返します。

例

親リストの使用方法の 1 つは、ワークフロー・スクリプトをテストするときに、コンテンツ・アイテムがアクセスしたジャンプをシミュレートすることです。たとえば、「Test」タブまたは「Edit Script」画面で、コンテンツ・アイテムを選択して「Load Item's Workflow State」をクリックすると、次のような行が表示されます。

```
wfParentList=step_1@Marketing#contribution@Marketing
```

WFRELEASEDOCUMENT

タイプおよび使用方法

- ワークフロー変数（4-39 ページ）
- ワークフロー（5-44 ページ）

説明

現在ワークフローによってロックされているドキュメントについて、すべての未処理のドキュメント・リビジョンがリリースされます。このようにしてリリースされたリビジョンを、索引付けエンジンによって評価することができます。その結果、リリースされたリビジョンの 1 つがフルテキスト索引付けされる場合もあります。

この関数は、ワークフロー・イベントで処理中のワークフロー・ステップの Idoc スクリプトの評価というコンテキストでのみ使用できます。ドキュメントがすでにワークフローからリリースされている場合、この関数は効果がありません。この関数によって、ドキュメントのワークフローが終了することはありません。

ドキュメント用に作成された新規リビジョンは、ロックされた（リリース不可）状態で戻ります。つまり、この関数は、現在のリビジョンをリリースさせますが、作成された新規リビジョンには影響を与えません。この関数にはパラメータがなく、結果を返しません。この関数は、現在のアクティブなワークフロー・ドキュメントに作用します。

パラメータ

なし。

出力

なし。

例

```
<$wfReleaseDocument()$>
```

WFSET

タイプおよび使用方法

❖ ワークフロー関数（4-38 ページ）

❖ ワークフロー（5-44 ページ）

説明

コンパニオン・ファイル内の特定の値を使用してキーを設定します。

パラメータ

パラメータを 2 つ取ります。

❖ 最初のパラメータは、キーです。

❖ 2 番目のパラメータは、値です。

出力

なし。

例

キー wfJumpName を MyJump に設定します。

```
<$wfSet("wfJumpName", "MyJump")$>
```

関連項目

– wfCurrentSet（6-918 ページ）

WFSETISNOTIFYINGUSERS

タイプおよび使用方法

- ワークフロー関数（4-38 ページ）
- ワークフロー（5-44 ページ）

説明

現在のアクション（チェックイン、更新、再発行など）中に、ワークフロー通知を送信するかどうかを示す内部フラグを設定します。

抑止は、電子メールおよびキュー内のワークフローに対する更新に適用されます。この関数のもう 1 つの使用方法は、ドキュメントの変換後まで、ワークフロー通知を抑止することです。これにより、ドキュメントの変換が終了したとき、自動コントリビュータ・ワークフロー・ステップの外へドキュメントが進むことを防止できます。

パラメータ

1 つのパラメータ（内部フラグ）を取ります。1（TRUE）に設定すると、通知は抑止されません。0（FALSE）に設定すると、通知が抑止されます。

出力

なし。

例

```
<$wfSetIsNotifyingUsers(0)$>
```

関連項目

– wfIsNotifyingUsers（6-926 ページ）

WFSTART

タイプおよび使用方法

- ワークフロー変数（4-39 ページ）
- ワークフロー（5-44 ページ）

説明

現在のワークフローの最初のステップにリビジョンを送ります。

- この変数は、大文字の W で始まることに注意してください。

出力

なし。

例

ジャンプのターゲット・ステップを設定してワークフローを再開します。

```
<$wfSet("wfJumpTargetStep",WfStart)$>
```

wfUpdateMetaData

タイプおよび使用方法

- ワークフロー関数（4-38 ページ）
- ワークフロー（5-44 ページ）

説明

ワークフロー内の現在のコンテンツ・アイテムのリビジョンのメタデータ値を定義します。

注意 : wfUpdateMetaData 関数は、カスタム・メタデータ・フィールドでのみ使用できます。この関数は、標準の事前定義済フィールドの更新には使用できません。

パラメータ

パラメータを 2 つ取ります。

- 最初のパラメータは、メタデータ・フィールドの名前です。
- 2 番目のパラメータは、メタデータ・フィールドの値です。

出力

なし。

例

「This is my comment.」を「Comments」フィールドの値として定義します。

```
<$wfUpdateMetaData("xComments", "This is my comment.")$>
```

WORKFLOWDIR

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ディレクトリおよびパス（5-22 ページ）
- ワークフロー（5-44 ページ）

説明

ワークフロー・データが含まれる workflow ディレクトリのパスを定義します。

- デフォルトは、<*install_dir*>/data/workflow/ です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
WorkflowDir=c:/workflow_info/
```

WORKFLOWINTERVALHOURS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- ワークフロー（5-44 ページ）

説明

各ワークフロー更新サイクル間の時間を定義します（時間）。

- ワークフロー内のリビジョンの更新イベント・スクリプトは、この間隔で実行されます。
- デフォルトは、1 時間です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
WorkflowIntervalHours=2
```

XML

タイプおよび使用方法

- グローバル関数（4-21 ページ）
- 文字列（5-39 ページ）

説明

XML で使用するための文字列を書式設定します。

- この関数は、英数字以外の文字を、その文字を表す適切な XML シーケンスに置換します。たとえば、アンパサンド（&）文字は、& シーケンスに置換されます。
- 文字列中に二重引用符文字を使用する場合、二重引用符を文字として表示するには、二重引用符の前にバックスラッシュ（\）が必要です。バックスラッシュをエスケープ・フラグとして使用しない場合、二重引用符は、文字列の終わりとして解釈されます。

パラメータ

1 つのパラメータを取ります。

- パラメータは、文字列のみです。

出力

URL で使用する書式に設定された文字列を返します。

例

アンパサンドをエスケープし、XML 書式の文字列「Me & you.」を返します。

```
<$xml("Me & you.")$>
```

英数字以外の文字をエスケープし、XML 書式の文字列「Test the ", >, and < characters.」を返します。

```
<$xml("Test the \", >, and < characters.")$>
```

XMLENCODINGMODE

タイプおよび使用方法

- 構成変数（4-3 ページ）
- コンテンツ・サーバー（5-8 ページ）

説明

コンテンツ・サーバーで使用するブラウザの URL の国際文字を指定するために使用されます。

この変数を FULL に設定すると、次のような場合に役立ちます。

- UTF-8 を使用しているブラウザでコンテンツ・サーバーを使用する場合（「Always send URLs as UTF-8」が有効な場合）。これは通常、Microsoft ブラウザの旧バージョンでのみ発生します。
- コンテンツ・サーバーの URL を Javascript 変数に設定するカスタム・コンポーネントを作成する場合。この場合、URL の完全なエンコードが必要です。
- PDA タイプのデバイスを使用してコンテンツ・サーバーにアクセスする場合（PALM、ハンドヘルド・コンピュータなど）。これらのデバイスによるアクセスを提供するコンポーネントを使用する場合は、URL の完全なエンコードが必要です。これらのデバイスの URL では、非 7 ビット文字がサポートされない場合があります。
- デフォルトは、PARTIAL です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
XMLEncodingMode=full
```

XMLTIMEOUT

タイプおよび使用方法

- 構成変数（4-3 ページ）
- XML Converter（5-16 ページ）

説明

XML 変換のタイムアウトを秒数で定義します。

- XML のタイムアウト値を返します。
- デフォルトは、60（1 分）です。

場所

<*install_dir*>/IdcRefinery/connections/*main*/intradoc.cfg

例

次は構成エントリの使用例です。

```
XMLTimeout=30
```

ZONEDSECURITYFIELDS

タイプおよび使用方法

- 構成変数（4-3 ページ）
- 検索（5-34 ページ）
- 索引付け（5-28 ページ）

説明

ゾーン検索可能（フルテキスト検索可能）なメタデータ・フィールドを指定します。

- フルテキスト検索では、指定したメタデータ・フィールドが検索に含められ、「Search」ページの指定されたフィールドに入力した検索条件はフルテキスト検索条件とみなされます。
- この構成変数を設定し、アカウントをフルテキスト検索可能なフィールドとして維持する場合は、*dDocAccount* をリストに含める必要があります。
- デフォルトは、*dDocAccount* です。

場所

<*install_dir*>/config/config.cfg

例

次は構成エントリの使用例です。

```
ZonedSecurityFields=dDocTitle,xComments
```

A

サード・パーティ・ライセンス

概要

この付録では、この製品に含まれるすべてのサード・パーティ製品のサード・パーティ・ライセンスを示します。

- Apache Software License（A-2 ページ）
- W3C® Software Notice and License（A-2 ページ）
- Zlib License（A-4 ページ）
- General BSD License（A-5 ページ）
- General MIT License（A-5 ページ）
- Unicode License（A-6 ページ）
- その他の帰属（A-7 ページ）

APACHE SOFTWARE LICENSE

```
* Copyright 1999-2004 The Apache Software Foundation.
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
```

W3C® SOFTWARE NOTICE AND LICENSE

```
* Copyright © 1994-2000 World Wide Web Consortium,
* (Massachusetts Institute of Technology, Institut National de
* Recherche en Informatique et en Automatique, Keio University).
* All Rights Reserved.  http://www.w3.org/Consortium/Legal/
*
* This W3C work (including software, documents, or other related items) is
* being provided by the copyright holders under the following license.By
* obtaining, using and/or copying this work, you (the licensee) agree that
* you have read, understood, and will comply with the following terms and
* conditions:
*
* Permission to use, copy, modify, and distribute this software and its
* documentation, with or without modification, for any purpose and without
* fee or royalty is hereby granted, provided that you include the following
* on ALL copies of the software and documentation or portions thereof,
* including modifications, that you make:
*
*   1. The full text of this NOTICE in a location viewable to users of the
*      redistributed or derivative work.
*
```

*   2. Any pre-existing intellectual property disclaimers, notices, or terms
*   and conditions.If none exist, a short notice of the following form
*   (hypertext is preferred, text is permitted) should be used within the
*   body of any redistributed or derivative code: "Copyright ©
*   [$date-of-software] World Wide Web Consortium, (Massachusetts
*   Institute of Technology, Institut National de Recherche en
*   Informatique et en Automatique, Keio University).All Rights
*   Reserved. http://www.w3.org/Consortium/Legal/"
*
*   3. Notice of any changes or modifications to the W3C files, including the
*   date changes were made.(We recommend you provide URIs to the location
*   from which the code is derived.)
*
* THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS
* MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT
* NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR
* PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE
* ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.
*
* COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR
* CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR
* DOCUMENTATION.
*
* The name and trademarks of copyright holders may NOT be used in advertising
* or publicity pertaining to the software without specific, written prior
* permission.Title to copyright in this software and any associated
* documentation will at all times remain with copyright holders.
*

ZLIB LICENSE

* zlib.h -- interface of the 'zlib' general purpose compression library
version 1.2.3, July 18th, 2005

Copyright (C) 1995-2005 Jean-loup Gailly and Mark Adler
This software is provided 'as-is', without any express or implied
warranty.In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
claim that you wrote the original software.If you use this software
in a product, an acknowledgment in the product documentation would be
appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly jloup@gzip.org
Mark Adler madler@alumni.caltech.edu

GENERAL BSD LICENSE

Copyright (c) 1998, Regents of the University of California

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

"Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

"Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

"Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GENERAL MIT LICENSE

Copyright (c) 1998, Regents of the Massachusetts Institute of Technology

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

UNICODE LICENSE

UNICODE, INC. LICENSE AGREEMENT - DATA FILES AND SOFTWARE

Unicode Data Files include all data files under the directories http://www.unicode.org/Public/, http://www.unicode.org/reports/, and http://www.unicode.org/cldr/data/ .Unicode Software includes any source code published in the Unicode Standard or under the directories http://www.unicode.org/Public/, http://www.unicode.org/reports/, and http://www.unicode.org/cldr/data/.

NOTICE TO USER: Carefully read the following legal agreement.BY DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING UNICODE INC.'S DATA FILES ("DATA FILES"), AND/OR SOFTWARE ("SOFTWARE"), YOU UNEQUIVOCALLY ACCEPT, AND AGREE TO BE BOUND BY, ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT.IF YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE THE DATA FILES OR SOFTWARE.

COPYRIGHT AND PERMISSION NOTICE

Copyright © 1991-2006 Unicode, Inc. All rights reserved.Distributed under the Terms of Use in http://www.unicode.org/copyright.html.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that (a) the above copyright notice(s) and this permission notice appear with all copies of the Data Files or Software, (b) both the above copyright notice(s) and this permission notice appear in associated documentation, and (c) there is clear notice in each modified Data File or in the Software as well as in the documentation associated with the Data File(s) or Software that the data or software has been modified.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

_____Unicode and the Unicode logo are trademarks of Unicode, Inc., and may be registered in some jurisdictions.All other trademarks and registered trademarks mentioned herein are the property of their respective owners

その他の帰属

Adobe, Acrobat, and the Acrobat Logo are registered trademarks of Adobe Systems Incorporated.
FAST Instream is a trademark of Fast Search and Transfer ASA.
HP-UX is a registered trademark of Hewlett-Packard Company.
IBM, Informix, and DB2 are registered trademarks of IBM Corporation.
Jaws PDF Library is a registered trademark of Global Graphics Software Ltd.
Kofax is a registered trademark, and Ascent and Ascent Capture are trademarks of Kofax Image Products.
Linux is a registered trademark of Linus Torvalds.
Mac is a registered trademark, and Safari is a trademark of Apple Computer, Inc.
Microsoft, Windows, and Internet Explorer are registered trademarks of Microsoft Corporation.
MrSID is property of LizardTech, Inc. It is protected by U.S. Patent No. 5,710,835. Foreign Patents Pending.
Oracle is a registered trademark of Oracle Corporation.
Portions Copyright © 1994-1997 LEAD Technologies, Inc. All rights reserved.
Portions Copyright © 1990-1998 Handmade Software, Inc. All rights reserved.
Portions Copyright © 1988, 1997 Aladdin Enterprises.All rights reserved.
Portions Copyright © 1997 Soft Horizons.All rights reserved.
Portions Copyright © 1995-1999 LizardTech, Inc. All rights reserved.
Red Hat is a registered trademark of Red Hat, Inc.
Sun is a registered trademark, and Sun ONE, Solaris, iPlanet and Java are trademarks of Sun Microsystems, Inc.
Sybase is a registered trademark of Sybase, Inc.
UNIX is a registered trademark of The Open Group.
Verity is a registered trademark of Autonomy Corporation plc

索引

記号

#active, 3-18
#env, 3-18
#local, 3-18
% 演算子 , 3-24
& 演算子 , 3-23
* 演算子 , 3-24
* ワイルドカード , 3-24
+ 演算子 , 3-24
- 演算子 , 3-24
/ 演算子 , 3-24
? ワイルドカード , 3-24
| 演算子 , 3-23

A

abortToErrorPage, 6-2
AcadUseLISPInterface, 6-3
Account, 3-27
AccountMapPrefix, 6-4
AcctPermDelim, 6-5
AcctPrefix, 6-7
actionItemInterval, 6-9
addEmptyOption, 6-10
AdditionalIndexBuildParams, 6-11
AdditionalSubscriptionCollatedFieldList, 6-14
AdjustPrinterMargins, 6-15
AdminAtLeastOneGroup, 6-16
AdminDataDir, 6-17
AdsSimpleAuth, 6-18
AdsUserName, 6-19
AdsUserPassword, 6-20
AfterLogin, 6-21
AllowAllRedirectUrl, 6-22
AllowAlternateMetaFile, 6-23
AllowCheckin, 6-24
AllowCheckout, 6-25
AllowConcurrentUpdate, 6-26
AllowForceDelete, 6-27
AllowIntranetUsers, 6-28
AllowMatchesInDateCheck, 6-29
AllowMixedCaseACL, 6-30
AllowMrSidRetryCompressionLevel, 6-31
AllowPassthru, 6-32
AllowPrimaryMetaFile, 6-33
AllowRetentionPeriodWithoutCutoff, 6-34
AllowReview, 6-35
AllowSamePrimaryAlternateExtensions, 6-36
AllowSecondaryConversion, 6-37
AllowSkippedHyperlinktoCauseIncomplete, 6-38
and 演算子 , 3-25
AppletChunkSize, 6-39
AppletChunkThreshold, 6-40
ARCHIVE_COLLECTION, 6-41
ArchiverDoLocks, 6-42
ArchiverTouchMonitorInterval, 6-43
ArchiverTraceLockMaintenance, 6-44
ArchiverTraceLocks, 6-45
ASCII 文字コード 39 番 , 3-37
ASCII 文字コード 96 番 , 3-37
AttributeMap, 6-46
Author, 3-27
AuthorAddress, 6-47
AuthorDelete, 6-48
AutoCAD, 5-14
AutoCad2000PlotterFilePath, 6-49
AutoMaxMergeTrace, 6-50
AutoNumberPrefix, 6-51
AutoNumberWidth, 6-52

B

BatchLoaderPath, 6-53
BatchLoaderUserName, 6-54
BinDir, 6-55

break, 3-15, 6-56
BrowserVersionNumber, 6-57

C

c, 6-58
CachedResultRowCount, 6-59
cacheInclude, 6-60
captionEntryWidth, 6-62
captionFieldWidth, 6-63
CGI_DEBUG, 6-64
CGI_RECEIVE_DUMP, 6-65
CGI_SEND_DUMP, 6-66
CgiFileName, 6-67
CharMap, 6-68
CheckMkvdkDocCount, 6-69
ChunkedRequestTrace, 6-70
CLASSPATH, 6-71
CleanUp, 6-72
clearSchemaData, 6-73
ClientControlled, 6-74
ClusterBinDirRule, 6-75
ClusterNodeAddress, 6-76
ClusterNodeName, 6-77
cmDiscussionDocType, 6-78
cmMemoFieldSize, 6-79
cmZonedSecurityFields, 6-80
Collaboration Manager, 5-5
CollectionDeleteEnabled, 6-81
CollectionHiddenMeta, 6-82
CollectionInhibitUpdateMeta, 6-83
CollectionLocation, 6-84
CollectionMeta, 6-85
CollectionMoveEnabled, 6-86
CollectionReadOnlyMarkedFolders, 6-87
CollectionReadOnlyMeta, 6-88
CollectionReleasedOnly, 6-89
CollectionSecurityReadOnly, 6-90
CollectionTrashDeleteDate, 6-91
CollectionTrashDeleteLocation, 6-92
CollectionTrashDeleteOldName, 6-93
CollectionTrashDeleter, 6-94
CollectionWebDAVServer, 6-95
ColumnMapFile, 6-96
Comments, 3-27
Compression, 5-18
CompressionMinHeight, 6-97
CompressionMinWidth, 6-98
computeDocUrl, 6-99
computeRenditionUrl, 6-100
ComputerName, 6-101
ConfigDir, 6-102
ConnectionName, 6-103
Content ID, 3-27
CONTENT_LENGTH, 6-104
ConversionCacheExpirationPeriodInDays, 6-105
ConvertXmlAsWebviewable, 6-106
CookieAuthTimeoutInDays, 6-107
CookieProxyPassword, 6-108
coreContentOnly, 6-109
createAlternateMetaFile, 6-110
CreatePDFThumbnails, 6-111
createPrimaryMetaFile, 6-112
CreateXMLRendition, 6-113
CURRENT_DATE, 6-114
CURRENT_ROW, 6-115
CustomConversionWaitTime, 6-116
CustomConverterPath, 6-117

D

DatabasePreserveCase, 6-118
DatabaseWildcards, 6-119
DataDir, 6-120
dateCurrent, 6-121
DatedCacheIntervalDays, 6-122
DateOutputFormat, 6-123
dCheckoutUser, 3-28
DCMaxFileSize, 6-124
dcShowExportLink, 6-125
DCTimeOut, 6-126
DCViewFormat, 6-127
dDocAccount, 3-27
dDocAuthor, 3-27
dDocFormats, 3-28
dDocID, 3-28
dDocName, 3-27, 3-36
dDocTitle, 3-27
dDocType, 3-27, 3-36
DebugMode, 6-128
DebugStdConversion, 6-129
Default Accounts, 6-130
DefaultAuth, 6-131
defaultFieldInclude, 6-132
DefaultFilterInputFormat, 6-133
DefaultFilterOutputFormat, 6-134
DefaultGraphicTimeout, 6-135
DefaultHtmlConversion, 6-136

DefaultHtmlRefreshTimeoutInSeconds, 6-137
DefaultMasterDomain, 6-138
DefaultNativeTimeout, 6-139
DefaultNetworkAccounts, 6-140
defaultOptionListScript, 6-142
DefaultPasswordEncoding, 6-143
DefaultPostscriptTimeout, 6-144
deleteActionItemInterval, 6-145
deleteWatchdogInterval, 6-146
DelimitedUserRoles, 6-147
dExtension, 3-28, 3-36
dFileSize, 3-28
dFlag1, 3-28
dFormat, 3-28
dID, 3-28, 3-36
dInDate, 3-27, 3-37
dIndexerState, 3-29
DirectoryLockingLogPath, 6-148
DisableAmberLayouts, 6-149
DisableClassicLayoutPublishing, 6-150
DisableCollegiateSkinPublishing, 6-151
DisableContentProfiles, 6-152
DisableDBLongIntegerField, 6-153
DisableForcedConversions, 6-154
DisableGzipCompression, 6-155
DisableHttpUploadChunking, 6-156
DisableKeepingForcedConversions, 6-157
DisableMeetingCenter, 6-158
DisableOptimizedLatestReleaseQuerying, 6-159
DisableSharedCacheChecking, 6-160
DisableStellent05SkinPublishing, 6-161
DisableStellentSkinPublishing, 6-162
DisableTopMenusPublishing, 6-163
DisableTraysPublishing, 6-164
DisableWindowsPublishing, 6-165
dIsCheckedOut, 3-29
DisplayPSOrigin, 6-166
dIsPrimary, 3-29
DistillerNormJobSetting, 6-167
DistillerOptJobSetting, 6-168
DistillerPath, 6-169
dIsWebFormat, 3-29
dLocation, 3-29
dMessage, 3-30
DoAutoMaxMerge, 6-170
docAccounts リスト , 3-33
docAuthors リスト , 3-33
DocConverterEngineDir, 6-171
docLoadResourceIncludes, 6-172
docRootFilename, 6-175
DocTypeSelected, 6-176
docTypes リスト , 3-33
DocUrl, 6-177
docUrlAllowDisclosure, 6-178
DoDocNameOrder, 6-179
DomainControllerName, 6-180
dOriginalName, 3-30
dOutDate, 3-27
DownloadApplet, 6-181
DownloadSuggestedName, 6-182
dpGet, 6-183
dpPromote, 6-184
dpPromoteRs, 6-185
dProcessingState, 3-30
dpSet, 6-186
dPublishState, 3-31
dPublishType, 3-31
dReleaseState, 3-31
dRendition1, 3-31
dRendition2, 3-32
dRevClassID, 3-32
dRevLabel, 3-27
dSecurityGroup, 3-27, 3-36
dStatus, 3-32
DtmPort, 6-187
dWebExtension, 3-32
dWfName, 6-188
dWfStepName, 6-189
Dynamic Converter, 5-17
DynamicConverterHtmlFlavor, 6-190
dynamichtml, 3-4

E

elseif 条件 , 3-11
else 条件 , 3-11
EmptyAccountCheckinAllowed, 6-191
EnableCopyConvertCopy, 6-192
EnableDocumentHighlight, 6-193
EnableEnvironmentPackager, 6-194
EnableErrorFile, 6-195
EnableExpirationNotifier, 6-196
EnableOptimizedLatestReleaseQuerying, 6-198
EnableRebuildForcesReconversion, 6-199
EnableSchemaPublishing, 6-200
encodeHtml, 6-201
endif 条件 , 3-11
endloop, 3-15

Enterprise Search, 5-24
EnterpriseSearchAsDefault, 6-203
entryCount, 6-204
eval, 3-20, 6-205
ExcludeConversionFilter, 6-207
ExclusiveCheckout, 6-208
ExclusiveIncludeConversionFilter, 6-209
exec, 3-19, 3-20, 6-210
executeService, 6-211
Expiration Date, 3-27
ExpireRevisionOnly, 6-213
ExternalUserAccounts, 6-214
ExternalUserRoles, 6-215

F

fieldCaption, 6-216
fieldCaptionInclude, 6-217
fieldCaptionStyle, 6-218
fieldDefault, 6-219
fieldEditWidth, 6-220
fieldEntryInclude, 6-221
fieldExtraScriptInclude, 6-222
fieldInclude, 6-223
fieldIsOptionList, 6-224
fieldMaxLength, 6-225
fieldName, 6-226
fieldOptionListType, 6-227
fieldType, 6-228
fieldValue, 6-229
fieldValueStyle, 6-230
fieldWidth, 6-231
FileEncoding, 6-232
fileUrl, 6-233
FILTER_DEBUG, 6-234
FIRSTREV, 6-235
Folders および WebDAV, 5-26
ForcedConversionRules, 6-236
ForceDistinctRevLabels, 6-237
ForceDocTypeChoice, 6-238
forceExpire, 6-239
ForceJdbcSync, 6-241
ForceSecurityGroupChoice, 6-242
formatDate, 6-243
formatDateDatabase, 6-245
formatDateDisplay, 6-246
formatDateOnly, 6-247
formatDateOnlyDisplay, 6-249
formatDateOnlyFull, 6-250
formatDateWithPattern, 6-251
formatTimeOnly, 6-252
formatTimeOnlyDisplay, 6-253
FormInheritMetaDataFieldList, 6-254
FrameMakerCyclesAllowed, 6-255
FrameMakerexePath, 6-256
FrameMakerTimeout, 6-257

G

GATEWAY_INTERFACE, 6-258
generateUniqueId, 6-259
getCookie, 6-260
GetCopyAccess, 6-261
getDebugTrace, 6-262
getErrorTrace, 6-263
getFieldConfigValue, 6-264
getFieldViewDisplayValue, 6-265
getFieldViewValue, 6-266
getFreeMemory, 6-267
getHelpPage, 6-268
getOptionListSize, 6-269
getParentValue, 6-270
getRequiredMsg, 6-271
getTextFile, 6-272
getTotalMemory, 6-273
getUserValue, 6-274
getValue, 6-275
getValueForSpecifiedUser, 6-278
getViewValue, 6-279
getViewValueResultSet, 6-280

H

hasAppRights, 6-281
hasDocInfo, 3-36
HasExternalUsers, 6-282
HasGlobalUsers, 6-283
HasLocalCopy, 6-284
hasOptionList, 6-285
HasOriginal, 6-286
HasPredefinedAccounts, 6-287
HasSiteStudio, 6-288
HasUrl, 6-289
HeavyClient, 6-290
HelpDir, 6-291
HighlightBegin, 6-293
HighlightEnd, 6-294
HtmlDataInputFilterLevel, 6-295

HTMLEditorPath, 6-296
htmlRefreshTimeout, 6-297
htmlRefreshUrl, 6-298
HTML プレビュー , 5-26
HTTP_ACCEPT, 6-299
HTTP_ACCEPT_ENCODING, 6-300
HTTP_ACCEPT_LANGUAGE, 6-301
HTTP_COOKIE, 6-302
HTTP_HOST, 6-303
HTTP_INTERNETUSER, 6-304
HTTP_REFERER, 6-305
HTTP_USER_AGENT, 6-306
HttpAbsoluteCgiPath, 6-307
HttpAbsoluteCgiRoot, 6-308
HttpAbsoluteWebRoot, 6-309
HttpAdminCgiPath, 6-310
HttpBrowserFullCgiPath, 6-311
HttpCgiPath, 6-312
HttpCommonRoot, 6-313
HttpEnterpriseCgiPath, 6-314
HttpHelpRoot, 6-315
HttpIgnoreWebServerInternalPortNumber, 6-316
HttpImagesRoot, 6-317
HttpLayoutRoot, 6-318
HttpRelativeAdminRoot, 6-319
HttpRelativeCgiRoot, 6-320, 6-321
HttpRelativeWebRoot, 6-321
HttpServerAddress, 6-322
HttpSharedRoot, 6-323
HttpSystemHelpRoot, 6-324
HttpWebRoot, 6-325

I

IDC_Name, 6-326
IdcAdminServerHostName, 6-327, 6-328
IdcAdminServerPort, 6-329
IdcAnalyzeLogDir, 6-330
IdcAuthExtraRequestParams, 6-331
IdcCommandFile, 6-332
IdcCommandLog, 6-333
IdcCommandServerHost, 6-334, 6-335
IdcCommandUser, 6-335
IdcHttpHeaderVariables, 6-336
IdcLocale, 6-338
IdcRegistrationPath, 6-339
IdcServerBindAddress, 6-340
IdcServerSocketQueueDepth, 6-341
Idoc スクリプト , 5-27
    構文 , 3-2
    コメント , 3-3
    使用方法 , 3-3
    タグ , 3-2
if 条件 , 3-11
ImageAlchemyExePath, 6-342
Inbound Refinery, 5-11
    構成ファイル , 4-16, 4-18
inc, 3-21, 6-343
incDynamicConversionByRule, 6-345
incGlobal, 6-346
include, 3-19, 6-347
IncludeXmlFlexionDocType, 6-349
include キーワード , 3-21
incTemplate, 6-350
InDesign, 5-15
IndexableRenditions, 6-351
IndexerAutoWorkInterval, 6-352
IndexerLargeFileSize, 6-353
IndexerPath, 6-354
IndexerRebuildStagingManualCopy, 6-355
IndexerRebuildStagingPath, 6-356
indexerSetCollectionValue, 6-357, 6-457
IndexerStyleDir, 6-358
IndexerTouchMonitorInterval, 6-359
IndexerTraceCycleListChanges, 6-360
IndexerTraceLockFileChanges, 6-361
IndexerTraceStartStop, 6-362
IndexVaultFile, 6-363
InitialColID, 6-365
install_dir/admin/bin/intradoc.cfg, 4-5
install_dir/admin/config/config.cfg, 4-5
install_dir/bin/intradoc.cfg, 4-6
install_dir/config/config.cfg, 4-7
install_dir/data/users/config/filter.hda, 4-14, 4-15
install_dir/IdcRefinery/connections/main/intradoc.cfg, 4-16
install_dir/IdcRefinery/shared/idcrefinery.cfg, 4-18
install_dir/search/search.cfg, 4-15, 4-16
InstanceDescription, 6-366
InstanceMenuLabel, 6-367
IntradocDir, 6-368
IntradocRealm, 6-369
IntradocServerHostName, 6-370
IntradocServerPort, 6-371
isActiveTrace, 6-372
IsAllowIndexRenditions, 6-373
IsAllowIndexVault, 6-374

IsAutoArchiver, 6-375
IsAutoNumber, 6-376
IsAutoQueue, 6-377
IsAutoSearch, 6-378
isCheckin, 6-379
IsCheckinPreAuthed, 6-380
isComponentEnabled, 6-381
IsContentRefineryPresent, 6-382
IsContributor, 6-383
IsCriteriaSubscription, 6-384
IsCurrentNav, 6-385
isDocPage, 6-386
IsDpSubmitErrorFatal, 6-387
IsDynamic, 6-388
IsDynamicConverterEnabled, 6-389
isEditMode, 6-390
IsEditRev, 6-391
isExcluded, 6-392
IsExternalUser, 6-393
IsFailedConversion, 6-394
IsFailedIndex, 6-395
isFalse, 6-396
isFieldExcluded, 6-397
isFieldHidden, 6-398
isFieldInfoOnly, 6-399
isFieldMemo, 6-400
IsFilePresent, 6-401
IsFormsPresent, 6-402
isFormSubmit, 6-403
IsFragmentOnly, 6-404
IsFullTextIndexed, 6-405
isHidden, 6-406
isInfo, 6-407
isInfoOnly, 6-408
IsIntranetAuthOnly, 6-409
IsJava, 6-410
IsJdbc, 6-411
IsJdbcLockTrace, 6-412
IsJdbcQueryTrace, 6-413
IsJspServerEnabled, 6-414
isLatestRevision, 3-36
isLayoutEnabled, 6-415
IsLocalSearchCollectionID, 6-416
IsLoggedIn, 6-417
IsMac, 6-418
IsMaxRows, 6-419
isMultiOption, 6-420
IsMultiPage, 6-421
isNew, 6-422
IsNotLatestRev, 6-423
IsNotSyncRev, 6-424
IsNTCompressionPresent, 6-425
IsOverrideFormat, 6-426
isPhysicallySplitDir, 6-427
IsPrimaryUserBase, 6-428
IsPromptingForLogin, 6-429
IsProxiedServer, 6-430
isQuery, 6-431
isRelocated, 6-432
isRepromptLogin, 6-433
IsRequestError, 6-434
isRequired, 6-435
IsSavedQuery, 6-436
IsSavedWfCompanionFile, 6-437
IsSoap, 6-438
isStrictList, 6-439
IsSubAdmin, 6-440
IsSun, 6-441
IsSysManager, 6-442
IsThumbnailPresent, 6-443
isTrue, 6-444
isUpdate, 6-446
isUploadFieldScript, 6-447
IsUploadSockets, 6-448
IsUserEmailPresent, 6-449
isUserOverrideSet, 6-450
isValidateFile, 6-451
isVerboseTrace, 6-452
IsWebServerPagesOnly, 6-453
IsWindows, 6-454
IsWorkflow, 6-455
IsXml, 6-456
isZoneField, 6-458

J

JdbcConnectionString, 6-459
JdbcDriver, 6-460
JdbcPassword, 6-461
JdbcPasswordEncoding, 6-462
JdbcUseColumnLabelForColumnName, 6-463
JdbcUser, 6-463
js, 6-464
jsFilename, 6-465
JSP, 5-29
JspAdminQuery, 6-466
JspDefaultIndexPage, 6-467
JspEnabledGroups, 6-468

JvmCommandLine, 6-469

L

lastEntryTs, 6-470
LatestReleasedTimestampTimeout, 6-471
lc, 6-472
lcCaption, 6-474, 6-498
LdapAdminDN, 6-475
LdapAdminPassword, 6-476
LdapPort, 6-477
LdapServer, 6-478
LdapSuffix, 6-479
like 演算子 , 3-23
    ワイルドカード , 3-24
LkDisableOnRebuild, 6-480
LkDisableReferenceInfo, 6-481
LkDisallowConversionFormats, 6-482
LkIsSecureSearch, 6-484
LkReExtractOnRebuild, 6-485
LkRefreshBatchSize, 6-486
LkRefreshErrorPercent, 6-488
LkRefreshErrorsAllowed, 6-489
LkRefreshErrorThreshold, 6-490
LMDefaultLayout, 6-491
LMDefaultSkin, 6-492
lmGetLayout, 6-493
lmGetSkin, 6-494
loadCollectionInfo, 6-495
loadDocMetaDefinition, 6-496
loadDocumentProfile, 6-497
loadEnterpriseSearchCollections, 6-499
loadEnterpriseSearchResults, 6-500
loadSchemaData, 6-501
loadSearchOperatorTables, 6-502
loadUserMetaDefinition, 6-503
LocaleArchiverDateFormat, 6-504
LocalGroupServer, 6-505
localPageType, 6-506
LogDir, 6-507
LogMessagePrefix, 6-508
LogName, 6-509
LogWithConnectionName, 6-510

M

MacSupportsSignedApplets, 6-511
MailServer, 6-512
MajorRevSeq, 6-513
MaxArchiveErrorsAllowed, 6-514
MaxCollectionSize, 6-515
MaxConversionCacheSizeInMegs, 6-516
MaxDocIndexErrors, 6-517
MaxErrorsAllowed, 6-518
MaxHitCount, 6-519
MaxIntegerAllowed, 6-520
maxLength, 6-521
MaxMergeBaseCount, 6-522
MaxNumberConvertedHistoryShow, 6-523
MaxNumRecursiveStepDefinitions, 6-524
MaxQueryRows, 6-525
MaxRequestThreadCount, 6-526
MaxResults, 6-527
MaxSavedSearchResults, 6-528
MaxSearchConnections, 6-529
MaxSizeResourceFileCacheMillions, 6-530
MaxStandardDatabaseResults, 6-531
MemoFieldSize, 6-532
MergeInclude, 3-38
MergeInclude を使用したレスポンスのフォーマット , 3-38
Microsoft Office, 5-14
MinIntegerAllowed, 6-533
MinMemoFieldSize, 6-534
MinorRevSeq, 6-535
MrSidLowestCompressionLevel, 6-536
MSIE, 6-537
MSPubexePath, 6-538
MultiUpload, 6-539

N

NetworkAdminGroup, 6-540
NeverTimeoutForcedConversionsTemplateChange, 6-541
NoAutomation, 6-542
noInfoColumn, 6-543
NoMatches, 6-544
noMCPrefill, 6-545
noRevisionsColumn, 6-546
NoSearchConnectionCloseOnDocChange, 6-547
NotationForRequiredFields, 6-548
NotificationIntervalInDays, 6-549
NotificationMaximum, 6-550
NotificationQuery, 6-551
NotifyExtras, 6-553
NotifyTime, 6-554
not 演算子 , 3-25

NtlmSecurityEnabled, 6-555
NumAdditionalRenditions, 6-556
NumConnections, 6-557

O

OdbcDateFormat, 6-558
OneMatch, 6-559
OpenOfficeListenerPort, 6-560
OpenOfficeListenerURL, 6-561
OpenOfficePath, 6-562
OptimizePDF, 6-563
optionListKey, 6-564
optionListName, 6-565
optionListResultSet, 6-567
optionListScript, 6-568
optionListValueInclude, 6-569
optionsAllowPreselect, 6-571
optList, 3-35, 6-572
or 演算子 , 3-25

P

PageMakerExePath, 6-574
PageParent, 6-575
parseClbraProject, 6-576
parseDataEntryDate, 6-577
parseDate, 6-578
parseDateWithPattern, 6-580
PATH_INFO, 6-581
PATH_TRANSLATED, 6-582
PDF Converter, 5-12
    全般 , 5-13
    その他の形式 , 5-15
PdfVersion, 6-583
PLUGIN_DEBUG, 6-584
pneNavigation, 6-585
PostConversionTimeOut, 6-586
PostprocessPDFPath, 6-587
PostscriptPrinterName, 6-588
PowerPointPrintOutput, 6-589
PreconversionPath, 6-590
PreconversionWaitTime, 6-591
PreviewOutputExtension, 6-592
PreviewPath, 6-593
PrimaryWorkQueueTimeout, 6-594
PrinterPortPath, 6-595
Priority, 6-596
prjUseCustomRouting, 6-597
prjWizardTokens, 6-598
prjWizardWorkflowField, 6-599
ProcessExcelUrlHyperlinks, 6-600
ProcessInDesignBookMarks, 6-601
ProcessInDesignHyperlinks, 6-602
ProcessInDesignInternalPDFThumbnail, 6-603
ProcessInDesignLinkUpdate, 6-604
ProcessInDesignPDFCompatibility50, 6-605
ProcessPowerpointHyperlinks, 6-606
ProcessVisioInternalHyperlinks, 6-607
ProcessVisioUrlHyperlinks, 6-608
ProcessWordBookMarks, 6-609
ProcessWordComments, 6-610
ProcessWordEmailLinks, 6-611
ProcessWordFootEndNote, 6-612
ProcessWordHeadings, 6-613
ProcessWordHyperlinks, 6-614
ProcessWordToc, 6-615
ProcessWordUrlHyperlinks, 6-616
ProfileCacheTimeoutInMins, 6-617
ProviderClass, 6-618
ProviderConfig, 6-619
ProviderConnection, 6-620
ProviderDir, 6-621
ProviderType, 6-622
proxiedBrowserFullCgiWebUrl, 6-623
proxiedCgiWebUrl, 6-624
pucDefaultTime, 6-625
pucEnableOnSearch, 6-626
pucHideTimeInDateFields, 6-627
pucUseMonthLocale, 6-628
pucUseWeekdayLocale, 6-629
pucXOffset, 6-630
pucYOffset, 6-631

Q

QUERY_STRING, 6-633
QueueDir, 6-634

R

RecordsManagementDenyAuthorFreePasOnRMSecurity, 6-635
RecordsManagementNumberOverwriteOnDelete, 6-636
RedirectHostsFilter, 6-637
ref
    接頭辞 , 3-36
regexMatches, 6-638

regexReplaceAll, 6-639
regexReplaceFirst, 6-640
Release Date, 3-27
REMOTE_ADDR, 6-641
REMOTE_HOST, 6-642
REQUEST_METHOD, 6-643
RequestAuditAdditionalVerboseFieldsList, 6-644
RequestAuditIntervalSeconds1, 6-645
RequestAuditIntervalSeconds2, 6-646
RequestAuditListDepth1, 6-647
RequestAuditListDepth2, 6-648
RequestAuditSubClassificationField, 6-647
requiredMsg, 6-650
ResultSet, 5-32
ResultSet のループ , 3-13
    例 , 3-14
ResultsTitle, 6-651
Revision, 3-27
RmaAddDocWhereClauseForScreening, 6-652
RMAHideExternalFieldsFromCheckInUpdate, 6-653
RMAHideExternalFieldsFromSearchInfo, 6-654
RmaNotifyDispReviewerAndCatAuthor, 6-655
RmaNotifyReviewerAndAlternateReviewer, 6-656
RolePrefix, 6-657
roles リスト , 3-33
rptDisplayMapValue, 6-659, 6-761
rsAddFields, 6-660
rsAddRowCountColumn, 6-661
rsAppend, 6-662
rsAppendNewRow, 6-663
rsAppendRowvalues, 6-664
rsCopyFiltered, 6-666
rsCreateResultSet, 6-667
rsDeleteRow, 6-668
rsDocInfoRowAllowDisclosure, 6-669
rsExists, 6-670
rsFieldByIndex, 6-671
rsFieldExists, 6-672
rsFindRowPrimary, 6-673
rsFirst, 6-674
rsInsertNewRow, 6-675
rsIsRowPresent, 6-676
rsMakeFromList, 3-35, 6-677
rsMakeFromString, 6-680
rsMerge, 6-682
rsMergeDelete, 6-683
rsMergeReplaceOnly, 6-684
rsNext, 6-685
rsNumFields, 6-686
rsNumRows, 6-687
rsRemove, 6-688
rsRename, 6-689
rsRenameField, 6-690
rsSetRow, 6-691
rsSort, 6-692
rsSortTree, 6-693

S

SafeDir, 6-695
Schema Publisher, 5-33
SchemaAllowRecursiveFileMove, 6-696
SchemaBuildDigestFile, 6-697
SchemaPublishCheckInterval, 6-698
SchemaPublishInterval, 6-699
SchemaPublishMaximumInterval, 6-700
SchemaPublishMinimumInterval, 6-701
SchemaPublishRenameRetryInitialInterval, 6-702
SchemaPublishRenameRetryIterations, 6-703
SchemaPublishRetryInterval, 6-704
SchemaPublishRetryIntervalMax, 6-705
SchemaVerifyPublishing, 6-706
SchemaWin32RenameDelayInterval, 6-707
SCRIPT_NAME, 6-708
ScriptDebugTrace, 6-709
ScriptErrorTrace, 6-710
SearchCacheCleanerInterval, 6-711
SearchCacheHardLimit, 6-712
SearchCacheMaxItemAge, 6-713
SearchCacheSoftLimit, 6-714
SearchCacheTrace, 6-715
SearchConnectionWaitTimeout, 6-716
SearchDebugLevel, 6-717
SearchDir, 6-718
SearchIndexerEngineName, 6-719
SearchQueryFormat, 6-720
SearchQueryTrace, 6-721
SecondaryConversionStepName, 6-722
Security Group, 3-27
securityGroups リスト , 3-33
SelectAccessMemberAllowManualEntry, 6-723
SelectAccessMemberResultLimit, 6-724
SelfRegisteredAccounts, 6-725
SelfRegisteredRoles, 6-726
SERVER_NAME, 6-727
SERVER_PORT, 6-728
SERVER_PROTOCOL, 6-729
SERVER_SOFTWARE, 6-730

setContentType, 6-731
setCookie, 6-732
setExpires, 6-733
setHttpHeader, 6-734
setMaxAge, 6-735
SetPdfVersion, 6-736
setResourceInclude, 6-737
setValue, 6-738
SharedDir, 6-739
SharedWeblayoutDir, 6-740
ShowCreateExternalMenu, 6-741
showDocActions, 6-742
ShowHyperlinkBox, 6-743
ShowOnlyKnownAccounts, 6-744
ShowRevisionsColumn, 6-745
showSearchResultsCheckoutColumn, 6-746
ShowWebDavUrlInBrowserRequest, 6-747
SingleGroup, 6-748
SkipMozillaMailToWorkaround, 6-749
SmtpPort, 6-750
SocketHostAddressSecurityFilter, 6-751
SocketHostNameSecurityFilter, 6-752
SourceID, 6-753
SourcePath, 6-754
SpecialAuthGroups, 6-755
StatusCode, 6-756
StatusMessage, 6-757
StdPageWidth, 6-758
stdSecurityCheck, 6-759
strCenterPad, 6-760
strCommaAppendNoDuplicates, 6-762
strConfine, 6-763
StrConfineOverflowChars, 6-765
strEquals, 6-766
strEqualsIgnoreCase, 6-767
strGenerateRandom, 6-768
strIndexOf, 6-769
strLeftFill, 6-770
strLeftPad, 6-771
strLength, 6-772
strLower, 6-773
strRemoveWs, 6-774
strReplace, 6-775
strReplaceIgnoreCase, 6-776
strRightFill, 6-777
strRightPad, 6-778
strSubstring, 6-779
strTrimWs, 6-781
strUpper, 6-782
StyleForRequiredFields, 6-783
SubjectsDir, 6-784
SubscriptionCollatedFieldList, 6-785
SubscriptionMethod, 6-786
super, 3-19
super タグ , 3-6
　例 , 3-7
SysAdminAddress, 6-787
SystemDateFormat, 6-788
SystemLocale, 6-789
SystemReportsDir, 6-791
SystemResourcesDir, 6-792
SystemTemplatesDir, 6-793
SystemTimeZone, 6-794

T

TempDir, 6-795
TemplateClass, 6-796
TemplateFilePath, 6-798
TemplateName, 6-799
TemplateType, 6-800
TextIndexerFilterExtensionMaps, 6-802
TextIndexerFilterFormats, 6-802
ThumbnailDPI, 6-803
ThumbnailHeight, 6-804
ThumbnailJPEGQuality, 6-805
ThumbnailSizeLimit, 6-806
ThumbnailSizingMethod, 6-807
ThumbnailTemplate, 6-808
ThumbnailWidth, 6-809
TimeoutChangedTemplateForcedConversionsInMins, 6-810
TimeoutMismatchedForcedConversionsInMins, 6-811
TimeoutPerOneMegInSec, 6-812
Title, 3-27
toInteger, 6-813
trace, 6-814
TraceConversionCacheCleanup, 6-816
TraceIsVerbose, 6-817
TraceResourceConflict, 6-818
TraceResourceLoad, 6-819
TraceResourceOverride, 6-820
TraceSectionsList, 6-821
Type, 3-27

U

UieHideSearchCheckboxes, 6-824
UniversalSearchAllowFullText, 6-825
UploadApplet, 6-826
url, 6-827
urlEscape7Bit, 6-828
UseAccounts, 6-829
UseAdobeDistiller, 6-830
UseAdsi, 6-831
UseAlchemy111, 6-832
UseAutoCad2000, 6-833
UseAutocadModelSpace, 6-834
UseBellevueLook, 6-835
UseCascadingStyleSheets, 6-836
UseCustomModaling, 6-837
UseDatabaseShortIndexName, 6-838
UseDCLApplet, 6-839
UseFourDigitYear, 6-840
UseFullGroupName, 6-841
UseGroupFilter, 6-842
UseHtmlOrTextHighlightInfo, 6-843
UseLightweightToPostscriptStep, 6-844
UseLocalGroups, 6-845
UseMicrosoftVM, 6-846
UseNetscape, 6-847
UseNtlm, 6-848
UserAccounts, 6-849
UserAddress, 6-850
UserAppRights, 6-851
UserCache, 6-852
UserCacheTimeout, 6-853
UserDefaultAccount, 6-854
UseRedirectedOutput, 6-855
UseRevisionCreateDate, 6-856
UserFilteredSubjects, 6-857
UserFullName, 6-858
userHasAccessToAccount, 6-859
userHasGroupPrivilege, 6-860
userHasRole, 6-861
UserIsAdmin, 6-862
UserLanguageID, 6-863
UserLocaleId, 6-864
UserName, 6-865
UserPublishCacheDir, 6-866
UserRoles, 6-867
UserSelectAppletColumnOption, 6-868
UserSelectAppletDatabaseCaseSensitive, 6-869
UserSelectAppletDatabaseIso8859DictionaryOrder, 6-870
UserSelectAppletExternalAliasDefaultPermissions, 6-871
UserSelectAppletExternalUserDefaultPermissions, 6-872
UserSelectAppletForceEntriesToLowerCase, 6-873
UserSelectAppletInternalAliasDefaultPermissions, 6-874
UserSelectAppletInternalUserDefaultPermissions, 6-875
UseSearchCache, 6-876
UseSearchML, 6-877
UseSecureLdap, 6-878
UseSelfRegistration, 6-879
UseShortNamesAlways, 6-880
UseSSL, 6-881
UseUserSelectApplet, 6-882
UseVdk4CustomStyle, 6-883
UseXmlEmbeddings, 6-884
UseXmlExtensibleDTD, 6-885
UseXmlNoDTD, 6-886
UseXmlNoReference, 6-887
UseXmlPStyleNames, 6-888
UseXmlSchema, 6-889
UseXmlSchemaDTD, 6-890
UseXmlUrl, 6-891
UseXpedioLook, 6-892
utGetValue, 6-893
utLoad, 6-894
utLoadDocumentProfiles, 6-895
utLoadResultSet, 6-896

V

valueStyle, 6-897
VaultDir, 6-898
VDKSUMMARY, 6-899
VerboseMode, 6-900
Verity, 3-18, 3-37
VerityAppName, 6-901
VerityAppSignature, 6-902
VerityEncoding, 6-903
VerityInstallDir, 6-904
VerityLocale, 6-905
Verity での日付 / 時間 , 3-37

W

WarnAboutTimeZone, 6-906, 6-908
watchdogInterval, 6-907

WebBrowserPath, 6-908
WebLayout Editor Query Design, 3-37
WeblayoutDir, 6-909
WebProxyAdminServer, 6-910
WebServerAuthOnly, 6-911
Web サーバー , 5-42
　変数 , 4-36
Web レイアウト・エディタ , 3-17
wfAction, 6-912
wfAddActionHistoryEvent, 6-913
wfAdditionalExitCondition, 6-914
wfAddUser, 6-915
wfComputeStepUserList, 6-916
wfCurrentGet, 6-917
wfCurrentSet, 6-918
wfCurrentStep, 6-919
wfDisplayCondition, 6-920
wfExit, 6-921
wfGet, 6-922
wfGetStepTypeLabel, 6-923
wfIsFinishedDocConversion, 6-925
wfIsReleasable, 6-927
wfJumpEntryNotifyOff, 6-928
wfJumpMessage, 6-929
wfJumpName, 6-930
wfJumpReturnStep, 6-931
wfJumpTargetStep, 6-932
wfLoadDesign, 6-933
wfMailSubject, 6-934
wfMessage, 6-935
wfNotify, 6-936
wfParentList, 6-937
wfReleaseDocument, 6-938
wfSet, 6-939
wfSetIsNotifyingUsers, 6-940
WfStart, 6-941
wfUpdateMetaData, 6-942
while ループ , 3-14
　例 , 3-15
WorkflowDir, 6-943
WorkflowIntervalHours, 6-944

X

xComments, 3-27
xml, 6-945
XML Converter, 5-16
XMLEncodingMode, 6-946
XMLTimeout, 6-947

Z

ZonedSecurityFields, 6-948

あ

アーカイバ , 3-17, 5-3
アスタリスク（*）ワイルドカード , 3-24
値変数 , 4-34
新しい Idoc スクリプト , 2-1
アンパサンド（&）演算子 , 3-23

い

インクルード , 3-4
　例 , 3-5
インタフェース , 管理 , 3-16

え

演算子 , 3-22
　and, 3-25
　like, 3-23
　not, 3-25
　or, 3-25
　アンパサンド（&）, 3-23
　係数（%）, 3-24
　乗算（*）, 3-24
　除算（/）, 3-24
　数値 , 3-24
　特殊文字列 , 3-23
　パイプ（|）, 3-23
　比較 , 3-22
　ブール , 3-25
　プラス（+）, 3-24
　マイナス（-）, 3-24

お

オプション・リスト , 3-33
　Idoc スクリプト , 3-34
　作成 , 3-35
　内部 , 3-33

か

外部セキュリティ , 5-37

概要
対象読者 , 1-3
関数 , 3-10
eval, 3-20
inc, 3-21
キーワードとの対比 , 3-19
パーソナライズ , 3-10
ワークフロー , 4-38, 5-44
カンマ , セパレータとして使用 , 3-8
カンマをセパレータとして使用 , 3-8
管理インタフェース , 3-16

き

キーワード , 3-18, 5-27
exec, 3-20
include, 3-21
関数との対比 , 3-19
キーワード active, 3-18
キーワード env, 3-18
キーワード local, 3-18
疑問符（?）ワイルドカード , 3-24
共通フィールド表示変数 , 4-27

く

クライアント , 5-4
クラスタリング , 5-5
グローバル関数 , 4-21

け

係数（%）演算子 , 3-24
検索 , 5-34

こ

構成 , 1-2
構成ファイル
Inbound Refinery, 4-16
install_dir/admin/bin/intradoc.cfg, 4-5
install_dir/admin/config/config.cfg, 4-5
install_dir/bin/intradoc.cfg, 4-6
install_dir/config/config.cfg, 4-7
install_dir/data/users/filter.hda, 4-14, 4-15
install_dir/IdcRefinery/connections/main/
intradoc.cfg, 4-16
install_dir/IdcRefinery/shared/idcrefinery.cfg, 4-18
install_dir/search/search.cfg, 4-15, 4-16
コンテンツ・サーバー , 4-5
構成変数 , 4-3
構文
Idoc スクリプト , 3-2
コメント
Idoc スクリプト , 3-3
コンテンツ・アイテム , 5-6
コンテンツ・サーバー , 5-8
構成ファイル , 4-5, 4-6, 4-7, 4-14, 4-15, 4-16

さ

索引付け , 5-28
作成
オプション・リスト , 3-35
変数 , 3-8
サムネイル , 5-13

し

時間と日付 , 5-19
条件 , 3-11
変数の参照 , 3-9
例 , 3-12
条件付き動的変数 , 4-2
乗算（*）演算子 , 3-24
使用方法
Idoc スクリプト , 3-3
除算（/）演算子 , 3-24

す

数値演算子 , 3-24
スキーマ , 5-33
スクリプト , ワークフロー , 4-37

せ

セキュリティ , 5-36
外部 , 5-37
内部 , 5-36
設定可能変数 , 4-34

そ

その他のフィールド表示変数 , 4-29

た

タグ
    Idoc スクリプト , 3-2
    super, 3-6
単一引用符 , 3-37
単一逆向き引用符 , 3-37

ち

置換順序 , 変数 , 3-9

て

ディレクトリおよびパス , 5-22
データベース , 5-18
デバッグ , 5-20
電子メール , 3-18
テンプレート , 5-40
テンプレート読取り専用変数 , 4-31
    例 , 4-32

と

動的サーバー・ページ
    メタデータの参照 , 3-36
動的サーバー・ページでのメタデータの参照 , 3-36
動的変数 , 4-34
特殊キーワード , 3-18
特殊な文字列演算子 , 3-23

な

内部セキュリティ , 5-36

ね

ネーミング , メタデータ・フィールド , 3-26
ネーミング規則 , 3-2

は

パーソナライズ , 5-31
    関数 , 3-10
パイプ（|）演算子 , 3-23
パスおよびディレクトリ , 5-22
バッチ・ローダー , 3-17, 5-3

ひ

比較演算子 , 3-22
非推奨の Idoc スクリプト , 2-1
日付と時間 , 5-19
標準変数 , 3-9
標準メタデータ・フィールド , 3-27

ふ

フィールド
    オプション・リスト , 3-33
    標準メタデータ , 3-27
    メタデータ , 3-26
フィールド情報変数 , 4-26
    例 , 4-26
フィールド表示 , 5-24
フィールド表示変数 , 4-26
    共通 , 4-27
    その他 , 4-29
    例 , 4-30
ブール演算子 , 3-25
プラス（+）演算子 , 3-24

へ

ページ表示 , 5-30
    変数 , 4-25
ページ変数 , 4-25
変換 , 5-11
    AutoCAD, 5-14
    Compression, 5-18
    Dynamic Converter, 5-17
    Inbound Refinery, 5-11
    PDF Converter, 5-12
    XML Converter, 5-16
変数 , 3-7
    Web サーバー , 4-36
    値 , 4-34
    値の割当て , 3-8
    オプション・リスト , 3-34
    関数 , 3-34
    カンマをセパレータとして使用 , 3-8
    共通フィールド表示 , 4-27
    構成 , 4-3
    作成 , 3-8
    参照 , 3-8
    条件付き動的 , 4-2
    設定可能 , 4-34

置換順序 , 3-9
テンプレート読取り専用 , 4-31
テンプレート読取り専用の例 , 4-32
動的 , 4-20, 4-34
標準 , 3-9
フィールド情報 , 4-26
フィールド表示 , 4-26, 4-29
ページ , 4-25
ページ表示 , 4-25
ユーザー読取り専用 , 4-33
読取り専用 , 4-31
ワークフロー , 4-39
変数の参照 , 3-8
条件 , 3-9

ま

マイナス（-）演算子 , 3-24

め

メタデータ
オプション・リスト , 3-33
動的サーバー・ページでの参照 , 3-36
標準フィールド , 3-27
メタデータ・フィールド , 3-26
ネーミング , 3-26

も

文字列 , 5-39
特殊演算子 , 3-23

ゆ

ユーザー , 5-40
ユーザー・トピック , 3-10
ユーザー読取り専用変数 , 4-33

よ

読取り専用変数 , 4-31
テンプレート , 4-31
テンプレートの例 , 4-32
ユーザー , 4-33

り

リスト , オプション , 3-33
関数 , 3-34
作成 , 3-35
変数 , 3-34

る

ループ , 3-13
ResultSet, 3-13
ResultSet の例 , 3-14
while, 3-14
while の例 , 3-15
終了 , 3-15
ループの終了 , 3-15

れ

例
ResultSet のループ , 3-14
super タグ , 3-7
while ループ , 3-15
インクルード , 3-5
条件 , 3-12
その他のフィールド表示変数 , 4-30
テンプレート読取り専用変数 , 4-32
フィールド情報変数 , 4-26

ろ

ローカライゼーション , 5-29

わ

ワークフロー , 5-44
概要 , 4-37
関数 , 4-38, 5-44
スクリプト , 4-37
変数 , 4-39
ワークフロー管理 , 3-16
ワークフローの概要 , 4-37
ワイルドカード , 3-24
アスタリスク（*）, 3-24
疑問符（?）, 3-24