

# ModusOne

ModusOne

Version: 5.2



**perceptive**software

Written by: Product Documentation, R&D  
Date: February 2013

© 2012 Perceptive Software. All rights reserved

ModusOne é uma marca registrada da Lexmark International Technology SA, registrada nos EUA e em outros países.

Perceptive Software é uma unidade de negócios autônoma da Lexmark International Technology SA. Todos os outros nomes de produtos e marcas mencionados neste documento são marcas comerciais ou registradas de seus respectivos proprietários.

Nenhuma parte desta publicação pode ser reproduzida, armazenada em um sistema de recuperação ou transmitida por qualquer outra forma ou por qualquer outro meio, eletrônico, mecânico, fotocópia, gravação ou qualquer outra manifestação de mídia hoje conhecida ou que se torne conhecida no futuro, sem a anterior permissão por escrito da Perceptive Software.

## Conteúdo

Visão geral.....	1
Janela do editor.....	1
Menu principal.....	1
Barra de ferramentas de acesso rápido.....	3
Navegador.....	3
<i>Janela do navegador</i> .....	3
<i>Menu de contexto</i> .....	3
<i>Barra de ferramentas</i> .....	5
Página anterior/Próxima página.....	5
Atualizar.....	6
Mostrar lista de filtros.....	6
Última visualização.....	6
Objetos de importação.....	6
Objetos bloqueados.....	7
Versões de trabalho.....	7
Histórico de transferência.....	7
Objetos excluídos.....	8
<i>Object Inspector</i> .....	8
<i>Caixa de ferramentas</i> .....	9
<i>Janela de saída</i> .....	9
<i>Fonte do sistema</i> .....	9
Visualizações de Odin.....	10
<i>Visualizações</i> .....	10
<i>Processos</i> .....	10
<i>Pilhas</i> .....	11
<i>Trabalhos abertos</i> .....	15
<i>Envelopes abertos</i> .....	15
<i>Visualizador do XPS</i> .....	16
<i>Guia Exibição de Odin</i> .....	16
<i>Visão geral dos status de Odin</i> .....	17
Assistente de consulta.....	19
Administração de alias de BD.....	19
<i>Administração de alias de BD</i> .....	19

<i>Criar novo alias de BD</i> .....	19
<i>Configurações de conexão</i> .....	19
<i>Excluir alias de DB</i> .....	19
<i>Exportar alias de BD</i> .....	20
Controle remoto .....	20
<i>Controle remoto</i> .....	20
<i>Monitoramento de um processo hospedado</i> .....	20
Importação de repositório .....	22
Sincronização de sistema .....	23
Administração de sistema.....	23
<i>Introdução</i> .....	23
<i>Criar nova conexão de banco de dados</i> .....	23
<i>Criação de um novo sistema</i> .....	23
<i>Editar conexões de banco de dados e sistemas</i> .....	24
<i>Excluir conexões e sistemas de banco de dados</i> .....	24
<i>Definir um sistema de transferência</i> .....	25
<i>Restaurar sistemas</i> .....	25
Criar um perfil .....	26
Criar uma função .....	26
Criar um mapeamento de função .....	27
Introdução .....	28
Criar um bloco de texto.....	28
Variáveis de bloco de texto .....	29
<i>Introdução</i> .....	29
<i>Inserir variáveis de bloco de texto</i> .....	29
<i>Conteúdo de variáveis especiais</i> .....	34
<i>Definição de regras de variáveis</i> .....	35
<i>Máscara de entrada</i> .....	36
<i>Variáveis manuais</i> .....	38
<i>Propriedades da variável</i> .....	38
Propriedades gerais de variável.....	38
Condição da variável.....	45
Eventos da variável .....	45
Pesquisar.....	45

O controle de conteúdo EmptyLine .....	49
Atribuir um modelo de dados.....	50
Inserir um bloco de texto filho.....	50
Aplicar propriedade de bloco de texto e variável .....	51
Vários blocos de texto.....	51
Pools de variável .....	52
Objetos de variável.....	52
Herança de atributos de bloco de texto .....	53
Transferir propriedades .....	54
Propriedades de bloco de texto.....	54
<i>Geral.....</i>	<i>54</i>
<i>Condições.....</i>	<i>58</i>
<i>Modelo de dados.....</i>	<i>58</i>
<i>Eventos de bloco de texto.....</i>	<i>58</i>
<i>Definições de objetos .....</i>	<i>58</i>
Introdução .....	60
Documentos dinâmicos .....	60
<i>Criar um documento dinâmico.....</i>	<i>60</i>
<i>Menu de contexto da janela do editor.....</i>	<i>60</i>
<i>Funções da ferramenta do editor .....</i>	<i>61</i>
Documentos estáticos.....	62
<i>Criar um documento estático .....</i>	<i>62</i>
<i>Documentos TIFF estáticos.....</i>	<i>63</i>
<i>Funções da ferramenta do editor .....</i>	<i>63</i>
Criar o controle de conteúdo 'CopyLabel' .....	64
Inserir um bloco de texto .....	64
Atribuir um modelo de dados.....	64
Objetos de variável.....	65
Visualização do documento .....	65
Variáveis globais.....	66
<i>Variáveis globais.....</i>	<i>66</i>
<i>Propriedades de variáveis globais.....</i>	<i>66</i>
Grupos e ações .....	67
<i>Introdução.....</i>	<i>67</i>

<i>Grupos</i> .....	67
<i>Ações</i> .....	67
Transferir propriedades .....	68
Teste de documentos .....	68
Propriedades do documento .....	69
<i>Geral</i> .....	69
<i>Condições</i> .....	70
<i>Modelo de dados</i> .....	71
<i>Eventos do documento</i> .....	71
<i>Definições de objetos</i> .....	73
Introdução .....	75
Criar um pacote .....	75
Cópias de documento .....	76
Atribuir um modelo de dados .....	76
Encaminhar um pacote .....	76
Grupos e ações .....	77
<i>Introdução</i> .....	77
<i>Grupos</i> .....	77
<i>Ações</i> .....	77
Pacotes dinâmicos .....	78
Teste de pacotes .....	78
Caso especial: Atributos de arquivo idênticos para todos os documentos de um pacote .....	80
Transferir propriedades .....	81
Propriedades do pacote .....	81
<i>Geral</i> .....	81
<i>Eventos de pacote</i> .....	81
Introdução .....	83
Criar scripts e condições .....	83
<i>Criar um script</i> .....	83
<i>Criar uma condição</i> .....	83
<i>Conclusão do código</i> .....	85
<i>Definir bookmarks</i> .....	86
<i>Informações de contexto</i> .....	86
<i>Parâmetros do script</i> .....	86

<i>Comentários</i> .....	87
<i>Ferramentas do editor</i> .....	87
<i>Atribuição de scripts e condições</i> .....	88
<i>Scripts de teste e condições</i> .....	89
Funções MonaLisa .....	91
<i>Migração de scripts do modus 4</i> .....	91
<i>Tipos de dados</i> .....	92
<i>Estruturas de controle</i> .....	93
Geral .....	93
CheckEvery .....	93
CheckFirst .....	94
Condição If .....	94
Condição IfElse .....	95
Loops While .....	95
Region .....	95
<i>Tratamento de exceção</i> .....	95
Introdução .....	95
TryFinally .....	95
TryOnError .....	96
TryOnErrorFinally .....	96
<i>Funções padrão</i> .....	97
Acrescentar .....	97
Beep .....	98
BigInteger .....	98
Bof .....	99
CallSystem .....	99
Fechar .....	100
CloseDatabase .....	101
CloseSQL .....	101
Confirmar .....	102
Contém .....	102
Contains_Exactly .....	103
Contagem .....	103
CreatePool .....	103

Decimal .....	104
Excluir.....	105
DeletePool .....	105
Eof .....	106
Execute .....	106
Encontrar .....	107
FindValue.....	107
First.....	109
Flutuante .....	110
Formatar .....	110
FormatStr .....	112
FreeObject.....	112
GetBlobField .....	113
GetConnection .....	113
GetField .....	114
GetLookupValue.....	114
GetObject .....	115
GetVariable.....	115
GetSystemOld.....	116
Índice.....	116
InitDataBase .....	117
InitDataSet.....	117
Inserir.....	118
Integer .....	118
Last.....	119
Max.....	119
Min.....	120
Move.....	120
Próximo .....	121
OpenSQL.....	121
Analisar.....	121
PrepareSQL.....	123
Anterior.....	124
Protocol.....	125



PutField .....	126
Raise.....	126
Leitura .....	126
RegisterLib .....	129
ReRaise .....	129
Rollback .....	129
Round.....	130
Search .....	130
SelFirstRecord.....	131
SelNextRecord .....	131
SelRecordCount.....	131
SelRecordNumber.....	132
SelVariable .....	132
SelVariableEx .....	133
SetExecSQL .....	134
SetPoolVariable.....	138
SetSQL.....	139
Sleep .....	140
SplitStr.....	140
StartTransaction.....	141
StoredProcedure.....	141
StrDelChar .....	142
String .....	142
StringListAdd.....	143
StringListClear.....	143
StringListCount .....	144
StringListGetCommaText .....	144
StringListGetText .....	145
StringListGetValue .....	145
StringListIndexOf .....	145
StringListItemValues .....	146
StringListLine .....	147
StringListLoad.....	147
StringListParams.....	148

StringListSave .....	148
StringListSetCommaText .....	156
StringListSetValue .....	157
StringListSort .....	157
StrLeft .....	158
StrLen .....	158
StrLower .....	158
StrPos .....	159
StrReplace .....	159
StrReverse .....	160
StrRight .....	161
StrTok .....	161
StrToLen .....	162
StrUpper .....	163
StrWord .....	163
StrWordCount .....	164
Trunc .....	164
UnRegisterLib .....	164
Update .....	165
VerifyPool .....	165
Version .....	166
Script de exemplo: acesso ao banco de dados .....	166
<i>Funções Xdata</i> .....	<i>167</i>
Controle de transações .....	167
Informações gerais .....	167
StartTransaction .....	168
Confirmar .....	168
Reverter .....	168
AddDataRecord .....	169
ApplyDataRecord .....	169
ApplyDataSet .....	169
DefineContainer .....	170
Eof .....	170
ExecuteQuery .....	171

First.....	171
GetField .....	171
Last.....	172
Message.....	172
Mensagem.....	173
Próximo .....	173
PrepareParameter .....	173
Anterior.....	174
RowCount .....	175
StartProcedure .....	175
WorkWithQuery .....	176
<i>Funções de data.....</i>	<i>176</i>
Geral.....	176
Métodos da classe MLDate .....	177
CurrentDate .....	177
CurrentDateToString .....	177
DateStringToInt64 .....	178
DayOfWeek.....	178
DaysInMonth .....	179
FormatDateTimeString .....	179
IncDate .....	180
Int64ToDateString .....	180
IsLeapYear.....	181
WeekOfYear.....	181
<i>FileProcessing .....</i>	<i>182</i>
Métodos da classe MLDos .....	182
ChangeExtension .....	182
Combine .....	182
CopyFile.....	183
CreateTextFile .....	183
CreateDirectory .....	184
DeleteDirectory .....	184
DeleteFile .....	184
DeleteFiles .....	185

DirectoryExists.....	185
ExecuteSynchron .....	186
FileAppendText.....	186
FileExists.....	194
FileReadAllText.....	195
FileSize .....	203
GetCommandLine .....	203
GetCommandLineArgByName.....	203
GetCommandLineArgs.....	204
GetCurrentDirectory .....	204
GetDirectories.....	204
GetDirectoryName.....	205
GetDirectoryRoot.....	205
GetEnvironmentVariable.....	205
GetExtension .....	206
GetFileName.....	206
GetFileNameWithoutExtension .....	207
GetFiles.....	207
GetFileSystemEntries .....	208
GetFullPath.....	208
GetLogicalDrives .....	209
GetParentDirectory.....	209
GetPathRoot.....	209
GetRandomFileName .....	210
GetTempFileName.....	210
GetTempPath .....	211
HasExtension.....	211
IsPathRooted.....	211
MoveDirectory .....	212
MoveFile .....	212
ReplaceFile .....	212
SetCurrentDirectory.....	213
Propriedades da classe MLDos.....	213
NewLine .....	213

NewLineLF .....	213
NewLineCRLF .....	213
Métodos da classe MonaLisaTextFile .....	213
Acrescentar .....	213
Fechar .....	221
Criar .....	222
Eof .....	229
Flush .....	230
Abrir .....	230
ReadLine .....	238
ReadToEnd .....	238
Write .....	238
WriteFormat .....	239
WriteLine .....	239
WriteLineFormat .....	239
Propriedades da classe MonaLisaTextFile .....	240
FileName .....	240
NewLine .....	240
Exemplos .....	240
<i>Processamento de XML .....</i>	<i>243</i>
Geral .....	243
Métodos adicionais da classe MLXmlDocument .....	243
GetAttribute .....	243
GetAttributeOfNode .....	243
GetXpathValue .....	244
SetAttribute .....	244
SetAttributeOfNode .....	245
Scripts de exemplo .....	245
Script de exemplo Ler e modificar atributos .....	245
Script de exemplo Criar um arquivo XML .....	247
<i>Processamento de arquivo Ini .....</i>	<i>248</i>
Métodos da classe MLInifile .....	248
Fechar .....	248
CopySection .....	249

DeleteKey .....	249
EraseSection.....	250
GetSectionNames .....	250
Abrir .....	251
ReadBool .....	251
ReadInteger .....	252
ReadSectionToCommatext .....	252
ReadString .....	253
WriteBool .....	253
WriteInteger .....	254
WriteSectionFromCommatext.....	255
WriteString .....	255
<i>MWS Functions .....</i>	<i>256</i>
Geral .....	256
Métodos das classes .....	256
Geral .....	256
ClearCache .....	256
Doc_GetFile_Mime .....	257
Doc_SetFile_Mime.....	257
GetList .....	258
Obj_GetStructure.....	258
Obj_SetStructure .....	259
Obj_Toggle .....	260
Ping.....	260
Process_Close.....	261
Process_Create .....	261
Process_Delete.....	262
Process_Forward.....	263
Process_GetInfo .....	263
Process_GetLastError .....	264
Process_Open .....	265
Process_SetData.....	265
Process_Start .....	266
Métodos da classe Helper .....	267

LoadBase64StringFromBinFile.....	267
SaveBase64StringToBinFile.....	267
Propriedades da classe .....	268
LastBase64DocumentString .....	268
LastProcessId.....	268
LastXmlResult.....	269
MwsUrl.....	269
Exemplo .....	269
Script de exemplo Criar um processo .....	274
<i>Atributos do usuário .....</i>	<i>274</i>
Geral.....	274
Métodos da classe MLUserRepository.....	274
GetCurrentUser .....	274
Métodos do objeto do usuário MonaLisa .....	275
GetAttribute .....	275
HasPermission.....	275
HasObjectPermission.....	280
IsInRole .....	283
Script de exemplo .....	283
Configuração.....	284
Propriedades e métodos de contextos.....	285
<i>Geral.....</i>	<i>285</i>
<i>Contexto ModusBatch.....</i>	<i>285</i>
Propriedades do contexto ModusBatch.....	285
Métodos do contexto ModusBatch .....	285
AssignData.....	285
DeleteDirectory .....	286
GetData.....	286
SetDataContext .....	287
<i>Context MwsProcess .....</i>	<i>287</i>
Geral.....	287
Propriedades do contexto MwsProcess .....	287
Métodos do contexto MwsProcess.....	289
GetActivityStatus.....	289

GetForwardInformation .....	290
GetObjectSelectionParametersRetorna parâmetros de seleção.....	291
GetOptions .....	291
IsInformationRequested .....	291
SetObjectSelectionParameters .....	292
<i>Contexto OdinArchive .....</i>	<i>292</i>
Geral.....	292
Propriedades do contexto OdinArchive.....	292
Métodos do contexto OdinArchive .....	295
HasData .....	295
<i>Contexto OdinConvert.....</i>	<i>296</i>
Geral.....	296
Propriedades do contexto OdinConvert.....	296
Métodos do contexto OdinConvert .....	303
HasData .....	303
<i>Contexto OdinCustomProcess.....</i>	<i>303</i>
Geral.....	303
Propriedades do contexto OdinCustomProcess.....	303
Métodos do contexto TextVariable.....	307
HasData .....	307
<i>Contexto OdinDbTask .....</i>	<i>307</i>
Geral.....	307
Propriedades do contexto OdinDbTask .....	307
Métodos do contexto OdinDbTask.....	311
HasData .....	311
<i>Contexto OdinEmail.....</i>	<i>311</i>
Geral.....	311
Propriedades do contexto OdinEmail.....	312
Métodos do contexto OdinEmail .....	322
AddAttachment.....	322
AddBodyHtmlImage .....	322
AddToHeader .....	323
ClearAttachments .....	323
ClearBodyHtmlImages .....	324



ClearCertificates .....	324
GetAttachments.....	324
LoadBodyFromFile .....	324
LoadCertificateFromCertFile .....	325
LoadCertifacteFromSignedFile.....	325
RemoveAttachment.....	326
RemoveFromHeader .....	326
Send .....	326
SetCredentials .....	327
SetSmtpServer .....	327
Script de exemplo .....	328
<i>Contexto OdinEmailConvert.....</i>	<i>328</i>
Propriedades do contexto OdinEmailConvert.....	328
Métodos do contexto OdinEmailConvert .....	333
SetPdfSigner.....	333
Métodos da interface IOdinStreamDocument.....	334
AddBookmark.....	334
EditPage .....	334
GetInputBin.....	335
Imagem.....	335
IsDuplex .....	336
IsPortrait .....	336
Line .....	336
PageCount.....	337
Publicar.....	338
Rectangle.....	338
SetFont .....	339
SetInputBin .....	339
TextOut .....	340
<i>Contexto OdinFilter.....</i>	<i>341</i>
Geral.....	341
Métodos FilterObject .....	341
Geral .....	341
DefinePostageDefinitionByName .....	341

DefinePostageDefinitionByOld .....	341
DefinePrinterByName .....	341
DefinePrinterByOld .....	342
EnumEnvelopes .....	342
Gen_Stack .....	343
GetParam .....	344
RunFilter .....	344
SetParam .....	345
Visão geral dos possíveis parâmetros dos métodos GetParam e SetParam .....	345
Script de exemplo: Filtro simples incluindo geração de pilha .....	346
Métodos do objeto Envelope .....	348
Geral .....	348
AppendVJob .....	348
CalcEnvelopePages .....	348
CalcEnvelopePapers .....	349
CalcEnvelopePostage .....	349
CalcEnvelopeWeight .....	349
Confirmar .....	350
Contagem .....	350
CreateEnv .....	350
EndTimer .....	351
Eof .....	351
First .....	351
GetField .....	352
GotoJobTillWeight .....	352
GotoRecNum .....	353
Init .....	353
InsertVJob .....	354
Último .....	354
Próximo .....	354
Anterior .....	355
ReadNextEnvelope .....	355
Recnum .....	355
Rollback .....	356

SetField.....	356
Split.....	356
StartTimer .....	357
StartTransaction .....	357
Sample Script: Simple Enveloping Filter with Stack generation .....	357
<i>Contexto OdinPrint.....</i>	<i>361</i>
Propriedades do contexto OdinPrint .....	361
Métodos do contexto OdinPrint .....	364
HasData .....	364
<i>Contexto OdinRendition .....</i>	<i>364</i>
Geral.....	364
Propriedades do contexto OdinRendition .....	365
Métodos do contexto OdinRendition.....	365
GetFieldNameEnumerator .....	365
GetValue .....	365
HasData .....	366
SetValue.....	366
<i>Contexto OdinStream.....</i>	<i>367</i>
Geral.....	367
Propriedades do contexto OdinStream .....	367
Métodos da interface IOdinStream.....	367
AppendDocument .....	367
GetProperty .....	368
LoadDocument .....	371
LoadEnvelopes .....	372
LoadJobs .....	372
SetProperty.....	372
Script de exemplo: LoadEnvelopes / LoadJobs .....	373
Métodos do contexto OdinStream .....	374
HasData .....	374
<i>Contexto OdinStreamDocument.....</i>	<i>374</i>
Geral.....	374
Propriedades de IOdinStreamDocument-Interface .....	375
Métodos da interface IOdinStreamDocument.....	376

AddBookmark.....	376
AddComment .....	377
AddPJLCommand .....	377
EditPage .....	378
GetInputBin.....	378
Imagem.....	379
IsDuplex.....	379
IsPortrait .....	380
Line .....	380
PageCount.....	381
Publicar.....	381
Rectangle.....	381
SetAfpCopyGroup .....	382
SetFont .....	383
SetInputBin.....	383
TextOut .....	384
<i>Contexto OdinStreamPrint .....</i>	<i>385</i>
Propriedades do contexto Odin StreamPrint .....	385
Métodos do contexto StreamPrint.....	388
HasData .....	388
<i>Contexto OnError.....</i>	<i>389</i>
Propriedades do contexto OnError.....	389
<i>Contexto OnNavigateObject .....</i>	<i>389</i>
Propriedades do contexto OnNavigateObject.....	389
Exemplo .....	390
<i>Contexto OnProcessError .....</i>	<i>390</i>
Propriedades do contexto OnProcessError.....	390
<i>Contexto Process.....</i>	<i>393</i>
Geral.....	393
Propriedades do contexto Process .....	393
<i>Contexto ProcessTimer.....</i>	<i>394</i>
Geral.....	394
Propriedades do contexto ProcessTimer .....	395
<i>Contexto SelectionQueryScript.....</i>	<i>396</i>

Propriedades do contexto SelectionQueryScript .....	396
<i>Contexto TextComponent</i> .....	<i>397</i>
Geral .....	397
Propriedades do contexto TextComponent .....	397
Métodos do contexto TextComponent.....	398
LoadExternalDocument .....	398
Navegar .....	399
<i>Contexto TextDocument</i> .....	<i>399</i>
Geral.....	399
Propriedades do contexto TextDocument.....	400
Métodos do contexto TextDocument .....	400
LoadExternalDocument.....	400
Navegar .....	401
<i>Contexto TextDocumentCollection</i> .....	<i>402</i>
Geral.....	402
Propriedades do contexto TextDocumentCollection .....	402
Métodos do contexto TextDocumentCollection.....	403
AddDocumentByName.....	403
AddDocumentByObjectId .....	406
AddDocumentCopyByName .....	408
AddDocumentCopyByObjectID .....	410
Navegar .....	411
SetArchiveAttribute.....	412
SetDataContext .....	412
<i>Contexto TextDocumentReference</i> .....	<i>413</i>
Geral.....	413
Propriedades do contexto TextDocumentReference .....	413
Métodos do contexto TextDocumentReference .....	415
DefinePrinterByName .....	415
DefinePrinterByOld.....	416
SetArchiveAttribute.....	417
<i>Contexto TextVariable</i> .....	<i>417</i>
Geral.....	417
Propriedades do contexto TextVariable .....	417

Métodos do contexto TextVariable.....	418
Navegar .....	418
Ordem do script .....	419
Informações gerais.....	424
Criar um trabalho de transferência.....	424
Exportar objetos .....	425
Introdução .....	426
Caixa de texto com máscara de entrada.....	426
Caixa de texto com opções .....	427
Calendário pop-up .....	428
Campo de entrada numérico .....	429
Introdução .....	431
Criar um processo.....	431
<i>Criar um processo .....</i>	<i>431</i>
<i>Ferramentas do designer de processo .....</i>	<i>431</i>
<i>Propriedades de processos padrão .....</i>	<i>433</i>
<i>Propriedades de um processo MWS .....</i>	<i>433</i>
Designer de processo - Componentes .....	434
<i>Áreas de um bloco de processo .....</i>	<i>434</i>
<i>Posicionar e conectar blocos de processo.....</i>	<i>435</i>
<i>Verificar.....</i>	<i>439</i>
<i>Condição.....</i>	<i>440</i>
<i>IF-Check .....</i>	<i>441</i>
<i>IF-ELSE-Check.....</i>	<i>441</i>
<i>Comentário .....</i>	<i>442</i>
<i>Loop.....</i>	<i>442</i>
<i>Execução de script.....</i>	<i>442</i>
<i>Temporizador universal .....</i>	<i>443</i>
Componentes do designer de processo.....	443
<i>Geral.....</i>	<i>443</i>
<i>Componentes do Modus.....</i>	<i>443</i>
DocumentCollectionManager .....	443
Informações gerais .....	443
DocumentCollectionManager Properties.....	444

Métodos de DocumentCollectionManager .....	444
AssignData .....	444
CreateDocuments .....	444
DefinePrinterForBundleByName.....	445
GetDataReferenceEnumerator.....	446
GetManualVariables .....	446
GetSelectionParameters .....	447
LoadByAlias .....	447
LoadById.....	447
LoadByName .....	448
PrintAndArchive .....	448
ProcessData.....	449
ResetDocuments .....	449
SetManualVariables .....	449
SetSelectionParameters .....	450
Script de exemplo.....	450
ModusBatchManager.....	450
Informações gerais .....	450
Propriedades de ModusBatchManager .....	450
Métodos de ModusBatchManager .....	451
DoWork .....	451
FinishWork .....	451
MwsAssistantManager.....	452
Informações gerais .....	452
Propriedades do MwsAssistantManager .....	452
Métodos de MwsAssistantManager.....	452
DeleteFinishedProcesses .....	452
Script de exemplo MWSAssistantManager.....	452
MwsActivityStatus .....	453
<i>Componentes do Odin .....</i>	<i>453</i>
OdinArchive.....	453
Informações gerais .....	453
Propriedades de OdinArchive .....	453
Eventos de OdinArchive .....	454

Métodos de OdinArchive.....	454
ArchiveJob .....	454
End .....	455
GetCurrentStackRow .....	455
GetEnvelopeEnumerator.....	456
GetJobEnumerator .....	457
HasEnvelopes .....	458
Start.....	458
Script de exemplo OdinArchive.....	459
OdinConvert .....	461
Informações gerais .....	461
Propriedades de OdinConvert.....	461
Eventos de OdinConvert .....	466
Métodos de OdinConvert .....	467
ConvertDocument.....	467
End .....	467
GetCurrentStackRow .....	467
GetDocumentEnumerator .....	469
GetNextStack.....	469
SetStackError.....	469
Start.....	470
Script de exemplo OdinConvert .....	470
OdinCustomProcess .....	471
Informações gerais .....	471
Propriedades de OdinCustomProcess .....	472
Eventos de OdinCustomProcess .....	472
Métodos de OdinCustomProcess.....	473
End .....	473
GetCurrentStackRow .....	473
GetEnvelopeEnumerator.....	475
GetNextStack.....	475
GetJobEnumerator .....	476
HasEnvelopes .....	477
Process .....	477



ProcessJob .....	477
ProcessJobs .....	478
Start.....	478
Script de exemplo OdinCustomProcess .....	478
OdinDbTasks.....	480
Informações gerais .....	480
Propriedades de OdinDbTasks .....	480
Métodos de OdinDbTasks.....	481
DeleteStack.....	481
DeleteProcessRecords .....	482
SetBackStackState.....	482
Script de exemplo OdinDbTasks.....	483
OdinEmail .....	483
Informações gerais .....	483
Propriedades de OdinEmail.....	484
Métodos de OdinEmail .....	485
AddAttachment .....	485
AddBodyHtmlImage .....	485
AddToHeader.....	486
ClearAttachments .....	486
ClearBodyHtmlImages.....	487
ClearCertificates .....	487
GetAttachments .....	487
GetDeliveryNotificationOptions .....	488
GetMailPriority .....	488
GetSmtpDeliveryMethod .....	489
LoadBodyFromFile.....	489
LoadCertificateFromCertFile.....	489
LoadCertificateFromSignedFile .....	490
RemoveAttachment .....	490
RemoveFromHeader.....	491
Redefinir .....	491
Send .....	491
SetCredentials .....	492

SetDeliveryNotificationOptions .....	492
SetMailPriority .....	493
SetSmtpDeliveryMethod .....	493
SetSmtpServer .....	493
Script de exemplo do OdinEmail .....	494
OdinFileConvert .....	496
Informações gerais .....	496
Propriedades de OdinFileConvert .....	496
Métodos de OdinFileConvert .....	497
AddDocument .....	497
AddFile .....	497
CreatePdfSigner .....	497
ConvertSingleFile .....	504
ConvertXDocToHtml .....	509
ConvertXDocToString .....	510
ConvertXDocToText .....	510
ConvertXDocToXps .....	518
End .....	519
LoadDocument .....	519
NumberOfFilesConverted .....	520
Iniciar .....	520
Script de exemplo OdinFileConvert: 1 arquivo .....	524
Script de exemplo OdinFileConvert: 1-n arquivos .....	525
Script de exemplo OdinFileConvert: xDoc em XPS .....	526
Converter Doc em Docx .....	527
ConvertDocToDocX .....	527
EndDocConvert .....	527
StartDocConvert .....	527
OdinFilterManager .....	528
Informações gerais .....	528
Propriedades de OdinFilterManager .....	528
Métodos de OdinFilterManager .....	528
Executar .....	528
Script de exemplo OdinFilterManager .....	528

OdinPrint .....	529
Informações gerais .....	529
Propriedades de OdinPrint .....	529
Eventos de OdinPrint.....	530
Métodos de OdinPrint .....	531
AddBatchPrintJob.....	531
BatchJobCount.....	531
CurrentPrinter .....	532
Término .....	532
ExecuteBatchPrint .....	533
GetCurrentStackRow .....	533
GetNextStack.....	534
GetPrinterEnumerator .....	534
GetPrintJobEnumerator .....	535
GetMailEnumerator .....	536
GetMailJobEnumerator.....	536
InitializeBatchPrint .....	537
IsEmailPrinter .....	537
IsPrinterActive .....	538
PrintJob.....	538
Iniciar.....	539
Script de exemplo OdinPrint .....	539
OdinProcessImport .....	543
Informações gerais .....	543
Propriedades de OdinProcessImport.....	543
Métodos de OdinProcessImport .....	544
ImportProcessFile.....	544
SearchAndImportProcessFile .....	545
Script de exemplo OdinProcessImport .....	545
OdinRendition .....	546
Informações gerais .....	546
Propriedades de OdinRendition .....	546
Eventos de OdinRendition.....	548
Métodos de OdinRendition .....	548

CreateRenditionForProcess.....	549
CreateRenditionForSelectedProcess .....	549
Script de exemplo OdinRendition.....	550
OdinStream .....	550
Informações gerais .....	550
Propriedades de OdinStream.....	563
Eventos de OdinStream .....	574
Métodos de OdinStream .....	578
AddBookmark .....	578
End.....	579
GetCurrentStackRow .....	580
GetEnvelopeEnumerator.....	581
GetJobEnumerator .....	582
GetNextStack.....	583
SetStackError.....	583
Start.....	583
StreamDocument.....	584
Script de exemplo OdinStream .....	584
OdinStreamPrint.....	585
Informações gerais .....	585
Propriedades de OdinStreamPrint .....	586
Eventos de OdinStreamPrint.....	587
Métodos de OdinStreamPrint.....	587
DefinePrinterByName .....	587
DefinePrinterByOld .....	588
Finalizar .....	588
GetCurrentStackRow .....	589
GetDocumentEnumerator.....	590
GetNextStack.....	591
PrintDocument .....	591
SetStackError.....	591
Start.....	592
Script de exemplo OdinStreamPrint.....	592
<i>Componentes de XData .....</i>	<i>593</i>

DataSelectionManager.....	593
Informações gerais.....	593
Propriedades do DataSelectionManager.....	593
Métodos de DataSelectionManager.....	594
GetData.....	594
GetDataExtended.....	594
LoadSelection.....	595
LoadSelectionByName.....	595
LoadSelectionByOld.....	595
WriteXml.....	596
Script de exemplo DataSelectionManager.....	596
XMLProcessingManager.....	596
Informações gerais.....	596
Propriedades de XmlProcessingManager.....	597
Métodos de XMLProcessingManager.....	598
Cancel.....	598
FileExists.....	598
FileExistsEx.....	599
GetJobAttribute.....	599
NextJob.....	599
StartProcessing.....	600
Terminate.....	600
XwsDataSelectionManager.....	601
Informações gerais.....	601
Propriedades de XwsDataSelectionManager.....	601
Métodos de XwsDataSelectionManagers.....	601
GetData.....	601
GetDataExtended.....	602
Ping.....	603
Script de exemplo XwsDataSelectionManager.....	603
Processos de hospedagem.....	604
<i>Instalação de um serviço Windows.....</i>	<i>604</i>
<i>Desinstalação de um serviço Windows.....</i>	<i>608</i>
Introdução.....	609

Criar definições de consulta.....	609
Propriedades de uma definição de consulta .....	609
Propriedades de colunas de resultados .....	610
Propriedades de condições .....	610
Propriedades de variáveis.....	611
Introdução .....	612
Objetos de dados.....	612
<i>Objetos de dados</i> .....	612
<i>Criar objetos de dados</i> .....	612
<i>Editar objetos de dados</i> .....	613
Editar objetos de dados.....	613
Criar scripts de leitura .....	613
Modificar uma consulta SQL .....	614
Criar consultas SQL adicionais.....	614
Atribuir scripts de leitura .....	615
<i>Inserir variáveis</i> .....	615
<i>Inserir objetos filhos</i> .....	615
<i>Nós de resumo</i> .....	616
<i>Ferramentas do editor de Xdata</i> .....	617
<i>Objetos de dados vazios</i> .....	617
Criar objetos de dados vazios .....	617
Editar ou excluir conjuntos de dados .....	617
Executar um procedimento armazenado em um objeto de dados .....	618
<i>Atualizar objetos de dados</i> .....	619
<i>Propriedades de objeto de dados</i> .....	619
<i>Propriedade da variável do objeto de dados</i> .....	620
Seleções.....	621
<i>Seleções</i> .....	621
<i>Criar seleções ativas</i> .....	621
Criar uma seleçãoSeleções ativas podem ser criadas quando o Xdata foi licenciado corretamente.....	621
Inserir objetos de dados .....	624
Ativar objetos de dados .....	624
Atualizar objetos de dados .....	625
Editar interface de seleção .....	625

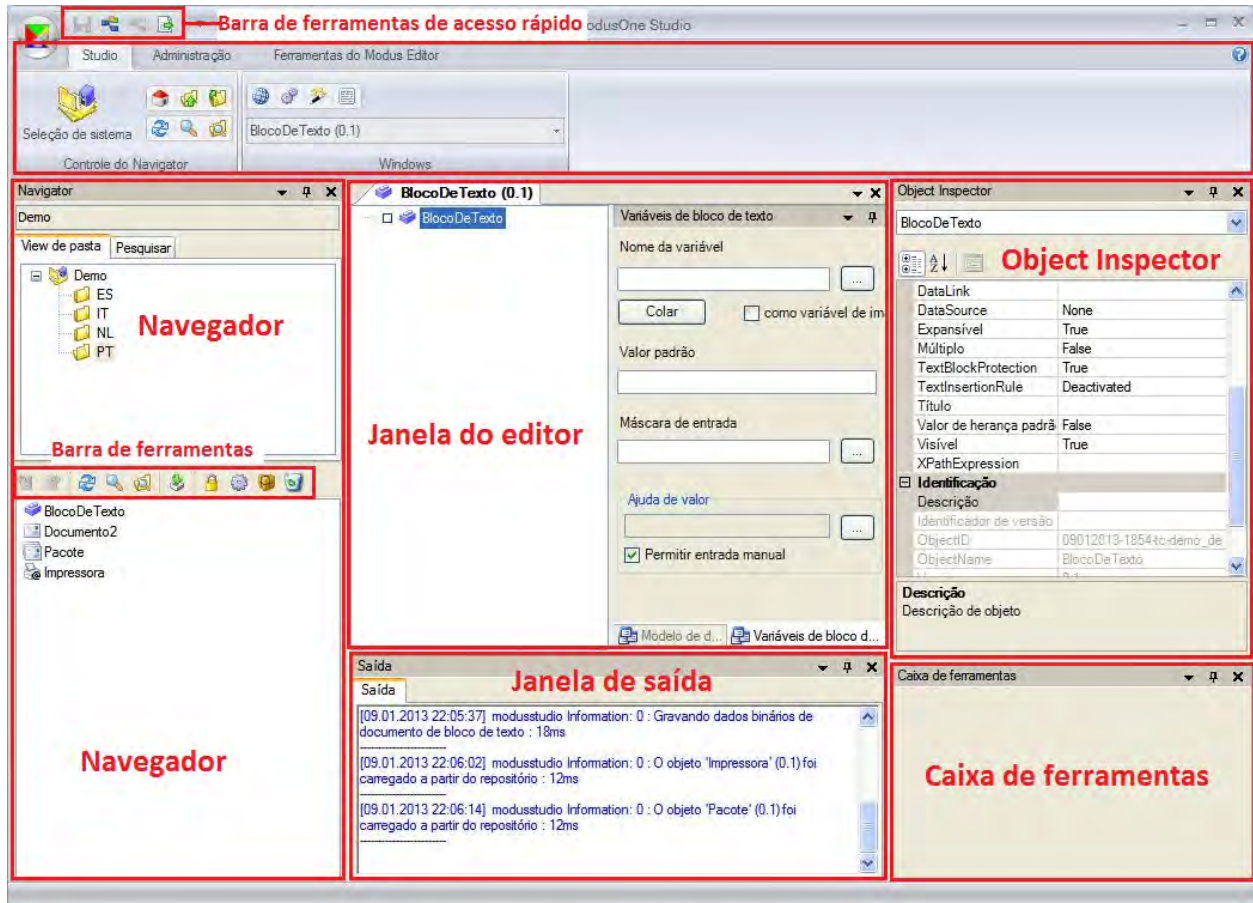
Testar seleções .....	625
Executar uma seleção .....	625
Depurar uma seleção .....	626
Conjuntos de teste .....	626
Gravar e reproduzir recuperação de dados .....	627
Objetos de dados ou atributos ausentes .....	628
<i>Salvar seleção como um esquema .....</i>	<i>628</i>
<i>Criar seleções XSD .....</i>	<i>628</i>
Criar seleções XSD .....	628
Construções de esquema XSD suportadas .....	628
Regras para objetos de dados .....	628
Regras para atributos .....	631
<i>Propriedades de seleção .....</i>	<i>633</i>
Introdução .....	634
Administração.....	634
Envelopes .....	634
Documentos anexos .....	634
Definições de postagem .....	635
Formulários.....	636
Impressoras.....	637
<i>Impressora terceirizada .....</i>	<i>637</i>
<i>Impressora de email .....</i>	<i>638</i>
<i>Impressora online .....</i>	<i>639</i>
<i>Propriedades da impressora .....</i>	<i>640</i>
Geral .....	640
Conversão .....	641
Eventos de impressora .....	641
Configurações de SMTP .....	642
<i>Propriedades de bandeja .....</i>	<i>643</i>
Informações gerais.....	648
Tipos de versão .....	648
Criar uma nova versão .....	649
Alterações a objetos com versão.....	650
Mostrar histórico de objeto .....	650

Reativar uma versão mais antiga .....	651
Transferir objetos com versão para outros sistemas.....	651
Atributos de usuário StudioStartSystemOld e StudioStartFolder .....	651



# Interface do usuário

## Visão geral




## Janela do editor

Trata-se de onde você abrirá e editará todos os objetos no ModusOne Studio.

É possível abrir diversos objetos simultaneamente.

Cada objeto é aberto em sua própria guia.

## Menu principal

O menu principal é acessado clicando no ícone de quatro cores  no canto superior esquerdo.

### Novo

Abre um submenu para a seleção de um novo objeto a ser criado.

### **Cria uma nova versão**

Esse item do menu fica disponível apenas quando o controle de versão é licenciado corretamente.

Consulte: [Criar uma nova versão](#)

### **Reativar versão**

Esse item do menu fica disponível apenas quando o controle de versão é licenciado corretamente.

Consulte: [Mostrar histórico de objeto](#)

### **Salvar**

Salva o objeto atual.

### **Salvar tudo**

Salva todos os objetos abertos.

### **Fechar objeto atual**

Fecha o objeto atual.

### **Fechar todo os objetos**

Fecha todos os objetos abertos.

### **Imprimir**

Esse item de menu permite que você imprima condições, scripts e processos.

### **Visualizar impressão**

Esse item de menu fica disponível para condições, scripts e processos e os exibe em uma visualização de impressão.

### **Limpar cache de studio e tempo de execução**

Se um objeto, por exemplo, um documento ou uma seleção é modificada e você testar as alterações sem salvar o primeiro objeto, uma nova compilação do objeto será criada.

Se você fechar o objeto sem salvá-lo, a nova compilação permanecerá no cache.

Portanto, se o objeto for executado novamente, a compilação armazenada em cache será executada.

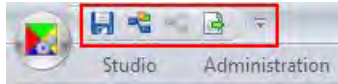
Nesse caso, você deve usar a função **Limpar cache** para executar o objeto em seu estado original.

### **Debug Manager**

O gerenciador de depuração é usado para administrar scripts no serviço de depuração.

Ele também permite que você os remova do serviço.

## Barra de ferramentas de acesso rápido



Além do botão para abrir o menu principal, a barra de ferramentas de acesso rápido também tem vários outros botões usados para acionar funções usadas com frequência.

Você pode usar o menu à direita para ocultar ou exibir esse botões.

### As seguintes funções estão disponíveis:

#### Salvar

Salva o objeto atual.

#### Criar nova versão

Este item de menu só está disponível quando o controle de versão foi licenciado adequadamente.

Consulte: [Criar uma nova versão](#)

#### Reativar versão

Este item de menu só está disponível quando o controle de versão foi licenciado adequadamente.

Consulte: [Reativar uma versão mais antiga](#)

#### Fechar objeto atual

Fecha o objeto atual.

## Navegador

### Janela do navegador

A janela do navegador é usada para administrar todos os objetos ModusOne.

Se o navegador não estiver visível, é possível ativá-lo pressionando o botão no Studio ribbon.



#### Visualização da pasta

O painel superior mostra a estrutura da pasta.

O painel inferior mostra todos os objetos da pasta selecionados no painel superior.

#### Pesquisar

Na guia **Pesquisar** é possível localizar objetos de acordo com seus nomes ou parte dos seus nomes.

### Menu de contexto

O menu de contexto no painel inferior oferece funções de administração de objetos específicos.

#### Novo

Abre um submenu para a seleção de um novo objeto a ser criado.

### **Renomear**

Abra uma caixa de diálogo para inserir o novo nome do objeto.

### **Excluir (Link)**

Esse item do menu está disponível apenas em pastas e exclui o objeto selecionado (link) da pasta.

### **Excluir (Objeto)**

Exclui o objeto selecionado.

Primeiramente, isso significa apenas que o objeto é excluído logicamente, por exemplo, está marcado como excluído no banco de dados e não é mais exibido no Navigator.

O objeto continua no banco de dados com todas as suas versões e, por meio da lista de [Objetos Excluídos](#), pode ser restaurando ou excluído permanentemente e fisicamente.

Enquanto o objeto estiver no banco de dados, é impossível criar um novo objeto do mesmo tipo com o mesmo nome.

### **Os objetos com versão controlada podem ser excluídos logicamente apenas se:**

- não estiverem sendo usados pela versão atual de outro objeto ou
- estiverem sendo usados apenas por objetos que foram, eles mesmos, excluídos logicamente.

### **Observe:**

Objetos de DB Alias e Pasta são excluídos fisicamente automaticamente e não podem ser restaurados.

### **Abrir**

Abre o objeto selecionado.

Alternativamente, todos os objetos podem ser abertos com um simples clique duplo.

### **Contém**

Mostra os objetos de primeiro nível usados pelo objeto selecionado.

### **É usado por**

Mostra uma lista de objetos usando o objeto selecionado.

### **Usado por documento**

Mostra uma lista de documentos usando o bloco de texto selecionado.

### **Copiar**

Copia o objeto selecionado para a área de transferência.

### **Copiar com objetos filhos**

Copia a árvore de objetos para a área de transferência. Um prefixo e/ou sufixo para os novos objetos que serão criados devem ser definidos na caixa de diálogo que é exibida.

### **Copiar para outro sistema**

Abre uma caixa de diálogo para a transferência de objeto. Para obter mais informações, consulte o capítulo Trabalhos de transferência.

### **Inserir link**

Esse item do menu está disponível apenas em pastas e insere o objeto copiado como link na pasta.

### **Inserir**

Insere um objeto copiado.

### **Mostrar histórico de objeto**

Exibe o histórico completo de versão de um objeto. Para obter mais informações, consulte o capítulo Histórico do objeto.

### **Observação:**

Esse item do menu está disponível apenas quando o controle de versão foi licenciado corretamente.

### **Mostrar informações de objeto**

Lista as diversas informações do objeto na janela de saída.

### **Iniciar no ModusOne Client**

Esse item do menu está disponível para os documentos e pacotes e inicia o objeto selecionado no ModusOne Client integrado para fins de teste.

### **Adicionar ao serviço de depuração**

Esse item do menu está disponível apenas para scripts.

### **Remover de serviço de depuração**

Esse item do menu está disponível apenas para scripts.

## **Barra de ferramentas**

Página anterior/Próxima página



Por padrão, todos os objetos no Navigator são exibidos em uma página.

### **Conclua as seguintes etapas para exibir objetos em várias páginas:**

1. Clique no ícone de quatro cores no canto superior esquerdo.
2. Clique no botão **Opções** no canto inferior direito.
3. No diálogo de opções na janela à esquerda, selecione **Configurações gerais**.
4. Na janela à direita, selecione **Ativar paginação para o Navigator**.
5. No campo do tamanho da página, insira o número de objetos a serem exibidos em uma página.

## Atualizar



Essa opção retorna para a visualização padrão, por exemplo, depois de executar a função **É usado por** ou após listar objetos bloqueados.

## Mostrar lista de filtros



Essa função abre uma caixa de diálogo para seleção dos objetos a serem exibidos.

## Última visualização



Se você acabou de usar a função **É usado por** e retornou à visualização padrão, pode usar essa função para exibir o resultado de **É usado por** novamente.

## Objetos de importação



Use esta função para importar arquivos docx existentes como documentos Modus ou blocos de texto.

### Execute as seguintes etapas para fazer isso:

1. Na janela 'Objetos de importação' selecione Bloco de texto ou Documento como tipo de objeto de destino.
2. Para documentos selecione, também, o tipo de documento 'Documento do Word dinâmico' ou 'Documento do Word estático'.
3. Se você estiver usando a função de uma subpasta, selecione a caixa de seleção 'Criar links para a pasta atual'.  
Os objetos importados são vinculados na pasta escolhida.
4. Agora, selecione o diretório de importação de onde os arquivos docx serão importados. O número de arquivos docx no diretório é mostrado.
5. Pressione o botão iniciar. O progresso de importação é exibido.
  - Os objetos criados dão o nome do respectivo arquivo de importação.
  - Se o nome do arquivo de importação contiver caracteres inválidos para os nomes dos objetos Modus, por exemplo, espaços, esses são automaticamente substituídos pelo caractere sublinhado.
  - Quando um objeto é importado com sucesso, o arquivo docx correspondente é renomeado como .docx\_done.
  - Se um objeto não puder ser importado, o nome do arquivo docx permanece inalterado.
  - Antes de cada objeto ser importado, o sistema verifica se um objeto com esse nome já existe. Nesse caso, uma caixa de diálogo é exibida e o usuário pode decidir se um objeto existente deve ser substituído.  
Se o controle de versão for licenciado, a mensagem de confirmação do diálogo resulta na criação de uma nova versão de trabalho do objeto.

Se o novo tipo de documento não for o mesmo que o do objeto existente, o objeto não será importado e uma mensagem correspondente aparecerá.

- Se um arquivo de importação não puder ser aberto porque já foi aberto exclusivamente por outro aplicativo, uma mensagem para esse efeito é exibida.

### Objetos bloqueados



Se o ModusStudio for encerrado inesperadamente enquanto os objetos ainda estiverem sendo editados, esses objetos permanecem bloqueados.

### Conclua as seguintes etapas para desbloquear objetos bloqueados:

1. Clique no botão **Objetos bloqueados** da barra de ferramentas. Uma lista de objetos bloqueados será exibida.
2. Selecione os objetos a serem desbloqueados.
3. Escolha o item de menu de contexto **Desbloquear**.

### Versões de trabalho



Este item de menu só está disponível quando o controle de versão foi licenciado adequadamente.

Essa função exibe uma lista de todos os objetos cujas versões mais recentes são versões de trabalho.

### Execute as seguintes etapas para criar um novo release a partir de uma versão de trabalho:

1. Clique no botão da barra de ferramentas "Versões de trabalho".
2. Selecione os objetos para os quais deseja criar um release.
3. Clique no botão "Criar release"
4. Você pode inserir uma descrição opcional do release na caixa de diálogo seguinte.

### Observação:

O número do release é definido automaticamente.

Não é possível criar um release para objetos bloqueados.

### Histórico de transferência



Essa lista de funções lista todas as transferências executadas nos quais o sistema atual foi o sistema de destino.

Os objetos de transferência da transferência selecionada são exibidos no painel inferior.

### Revogar transferência

Essa função está disponível apenas quando os usuários têm o direito de **Administração** no Modus e a licenças do Modus-Server contém a opção **Controle de versão de objetos**.

Para restaurar uma transferência, selecione-a na lista e clique no botão **Revogar transferência**.

Após a confirmação na consulta seguinte, a transferência é revogada e exibida em uma lista com itens riscados.

As transferências devem ser revogadas na ordem em que foram executadas originalmente. A função não é disponibilizada se você tentar revogar uma transferência na ordem incorreta.

Se um objeto foi criado recentemente em um sistema alvo durante a operação de transferência, a revogação da transferência resultará na exclusão do objeto do sistema de destino.

Se um objeto for restaurado da lixeira durante a transferência, ele é obviamente excluído quando a transferência for revogada ou devolvido à lixeira.

### Objetos excluídos

Essa função lista todos os objetos excluídos anteriormente. Ao fazer isso, a versão mais recente de cada objeto é exibida.

### Restaurar objetos

Para restaurar um objeto, selecione-o da lista e clique no botão **Restaurar objetos**.

Após confirmar a solicitação de confirmação, o objeto se torna visível no navegador novamente e pode ser editado.

### Excluir objetos permanentemente

Essa função está disponível para os usuários apenas quando o direito **Excluir objetos permanentemente (37)** é atribuído.

Pressionar o botão **Excluir objetos permanentemente** excluirá permanentemente o objeto e todas as versões anteriores do banco de dados.

Um objeto pode ser excluído tanto permanentemente quanto fisicamente apenas se não estiver sendo usado por uma versão anterior ou atual de outro objeto.

**Um objeto é classificado como sendo usado por outro objeto quando:**

- O objeto é o objeto filho de outro.  
**Exemplo:**  
um bloco de texto é usado por outro bloco de texto.
- O objeto é referenciado por outro.  
**Exemplo:**  
um script é executado por um evento de bloco de texto.

Você não pode excluir permanentemente um objeto se ele ainda for usado por um objeto excluído logicamente.

### Object Inspector

É possível usar o object inspector para editar propriedades e eventos de objetos abertos.



Se o object inspector não estiver visível, é possível ativá-lo pressionando o botão no Studio ribbon:



### Menu de contexto

O menu de contexto fornece funções de redefinição, criação e edição.

#### Redefinir

Redefine o valor da propriedade do valor padrão ou exclui o valor da propriedade.

#### Criar

Cria um novo script, condição ou definição do objeto variável.

#### Editar

Abre o script ou condição atribuída para a propriedade.

### Caixa de ferramentas

A caixa de ferramentas fornece as ferramentas necessárias para editar determinados objetos. Dependendo do objeto selecionado, a caixa de ferramentas pode ficar vazia.

Se a caixa de ferramentas não estiver visível, você poderá ativá-la pressionando o botão no ribbon do

Studio: 

### Janela de saída

Esse espaço é reservado para o seu conteúdo.

### Fonte do sistema

ModusOne usa o sistema de fonte do Windows **Segoe UI** para exibir todos os textos de interface.

Essa fonte é capaz de exibir caracteres especiais de todas as línguas europeias.

Observe, no entanto, que a fonte não pode ser usada para exibir determinados caracteres matemáticos especiais.

# Administração

## Visualizações de Odin

### Visualizações

As diversas visualizações do Odin podem ser acessadas por meio do menu **Administração** na Barra de tarefas.

### Processos

[▶ Mostrar tudo](#)

#### Visualizar processo

A visualização do processo lista todos os trabalhos de impressão pendentes.

Os dados mostrados são definidos na consulta **ViewProcess**.

Usando **CTRL+C** é possível copiar os conteúdos das linhas selecionadas para a área de transferência.

Clique duplo em um processo para exibir os trabalhos de processo correspondentes na janela inferior.

#### Menu de contexto de visualização de processo

##### Bloquear processo

Define o status do processo para 'bloqueado'.

##### Desbloquear processo

Redefine o campo ODP\_Locked\_by para que o processamento do conjunto de dados possa continuar.

##### Redefinir status para...

Define o status do processo para Importado ou Apresentação criada.

##### Definir status para *Excluir*

Define o status do processo para 'excluir'.

##### Criando pilha a partir de trabalhos abertos...

Cria uma pilha para impressão ou para exclusão.

##### Exibir trabalhos de processos

Exibe os trabalhos de processo na janela inferior.

##### Exibir visualização de erro

Abre outra janela para mostrar todos os erros relacionados.

##### Selecionar colunas

Abre uma caixa de diálogo na qual as colunas exibidas podem ser selecionadas.

##### Classificar por várias colunas

Abre uma janela de diálogo na qual as colunas para os resultados da lista de classificação que são classificadas podem ser selecionadas.

**Atualizar**

Atualiza a lista de processos

**Trabalhos de processo**

Os dados mostrados são definidos na consulta 'ViewProcessJobs'.

**Menu de contexto de trabalhos de processo****Mostrar fluxo de impressão**

Abre o documento correspondente no visualizador XPS.

**Consulte:**

[Visualizador do XPS](#)

**Mostrar pilha**

Mostra a pilha correspondente.

**Selecionar impressora**

Permite alterar a impressora.

**Criar pilha**

Cria uma pilha para impressão ou para exclusão.

Para fazer isso, o usuário deve ter o direito de Criar pilha de Odin para exclusão/impressão na visualização de Odin 'Processes' (3002).

**Selecionar todos**

Selecionar todos os trabalhos do processo especificado.

**Exibir visualização de erro**

Abre outra janela para mostrar todos os erros relacionados.

**Selecionar colunas**

Abre uma caixa de diálogo na qual as colunas exibidas podem ser selecionadas.

**Classificar por várias colunas**

Abre uma janela de diálogo na qual as colunas para os resultados da lista de classificação que são classificadas podem ser selecionadas.

**Atualizar**

Atualiza a lista de processos

## Pilhas

[▶ Mostrar tudo](#)

**Visualização de pilhas**

A visualização de pilhas exibe as pilhas de impressão.

Os dados mostrados são definidos na consulta **ViewStacks**.

Usando a combinação de teclas **CTRL+C** é possível copiar o conteúdo das linhas selecionadas para a área de transferência.

### **Visualização de pilhas do menu do contexto**

#### **Redefinir pilha**

Define o status como **aguardar**.

#### **Liberar pilha**

O termo ou a função liberar pilha se refere à ação de remoção da referência de pilha em trabalhos e envelopes, que os libera para serem processados por uma nova pilha.

#### **Excluir pilha**

Define o status da pilha como **excluir**.

#### **Bloquear pilha**

Define o status como **bloqueada**.

#### **Editar pilha**

Exibe a pilha na janela do editor.

#### **Desbloquear pilha**

Redefine o campo **ODS\_Locked\_by** para que um dataset possa ser processado posteriormente.

#### **Redefinir pilha para fluxo**

Define o tipo de pilha como **fluxo**.

#### **Redefinir pilha para...**

Define o status de uma pilha como o tipo selecionado.

#### **Mostrar fluxo**

Abre o fluxo de impressão em um visualizador de XPS.

#### **Consulte também:**

Visualizador do XPS

#### **Marcar pilha**

Define um marcador para a pilha selecionada no cabeçalho da lista.

#### **Mostrar pilha**

Exibe a pilha em uma nova guia.

#### **Selecionar todas**

Seleciona todas as pilhas da visualização

#### **Mostrar visualizaçãode erro**

Abre outra janela para mostrar todos os erros relacionados.

#### **Selecionar colunas**

Abre uma caixa de diálogo na qual as colunas exibidas podem ser selecionadas.

### **Classificar por várias colunas**

Abre uma caixa de diálogo na qual as colunas dos resultados da lista de classificação podem ser selecionados.

### **Atualizar**

Atualizando a lista de processos

### **Trabalhos de pilha**

Os dados mostrados são definidos na consulta **ViewStackEditJobs**.

### **Trabalhos de pilhas do menu do contexto**

#### **Selecionar impressora**

Permite alterar a impressora.

#### **Redefinir trabalho para...**

Define a pilha com o status **selecionada**.

#### **Mostrar fluxo de impressão**

Abre o fluxo de impressão em um visualizador de XPS

#### **Mostrar pilha**

Mostra a pilha no painel superior.

#### **Mostrar visualizaçãode erro**

Abre outra janela para mostrar todos os erros relacionados.

#### **Remover trabalho**

Remove o trabalho da pilha.

Para fazer isso, o usuário deve ter o direito **Remover trabalho de Odin de pilha/envelope (3512)**.

Além disso, a pilha deve estar definida como **bloqueada**.

#### **Selecionar colunas**

Abre uma caixa de diálogo na qual as colunas exibidas podem ser selecionadas.

### **Classificar por várias colunas**

Abre uma caixa de diálogo na qual as colunas dos resultados da lista de classificação podem ser selecionados.

### **Atualizar**

Atualizando a lista de processos

### **Trabalhos de envelope**

Os dados mostrados são definidos na consulta **ViewStackEditJobs**.

### **Trabalhos de envelope do menu do contexto**

#### **Mostrar trabalhos de envelope**

Exibe os trabalhos do envelope na janela inferior.

#### **Mostrar pilha**

Mostra a pilha no painel superior.

#### **Mostrar visualizaçãode erro**

Abre outra janela para mostrar todos os erros relacionados.

#### **Remover envelope de pilha**

Remove o envelope da pilha.

Para fazer isso, o usuário deve ter o direito **Remover envelope de Odin de pilha (3511)**.

Além disso, a pilha deve estar definida como **bloqueada**.

#### **Selecionar colunas**

Abre uma caixa de diálogo na qual as colunas exibidas podem ser selecionadas.

#### **Classificar por várias colunas**

Abre uma caixa de diálogo na qual as colunas dos resultados da lista de classificação podem ser selecionados.

#### **Atualizar**

Atualiza a visualização

#### **Trabalhos de envelope do menu do contexto**

##### **Selecionar impressora**

Permite alterar a impressora.

##### **Redefinir trabalho para...**

Define a pilha com o status selecionada.

##### **Mostrar fluxo de impressão**

Abre o fluxo de impressão em um visualizador de XPS.

##### **Mostrar pilha**

Mostra a pilha no painel superior.

##### **Mostrar visualizaçãode erro**

Abre outra janela para mostrar todos os erros relacionados.

##### **Remover trabalho**

Remove o trabalho da pilha.

Para fazer isso, o usuário deve ter o direito **Remover trabalho de Odin de pilha/envelope (3512)**.

Além disso, a pilha deve estar definida como **bloqueada**.

##### **Selecionar colunas**

Abre uma caixa de diálogo na qual as colunas exibidas podem ser selecionadas.

##### **Classificar por várias colunas**

Abre uma caixa de diálogo na qual as colunas dos resultados da lista de classificação podem ser selecionados.

### **Atualizar**

Atualiza a visualização

## **Trabalhos abertos**

Lista todos os trabalhos abertos.

Os dados mostrados são definidos na consulta **ViewOpenJobs**.

Usando a combinação de teclas **CTRL+C** é possível copiar o conteúdo das linhas selecionadas para a área de transferência.

### **Menu de contexto de trabalhos abertos**

#### **Mostrar fluxo de impressão**

Abre o fluxo de impressão em um visualizador do XPS

#### **Criando pilha...**

Cria uma pilha para impressão ou para exclusão.

Para fazer isto, o usuário deve ter o direito **Criar pilha de Odin para exclusão/impressão na visualização do Odin 'Trabalhos abertos' (3005)**.

#### **Exibir visualização de erro**

Abre outra janela para mostrar todos os erros relacionados.

#### **Selecionar colunas**

Abre uma caixa de diálogo na qual as colunas exibidas podem ser selecionadas.

#### **Classificar por várias colunas**

Abre uma janela de diálogo na qual as colunas para os resultados da lista de classificação que são classificadas podem ser selecionadas.

### **Atualizar**

Atualizando a lista de processos.

## **Envelopes abertos**

Lista todos os envelopes abertos.

Os dados mostrados são definidos na consulta **ViewOpenEnvelopes**.

Usando a combinação de teclas **CTRL+C** é possível copiar o conteúdo das linhas selecionadas para a área de transferência.

### **Envelopes abertos do menu de contexto**

#### **Mostrar trabalhos de envelope**

Exibe os trabalhos do envelope na janela inferior.

Os dados mostrados são definidos na consulta **ViewOpenEnvelopesJobs**.

### **Exibir visualização de erro**

Abre outra janela para mostrar todos os erros relacionados.

### **Selecionar todos**

Seleciona todos os trabalhos do envelope especificado.

### **Atualizar**

Atualizando a lista de processos

### **Selecionar colunas**

Abre uma janela de diálogo em que as colunas a serem exibidas podem ser selecionadas.

### **Classificar por várias colunas**

Abre uma janela de diálogo na qual as colunas para os resultados da lista de classificação que são classificadas podem ser selecionadas.

### **Liberar**

Libera o envelope e exclui a referência nos trabalhos, criando, assim, trabalhos abertos.

Observação:

O termo **Liberar** se refere à ação de destruir logicamente o envelope e liberando os trabalhos a serem empacotados em outros envelopes lógicos.

## Visualizador do XPS

O Visualizador do XPS permite visualizar os documentos gerados em visualizações do Odin.

### **Visualização de Trabalhos de processo**

O visualizador pode ser acessado a partir do item de menu **Mostrar fluxo de impressão** no menu de contexto. Ele permite a impressão local do documento, além de pesquisar por texto no documento.

### **Visualização de pilhas**

O visualizador pode ser acessado a partir do item de menu **Mostrar fluxo** no menu de contexto. Ele permite a impressão local do documento, além de pesquisar por texto no documento.

Use o botão **Mostrar informações de fluxo** da barra de ferramentas na guia Visualizações do Odin para exibir as informações do arquivo de fluxo.

## Guia Exibição de Odin

As funções neste menu estão disponíveis para todas as visualizações.

### **Atualizar visualização**

Atualiza a visualização.

### **Página anterior**

Rola uma página para trás.

### **Próxima página**



Rola uma página para frente.

### Filtro

Permite a alteração do filtro.

### Condições

Permite a alteração dos parâmetros do filtro atual.

### Informações

A janela de informações exibe os parâmetros da consulta atual.

O campo **DataProviderDefinition** mostra o nome da definição de consulta.

Para obter mais informações, consulte o capítulo [Definições de consulta](#).

## Visão geral dos status de Odin

### Processos

ODP_STATE	Significado
0	Importado
1	Apresentação criada
2	Bloqueado
3	Excluir

### Status da pilha

ODS_STATE	Significado
0	Espera
1	Em andamento
2	Em suspenso (impresso com êxito)
3	Bloqueado (no caso de um erro)
4	Excluir

### Tipo de pilha

ODS_STACKTYPE	Significado
---------------	-------------

0	Imprimir (padrão)
10 *	Otimização de postagem
20 *	Fluxo
30 *	Converter
40 *	StreamPrint
50	Arquivar
60	Transferir (Exemplo de um Processo personalizado)
99	Concluído

\* = O tipo de pilha não pode ser definido e resultará em um erro se for definido em um script.

### Trabalhos

ODS_STATE	Significado
0	Criado (o trabalho foi inserido no banco de dados)
1	Apresentação criada
2	Apresentação de criação de erro
3	Impresso
4	Erro durante a impressão

### Envelopes

ODE_STATE	Significado
0	Criado
1	IT Franking
11	Na impressão
12	Impresso
13	Erro durante a impressão

## Assistente de consulta

O assistente de consulta pode ser acessado por meio do menu **Administração**.

O assistente permite que você execute [Definições de consulta](#) predefinidas.

**Execute as seguintes etapas para fazer isso:**

1. Selecione uma definição de consulta na lista e clique no botão **Próximo**.
2. Selecione a condição desejada e clique no botão **Próximo**.
3. Se a definição de consulta contiver variáveis, defina-as e clique no botão **Próximo**.

A janela final exibe os resultados da consulta.

## Administração de alias de BD

### Administração de alias de BD

Um alias de BD é necessário para criar uma conexão à origem de dados, por exemplo, para um banco de dados.

A administração de alias de BD pode ser acessada pelo menu **Administração**.

Ele permite que você especifique as informações de conexão a todos os bancos de dados usados pelo ModusOne.

### Criar novo alias de BD

**Para criar um novo alias de BD, execute estas etapas:**

1. Clique no botão **Criar novo alias de BD** na guia Ferramentas de administração de alias de BD.
2. Após inserir o nome, especifique as configurações de conexão necessárias e teste-as.
3. O alias de BD agora está disponível.

### Configurações de conexão

**Para verificar as configurações de um alias de BD, execute estas etapas:**

1. Selecione um alias de BD.
2. Selecione **Configurações de conexão** do menu de contexto.
3. Verifique as configurações.
4. Teste a conexão usando o botão **Verificar conexão**.

### Excluir alias de DB

**Para excluir um alias de DB, siga estas etapas:**

1. Clique no botão **Excluir alias de DB** na guia Ferramentas de administração de alias de DB ou selecione **Excluir** do menu de contexto.
2. A entrada selecionada é, primeiramente, marcada para exclusão.
3. Após o salvamento, o alias é então excluído permanentemente. Um alias só pode ser excluído caso não esteja em uso.

## Exportar alias de BD

Para exportar um alias de BD, siga estas etapas:

1. Clique no botão **Exportar aliases de BD como XML** na guia Ferramentas de administração de aliases de BD.
2. Insira o nome do arquivo na caixa de diálogo **Save As**. Todos os aliases de BD serão salvos em arquivo no formato XML.

## Controle remoto

### Controle remoto

O RuntimeRemoteControl pode ser acessado por meio do menu **Administração**.

Isso permite que você monitore e controle os serviços, processos e subprocessos em diversos servidores.

A barra de ferramentas contém botões para iniciar, parar, pausar e continuar.

Botões adicionais também permitem que você atualize a visualização e esvazie o cache de repositório.

Use o botão **Logon** da barra de ferramentas para ativar uma atualização após qualquer alteração de status.

O botão **Sign off** desativa essa funcionalidade.

### Observe:

O usuário conectado ao Windows deve ter direitos de administrador para ter acesso aos servidores remotos.

Usuários sem direitos de administrador não têm permissão para iniciar o módulo de Controle remoto.

## Monitoramento de um processo hospedado

[▶ Mostrar tudo](#)

Para fazer isto, faça as alterações mostradas aqui para os seguintes arquivos de configuração:

### ModusStudio.exe.config

1. Localize a linha **<system.serviceModel>**:

```
<system.serviceModel>
  <client>
    <endpoint name="RemoteControlServiceXWS"
      address="http://localhost:4723/RemoteControlService"
```

```

        binding="wsDualHttpBinding"
        bindingConfiguration="Binding1"
        contract="ModusSuite.Runtime.RemoteControlContract.IRuntimeControlS
ervice" />

```

2. Duplique um dos elementos da extremidade final existente.

3. Altere os seguintes atributos:

Name	Nome do processo.
Address	Insira um número da porta que ainda não tenha sido usado por outra extremidade final.

**A nova entrada pode parecer com isto:**

```

<endpoint name="ModProcess_SVC"
        address="http://localhost:4724/RemoteControlService"
        binding="wsDualHttpBinding"
        bindingConfiguration="Binding1"
        contract="ModusSuite.Runtime.RemoteControlContract.IRuntimeControlS
ervice" />

```

### RemoteControl.config

1. Adicione um novo elemento <service>.
  2. Altere os atributos **name** e **display**.
  3. Para a extremidade final, insira o nome da extremidade final que você acabou de criar.
- Neste exemplo, um computador local com dois serviços serão monitorados:

```

<remotecontrol>
  <computers>
    <computer name="." display="local computer">
      <services>
        <service name="W3SVC" display="" runtime="false" />
        <service name="ModProcess_SVC" display="Modus Process Service
" runtime="true" endpoint="RemoteControlService"/>
      </services>
    </computer>
  </computers>
</remotecontrol>

```

### Atributos e seus significados

name	Nome do serviço como exibido no console de gerenciamento de serviço do Windows (Administration - Services - Service Properties - Display Name)
display	O texto mostrado no console de gerenciamento de serviço do Windows.

runtime	True: False:	Serviço de tempo de execução Serviço normal do Windows
	Também é possível configurar um serviço de tempo de execução com a configuração 'False' se o processo em execução não precisar ser mostrado.  Isso faz sentido, por exemplo, quando um tempo de execução hospeda apenas serviços da web e não tem que executar processos.	
endpoint	O ponto de extremidade para comunicação WCF para um serviço de tempo de execução.	

### Arquivo de configuração Exe do WindowsService

Consulte também: [Instalação de um Serviço Windows](#)

1. Localize a linha `<remotecontrol address="http:..." />` e insira o número da porta do endereço da ponto da extremidade que você definiu no ModusStudio.exe.config, por exemplo.

```
<remotecontrol address="http://127.0.0.1:4724/RemoteControlService" />
```

2. Localize a linha `<system.serviceModel>` e insira o mesmo número de porta nos seguintes elementos:

```
<system.serviceModel>
  <services>
    <service name="ModusSuite.Runtime.RemoteControlService">
      <endpoint
        address="http://127,0,0,1:4724/RemoteControlService"
        binding="wsDualHttpBinding"
        bindingConfiguration="RemoteControlBinding"
        contract="ModusSuite.Runtime.RemoteControlContract.IRuntimeControlService"/>
```

## Importação de repositório

A função **Importação de repositório** podem ser acessada por meio do menu **Administração**.

Essa função é usada para importar os objetos ModusOne exportados anteriormente.

Informações sobre exportação dos objetos ModusOne podem ser encontradas no capítulo [Trabalhos de transferência](#).

### Execute as etapas a seguir para importar os objetos ModusOne:

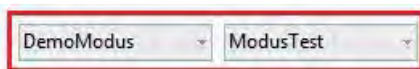
1. No campo **Nome do diretório**, insira o nome do diretório de objetos a serem importados. O diretório deve ser uma subpasta do diretório de trabalho RWS configurado. A configuração padrão desse diretório é `..\modusone\Admin\rws_work\`.
2. Ative a opção **Definir proteção contra gravação** se os objetos devem ser sinalizados como protegidos contra gravação no sistema de destino.
3. Inicie a importação com o botão **Executar** na guia **Importação de objeto**.


## Sincronização de sistema

A sincronização de sistema pode ser acessado no menu Administração.

**Execute as etapas a seguir para sincronizar dois sistemas:**

1. Selecione os sistemas a serem comparados na barra de ferramentas.



2. Clique no botão **Comparar sistemas** na barra de ferramentas.  Uma lista de objetos dos dois sistemas é exibida.
3. Os objetos encontrados em um sistema, mas ausentes no outro, são exibidos em vermelho e podem ser marcados para exclusão no menu de contexto.
4. Os objetos exibidos em azul têm nomes de objeto diferentes em cada sistema.
5. Os objetos que têm números de versão diferentes são exibidos em itálico negro.
6. Os objetos que têm números de versão e nomes diferentes são exibidos em itálico azul.
7. Clique no botão **Executar sincronização** na barra de ferramentas para iniciar a sincronização.



## Administração de sistema

### Introdução

A Administração de sistema pode ser acessada por meio do menu **Administração**.

A administração de sistema é uma ferramenta que oferece suporte gráfico para auxiliar na criação e administração de sistemas, gerenciar as dependências entre os sistemas e para criar conexões de bancos de dados.

### Criar nova conexão de banco de dados

**Para criar uma nova conexão de banco de dados, execute estas etapas:**

1. Arraste o elemento **Novo alias de BD** da caixa de ferramentas para a janela de edição de administração do sistema.
2. Após inserir o nome, especifique as configurações de conexão necessárias e teste-as.
3. O alias de BD agora foi adicionado à administração do sistema.

#### Observação:

Se a caixa de ferramentas não estiver visível, você poderá ativá-la pressionando o botão no ribbon do

Studio: 

### Criação de um novo sistema

**Para criar um novo sistema, execute estas etapas:**

1. Arraste o elemento **Novo sistema** da caixa de ferramentas para a janela de edição de administração do sistema.
2. Defina o nome do sistema e a ID de objeto do sistema na caixa de diálogo.  
A ID de objeto do sistema pode ser editada apenas neste estágio de criação do sistema.  
A ID deve ser exclusiva, ou seja, dois sistemas não podem existir com a mesma ID de objeto.
3. Vincule o sistema a um alias de DB clicando no ponto vermelho do sistema e mantendo o botão esquerdo do mouse pressionado, arrastando-a para um alias de DB.  
Após salvar o vínculo para um alias de DB, nenhuma outra edição é possível.  
Sistemas existentes não podem ser movidos de banco de dados para banco de dados.  
No salvamento, o sistema é atribuído a uma **DefaultAliasTable** e uma **DefaultLookupTable**.  
As duas tabelas são criadas no novo sistema.

#### Observação para a migração modus4:

Como a importação do objeto modus 4 requer um sistema completamente vazio, a DefaultAliasTable e a DefaultLookupTable devem ser excluídas do sistema antes que ele possa ser usado.

#### Para fazer isso, execute as ações a seguir no ModusOne Studio:

1. Abra a administração do sistema.
2. Selecione o sistema e, depois, usando o menu de contexto no object inspector, remova as atribuições nas propriedades **DefaultAliasTable** e **DefaultLookupTable**.
3. Após salvar as alterações, você pode fechar a administração do sistema.
4. Abra o sistema a ser usado para a importação de objetos modus 4.
5. Exclua os objetos **DefaultAliasTable** e **DefaultLookupTable**.

## Editar conexões de banco de dados e sistemas

#### Para editar uma conexão de banco de dados, execute estas etapas:

1. Selecione o alias de BD a ser editado
2. Edite a propriedade 'ConnectionString'.

#### Para editar um sistema, execute estas etapas:

1. Selecione o sistema a ser editado
2. Edite a propriedade em questão.

## Excluir conexões e sistemas de banco de dados

As conexões de banco de dados podem ser excluídas apenas quando não estiverem atribuídas a um sistema.

#### Para excluir as conexões e sistemas de bancos de dados, execute estas etapas:

1. Selecione o objeto a ser excluído.
2. Selecione o item de menu **Excluir objeto** da Barra de ferramentas.



## Definir um sistema de transferência

Sistemas de transferência são retratados por setas vermelhas de um sistema para o outro.

### Para definir um sistema de transferência, execute estas etapas:

1. Dê um clique duplo no sistema de origem.
2. Marque a caixa de seleção do(s) sistema(s) de destino necessários na janela de diálogo seguinte.

### Alternativa:

1. Clique no sistema de origem com o mouse.
2. Clique na seta vermelha do sistema de origem.
3. Mantenha o botão esquerdo pressionado e arraste o cursor para o sistema de destino.

Agora, você pode copiar objetos do sistema de origem para o seu sistema de destino com a função Copiar do Navigator para outro sistema, ou usando [Trabalhos de transferência](#).

## Restaurar sistemas

Se os sistemas ModusOne estiverem armazenados no banco de dados, mas não forem exibidos no momento na ferramenta de administração do sistema, é possível ativar a exibição da seguinte maneira.

### Execute as seguintes etapas para exibir uma lista de todos os sistemas ModusOne em um banco de dados:

1. Abra a administração do sistema.
2. Se não estiver disponível, crie um alias de BD para o banco de dados a ser pesquisado e salve a administração de sistema.
3. No ribbon de administração do sistema, pressione o botão **Restaurar sistemas**.
4. Isso inicia uma pesquisa em todos os bancos de dados com um alias de BD na administração do sistema por sistemas não exibidos no momento na visão geral da administração do sistema.
5. Selecione na lista de sistemas encontrados o(s) sistema(s) que você deseja adicionar à visão geral da administração do sistema.

## Direitos de usuário

### Criar um perfil

Os perfis podem ser usados para agrupar atributos específicos do usuário.

Se o LDAP for usado, os atributos do usuário LDAP são mapeados para os atributos correspondentes.

#### Para criar um novo perfil, execute estas etapas:

1. Selecione **Novo -> Perfil** do menu de contexto do Navigator.
2. Digite o nome do perfil na caixa de diálogo seguinte.
3. O botão **Adicionar atributo** nas ferramentas do editor insere um novo atributo. Os atributos também podem ser inseridos ou excluídos pelo menu de contexto
4. Edite as propriedades do atributo no object inspector:

#### Propriedades do perfil

Propriedade	Definição
Mapping	Nome do atributo do Active Directory, que é atribuído ao atributo.
Value	Valor atribuído à propriedade Mapping que não é preenchida ou a entrada definida ali não existe.

#### Observação:

- Os atributos **UserId** e **UserRoleNames** não podem ser adicionados, pois são criados automaticamente na autenticação.
- No arquivo **UserRepository.config** é definido qual perfil será usado.
- Para aplicar quaisquer alterações, o serviço **Modus\_Core** deve ser reiniciado.

### Criar uma função

As funções são usadas para atribuir direitos e permissões.

#### Para criar um novo perfil, execute estas etapas:

1. Selecione **Novo -> Função** do menu de contexto do Navigator.
2. Digite o nome da função na caixa de diálogo seguinte.
3. Atribua um perfil a uma função arrastando e soltando.
4. Defina as autorizações do sistema na janela central clicando na coluna correspondente ou selecionando um item do menu de contexto.
5. Defina as autorizações do objeto na janela inferior clicando na coluna correspondente ou selecionando um item do menu de contexto.

## Criar um mapeamento de função

Os mapeamentos de função são usados para definir os grupos de usuário.

Eles devem corresponder os nomes de grupo existentes no Windows ou LDAP.

### Para criar um mapeamento de função, execute estas etapas:

1. Selecione **Novo -> Mapeamento de função** do menu de contexto do Navigator.
2. Digite o nome do mapeamento de função na caixa de diálogo seguinte.  
Agora, você pode atribuir as funções adequadas aos grupos de usuário.

### Observação:

- Se um usuário foi atribuído a diversos grupos, os direitos do usuário são uma culminação dos direitos atribuídos aos grupos respectivos.
- O arquivo **UserRepository.config** define qual perfil será usado.
- Para aplicar quaisquer alterações, o serviço **Modus\_Core** deve ser reiniciado.

# Blocos de texto

## Introdução

Os blocos de texto são partes integrais de um documento.

Como uma regra, eles são compostos por um fragmento de texto contendo o texto fixo padrão e/ou variáveis de texto.

A formatação dos blocos de texto são feitas no Word.

Blocos de texto podem ser aninhados uns dentro dos outros em qualquer profundidade desejada.

## Criar um bloco de texto

► [Mostrar tudo](#)

**Para criar um novo bloco de texto, execute estas etapas:**

1. Selecione Novo -> Bloco de texto do menu de contexto do Navigator.
2. Digite o nome do bloco de texto na caixa de diálogo seguinte.

**Na janela do editor, você poderá:**

- Atribuir um modelo de dados
- Inserir uma variável
- Inserir um bloco de texto filho
- Definir as [propriedades](#) do bloco de texto.

**Funções no menu de contexto de blocos de texto**

**Redefinir Data Link**

Exclui o valor da propriedade DataLink.

**Obrigatório**

Altera o status do bloco de texto. O valor da propriedade [TextInsertionRule](#) é definido como **Required**.

**Ativar**

Altera o status do bloco de texto. O valor da propriedade [TextInsertionRule](#) é definido como **Activated**.

**Desativado**

Altera o status do bloco de texto. O valor da propriedade [TextInsertionRule](#) é definido como **Deactivated**.

**Desabilitado**

Altera o status do bloco de texto. O valor da propriedade [TextInsertionRule](#) é definido como **Disabled**.

**Funções na guia Ferramentas do editor**

## Atualizar

Atualiza a estrutura do bloco de texto se os blocos de texto filhos ou seleções foram modificados enquanto o bloco de texto ainda está aberto no modo de edição.

Se as modificações forem feitas a um bloco de texto, a função se mantém desativada. A atualização é feita no salvamento.

## Abrir bloco de texto

Abre o bloco de texto selecionado.

## Adicionar variável

Insere uma nova variável na posição atual do cursor.

## [Aplicar propriedades de bloco de texto e variável](#)

## Mostrar documento do Word

Traz o documento do Word de um bloco de texto para o primeiro plano.

## Alternar visualização

Alterna entre ModusOne em modo de tela cheia e modo de tela dividida com o Word.

# Variáveis de bloco de texto

## Introdução

As variáveis de bloco de texto são contêineres para os dados das variáveis, por exemplo, nome e endereço.

## Inserir variáveis de bloco de texto

[▶ Mostrar tudo](#)

### Conclua as seguintes etapas para inserir uma nova variável em um bloco de texto:

1. Alterne para a guia **Variável de bloco de texto** na janela do editor.
2. As seguintes alternativas de entrada serão disponibilizadas:
  - 1.

N o m e v a r i á v e	<p>Nome das variáveis no bloco de texto.</p> <p>O botão <a href="#">Valor da variável padrão</a> abre uma janela de diálogo para selecionar uma máscara de entrada.</p>
---	---

l:	
Variável de imagem:	<p>Esta opção insere a variável em uma caixa de texto que você pode posicionar livremente no bloco de texto.</p> <p>Insira o caminho e o nome do arquivo de imagem, ou um link para a imagem como valor de variável. Essa imagem será inserida na caixa de texto dinamicamente quando o documento for gerado.</p> <p><b>Exemplo:</b></p> <p>'C:\Logo.jpg' ou</p> <p>'http://www.harmonischearchitectuur.com/portfolio/14-zakbev/high/zakbev409_01_hi.jpg'</p> <p><b>Observação:</b></p> <p>Se um link for definido como valor de variável, a imagem será inserida como um link (por exemplo, não incorporado).</p> <p>Se o documento criado estiver modificado e salvo, a imagem é baixada e salva no documento (incorporado).</p>
	<p><b>Inserção de imagens com código Base64</b></p>
	<p>Aqui, ao invés de inserir o caminho para imagem, é necessário inserir a string com código Base64 como o valor de variável.</p>
	<p>Para garantir que o valor seja reconhecido como Base64, a string é incorporada em uma identificação de string como o exemplo a seguir mostra:</p>
	<p><b>ModBASE64Image*[Base64-Bildcode]*ModBASE64Image</b></p>
	<p>O código de imagem Base64 não pode conter quebras de linha.</p>
	<p><b>Exemplo:</b></p> <pre>&lt;?xml version="1.0" encoding="utf-8"?&gt; &lt;PARTNER xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="Partner_Immobilie.xsd" PAR_Par_Num="0816" PAR_Nachname="Bardot" PAR_Vorname="Berta" PAR_Zusatz="" PAR_Familienstand="1" PAR_Titelcode="4" PAR_Anredecode="2" PAR_Telefon="0816 - 112233" PAR_Fax="0816 - 112230" PAR_Email="Berta.Baeuerlich@hugo.de"&gt;   &lt;ADRESSEN ADR_Par_Num="0816" ADR_Lfd_Num="1" ADR_Gueltig="J" ADR_Typ="D" ADR_Bezeichnung="Hauptadresse" ADR_Strasse="Hauptstraße 1a" ADR_Postfach="" ADR_PLZ="70000" ADR_Ort="Stuttgart" /&gt;   &lt;ANREDE ANR_Anredecode="2" ANR_Adressanrede="Frau" ANR_Briefanrede=" Frau" ANR_Persoendlich="Frau" ANR_Adressanrede_E="Mrs." ANR_Briefanrede_E="Mrs." ANR_Persoendlich_E="Mrs." ANR_Adressanrede_F="Mme" ANR_Briefanrede_F="Madame" ANR_Persoendlich_F="Madame" /&gt;   &lt;TITEL TIT_Titelcode="5" TIT_Bezeichnung="Dipl.-Ing." /&gt;   &lt;Immobilien IM_Code="220-D-986" IM_LivingSpace="200" IM_Location="Cannes" IM_Price="1.000.000" IM_Title="Villa Konstantin" IM_PictureFile="c:\PerceptiveSoftware\modusSuite\data\XMLData\villa_k onstantin.jpg" /&gt;</pre>

<Immobilien IM\_Code="110-A-679" IM\_LivingSpace="180"  
 IM\_Location="Nizza" IM\_Price="800.000" IM\_Title="Villa Belle Vue"  
 IM\_PictureFile="c:\PerceptiveSoftware\modusSuite\data\XMLData\villa\_b  
 ellevue.jpg"></Immobilien>

<Immobilien IM\_Code="132-D-684" IM\_LivingSpace="230"  
 IM\_Location="Miami" IM\_Price="500.000" IM\_Title="Villa Miami Beach"  
 IM\_PictureFile="ModBASE64Image\*/9j/4AAQSkZJRgABAQAAQABAAD//gA7Q1JFQV  
 RPUjogZ2QtanBlZyB2MS4wIChlc2luZyBJSkcgSlBFRyB2NjIpLCBxdWFnZXoZHS5ID0gNzc  
 K/9sAQwAHBQYGBgUHBgYGCAGHCQsSDAsKCgsXEBENEhsXHBwaFxoZHSEqJB0fKCAZGiUy  
 JSgsLS8wLx0jNDg0LjZjLj8u/9sAQwEICAgLcGsgWDAwWlh4aHi4uLi4uLi4uLi4uLi4uLi  
 i4uLi4uLi4uLi4uLi4uLi4uLi4uLi4uLi4uLi4uLi4uLi4uLi4uLi4uLi4uLi4uLi4uLi4u  
 /8IAEQgAlQDIaWEiAAIRAQ  
 MRaf/EABsAAQADAAMBAAAAAAAAAAAAAAAAAAEQBQYBAGMH/8QAGgEBAAMBAQEAAAAAAAAAAAA  
 AAACBAMFBv/aAAwDAQACEAMQAAABpx6WUAAAAAAAAAAAAAAAA7X1Zz6xrpGJAAAFnXQ6iQAA  
 BYeXC8Ry7UvtjGnYu+AqdXcdufzxusd0rGbmZW3zpc6S0YLV9AyNX0X5V9XqOHT5030W0  
 8699C86zgUvVWrip+zp/P0e0Wx5+b9HOtM+kwV1zn7vTxxttPs5iZTXdLyvYYLe4HtS81  
 GW1X00Dyq4to3vzj6P81h9LoNBnqXjs4culeLinuKziLWn0XmafNxx4uutTH1/k09nkmm  
 llqfniG/zFONzN+com40eETH0fOZtE/Qfb5whr6CDNrbWxKaXxtxaK00VHXelotKH19TQ  
 MK618ho5gAAAAD0qtbCP28D1JXh17681nC7drxD85kaHWbA7RNM5ehlijrAAAAAC8rdPg  
 08end52tzxyeB2mLVEkTXuJZddNfHCj0sgAAADngW1jmlfDp0A83WjyOLVjyu08T6cS0x  
 F7SBTID0MmfG7gAAAAAtaq14dNEPG9AB269ieL0AzDh62LPjvzAAAAWhn66Tk8feBx6C  
 J/UvTxjKwzo9jF//EACsQAAlCAQMcbQQCAwAAAAAAAAIDAQQFABESMDMQEYAhMhQVIjEj  
 NCRDYP/aAAgBAQABBQL/AIuIkPDFWSi3VOrPRClaoJjaegFUzG0MDX61hhgrdyxFZN639  
 VPQD5x22dz11V8z25aNTJmwEwFKqlQvSt4WKaBufbKmm4pExKWQ9GKUMTjaerVXjdVi64  
 iWMqTFqqVV4/D7dV1YxaZCnW0y0MZVGGYusUPQaxVSWECePpQABwiPlOULfrEOab8qw1V  
 YbYY3jltVPN8jjH3rIsJVSLNiNYoydduLY1GPqvrTmI/hj4Hbs+ePuOKiIjMvassQ5jUX  
 BicjZKV16rWsjf68LdJy04X+zmv6eDAeF65NWazJanfbNZJZNqRWSTrFCSbtuXCj6zJat  
 OuHEduazyepSOIKCHMPyc4geKbhrGRSjLK9Kq8rDFkvW+/hfqb1cy9aHZK4h9bHW/piLI  
 U9gyladXbEHcRlUkM5GpGrdqDuIyqCH7jU1krarEryVXb7jU1bygyFOyVvVhZGuxKryK9e  
 OFo6+TDjORq6fdm0abNEV2bH+SGTrSHWijKYrM0ioZaq1eRALcxdCAepfECgVggAlNTzN  
 GHGetT22IoGBaK1bzoATM24mTj20f46Hbc3gAN/fVD5gMskNwmNonffSpkFN5uOFOjUpd  
 qFsJRxGnR79WqmNeSvUCMeM94vYvR1B/PqIX5jNuLPQfz/2JL8BKC8bIw1PUq2ASKnecz  
 xP3JPuIxucqjfyz1xbri3VqyyvPVot+fjMb+Ad30ZX59Wh3fSHd9GV+fVod30h3fR1fn1  
 aHd9IR/Jt4c9S/bWRPmz/xAALEQACAgEDAwQDAIAAAAAAAAAAAAAAAAAAQIRAgIRATMgQWUMFC  
 UXH/2gAIAQMBAT8B91K/b7cquukU0jS2OLRyDotmlolGxxa02xRbMOG55fjVmaK1C8RLi2  
 P6J3ZH4LVUSH8o51H7PT5Eo0yzLbkauKFNocmzWxSY5tncZGOSfMUPulyfTbHKTNB3rk  
 Wbtr0IlqkaGKDojBtmW1Llf6eH5dV1n3v5XenpJWhcIssyTalvw+a3ZfLfi8luy+R/8Q  
 AKBEAAgECBgIBBAMAAAAAAAAAAAAAAAAQIAAxEEEHmGITEQMwUiIzAyQlFfx/9oACAECAQE/Afy1  
 gO4DfrbcHaMTSLZb8+HYM95qqoila011hqqBBVDSm4XuCopF5rrGqAC8xunCU7DsxaxUz  
 4+u+jzGH3Iz/VlAi9mUbZeZU/fiBte8pW5vb0YSMgn9T5SgzNmXkQIeph1VaQENMFs0ak  
 GN4tIA3mgsNJSLSRaIBvNBY70qQsxxg0yeDP9hCgRUpqYKSnrexsI+H131GNooRjQLGqAmN  
 UFuJRtkFt+Lq/xHk+aWJyLbfiaK5S3hTYw8nzRooVud+J9Z3Yb1jfiFwD2E9U/8QANxAA  
 AgECAGcGAWYHAAAAAAAAAQIAAxESISiWMTJBUXEQEYAjM2GBkaEEFEJSYnJAQ2BjgpKi/  
 9oACAEBAAAY/Av6LAAuTLkqsXGQcXLVXF1294Ry1N7izjsufwid5a8Q4MOHUrlnwj/uOoz  
 2CccrRMziExFbRaiLzivOYKguJQoou118Wc3D855ZKmdxbTvaebdmnpfWfd6PGeZdzMkw  
 9DEUm6k5GDpCSlyfeE0dFphGQG8ZmpbqZoAoYaLDPh7wNXuW5Qk07Ae8wLlftIqjREKPU  
 JULsmKmxU32iBg7tU4T+Z/tE7710MJ/TeOyGzTl3+cLlDibDClJsJmbvKgzTwlI/3BB0h  
 PfNvQGVz+uU0RyOPKN3jYrHbPsplR12hYe8qs3teX7TTQ6T7ekb9s/yEqVPxXtFHDf7xa  
 hTATWnVI6oLmei/wAoUqDCxWE0Bd56X/Mpj7QmFcYtPhCBSbNoBK1vzym6qSoyNox4Ez7  
 KJURdpWKDTRxm0t383atRR5ijOF6hsLTBTbSvGDi6N9Jctf4Q3uvKd/RbZsnmgo09Sfe  
 KN8uc8y6NPV1Lu76LXMALEdrPUhSgMz+IzEM14iObCMRsMVBdnjv69bAw2CYa4IP5p61/  
 hBSpgrTGZ94Ffe4wvQY4OEGM2PEa+wE4S94xa0ps6g7VygaXLLDcXMcgZ7Jprfk8x0tnI  
 y2vPObc4gXNxxl2F+spuFsrcjxi30kuXWftgtJLE7xzhRN7ZLa4QKuVhmYELaDb0N9vtN  
 xj8IyMdc7sS4UPPTyvcs5zOm15mj/KC+0+2yX12mOF5uzJR2r0mIbQLzLwA60CKv6fCkHS

WtmuXgI4606F2mK1svARiZbC+XGY23jBLhiDPVaZVfPn8fKAZNf21xHgZ7F8KdNceniXp  
 4U6a49PEvhXp/CL27Ju/Xs//8QAKhABAAICAgAEBQUBAQAAAAAAAAQARITFBUTBhcZEQga  
 HB0SBASEHw8WD/2gAIAQEAAT8h/wDFuWRQEAEqMD0eV6P++DvBKQHbCM7tU+DUQc0wS4I  
 qkqUktVIge8AisbPO7r8eD9D/AJmj/dT/AHO/Ap0c0qMZ4EPtZI5JnUXMNBz17lt4ai/C  
 wLT/AL2Kun5sm4OvCq+UXRFFYeYoICTBLYY8xbojMRdpKAF+/BYQX2cVVQUU9F4Z/CreO  
 jvaRj5gGWUOqHoUWloJWkl2KGgZO9EFAX8vhmEC91MCDCCeA0TLIF41IykELPePzLKtUg  
 apr3qmuMA9Zi1lsJPCtsD7ynqEihgLhhPLPvTF04iUBeNxWHJcMBlcIUctquWKKsUeRmb  
 7MkNQHN9oG3Z+Owx+TT46nKi+UQRd6Oon7k+JWh6D2mistd5iFF0ZY4gW5w8PUH5wD5Ew  
 5miMj45TG4LDgqebQyx6nQlbfDnmPJzDutwQiRc2Klmi+iA0R9I/V1F36z3yxHSV1LdGl  
 qoLptVtRQD5IzKI1Y2QqrAtXJCBzB2QS8/QgFRQpEBV5HJEoHtBhvcEWWwqCBdfamuorR  
 XpKI2Y7yOy1KJW0TIGLhrtOolMC4oWMCWlyEZGJLuk0EuBNuFE/KK6KaLp8e+JeikLaeU  
 SwDNZ5iXU3Reo7ntNNOJXKxYL2ATo/LDYK1jqiGeV6ka4345OCXFjDawJtjoP3lZiebfA  
 FmwMxXeD2TpfDZMN7Bfz4mQqQZ6vmBULZvpLLwYPEfBXPjC3uIa5AekaAvvwlFfR8q6Jx  
 HzWXpbY84se9N7iZdtAMxmVLb3EgrlykHguRVnohlWnnxmnK0ecB19U0wlBo+GFnlJR4  
 EgiXo6+FzcpGUJ4rAEoAfpC691+0YGTcrh5cRRCxOE+DiDosFmPF2pMXep16R+ilxKwIK  
 NzyvUp3kz1QQnV9D8T+9myrKut5YpBeaeM3UC/wBACgJ5/D6R/c5nlz9zGfTP7NtvpKlS  
 pUqZq+GeqJUaf2mZJBxVfif/2gAMAwEAAGADAAAAEPPPPPPPPPPPPPPPPPPPPPPPPPPPP  
 +/cP59tc41TJT59Ulp9GQ3uAzntfTrjyF8S3jPPPPPPPPwthLiJ/ADzzzzzE8odtJLzzzz  
 yzj6/r64/zzzzzyn777zwHzzzzzyKACN/wL//EACARAQACAgMAAGMAAAAAAAAAAAAAEAESE  
 xECBBMIFrkfD/2gAIAQMBAT8Q+VNciJt1RN9U8iuKSbl5ubiEI5rGrUUrHqojm6EGLW  
 NtN41k+k8k33HEDL4rgBY+Q/CapN/uVq4HseQAqCUwkJsgFPArYaliCljq9vYxnRlhMpw  
 nc2qYgtPf72W7qC6IdWZYEEgAd7FX1zuxK4Wyu6T44SgagwS3BYX8Qdc7PizqEJvn//xA  
 AkEQEAAGIBBAEFAQAAAAAAAAABABEMXEQIEFRMGGBseHwKf/aAAGBAGeBPxD5d2qAL07  
 TEhtUg08XBvMvDUBAGpfVEGoaxZg6NxVh0PEylmBk7jPDgPp7Zbink/DuICnbXEKhXqaA  
 Zb7DM7DclcOistJkh+fCh5Y++v3PDEMHiNcMwN7DPMALctjcpyKNwkvEoyPsZRiX7cwLb  
 gPzBvGYaJksrg5ZmJBLW0+2ZYsD439P8hNDmIbYNpwQyV8BKgBmX00IPqh9e8Qbv99DuS  
 4skolQuGfv3/25Opv4p/Tnqb66p//EACoQAQEAAGEEAQMEAGMBAAAAAERACEXQVFhcaeE  
 wgZEgscHwENFA4ffg/9oACAEBAAE/EP8A4sqxBqq5YkFsqe8fklbtaW/h9EFAVWAdcEKR  
 axMCEGS7jH6AKw2uFEKd10+ThlhHnLxLs5Cai+i6v4cd2wqyr5wnLBOX0Q/qezPhmf3Hf  
 9DvrBeNYnXfwif76acuhgQI/TOcjQBDK+sscEgNBdLjmcPVU42YEvakoSbuuuK/L/75pM  
 my7ebi4QC2X1lxaqNDH1MRgPsCfOTW3GZeavTJ7j2OngMdOSBCfOP4UQlgo+cBbh8Y  
 tjsUiteUE6pb1u0eMYX2VOPb3ghDzupTZh5D6Wl8jrHZCLDREe/vRw5ZEPg65yigEUK++  
 Xoq086BerLmqRDjh/hvnb8T7Me5vfwSazhCPbjc2qHOLxOmGG23ofDh2s0FRdKHHiY0BS  
 H4b843WxvNAZkbnfeTinc0rj7XBZOWynczZmZbZzs94ogjj4TCQOQp+MacJi4CJMfmhr7  
 mHegM8LiAtexEAv3cILXmqJw4KZ2HXMaFu440R+yDMgZikjG8ey5YZbOpP83XH2clsXzH  
 8581+5nJ/bnK+ABU2Bde78ZIGV0Dw4d4OCFVrsnY95XcEqdXf+M3uczyALPtYnX++I/  
 pGXnxjt5EK2twpuYhDmnP/eaB9FFGu70XConM/tmzDI0i91xWBTT6MAEXQ6xVME8vJNI3  
 lp3jjUUNlBt48mAlBVrwLD9nDRDdZVEMWN6GIHjvuZLIZOHmFOjvyecKvvFzhBODyc02E  
 LaX5BXF9lKvumMJ81NC3JsLeiLs7879GQ5pRwn5w3UmWe9DjnQxViAcMcsPAkvcefjNZf  
 Fbkjb5Cpbq8YAmzWbe48/GDW/pYMEJiien4KEC0dPC5ti9PEI1PUD1HVyd+6Gep6OE3Qy  
 NScl74TSRnBd31MFi3YZsj7u0213sPw4t8gS4hkVmG0PB168ZbZtsc3XD2mAmy8NCC6fN  
 wsA+Ohs8nlzDnwFzbx7nCKEickFQ/HPxgpbIChPAL8/bBBwxI22g88/GBWVI6GhTEYOCP  
 VdH5yCDUztXa/E++JrdXL5Wnlf2xWjKcAZad+euIgoZrv9dTzbynTHkISj16YVFLJVYie  
 UXWc210Rh3Aw/GC3S0gl6HlvwYJ+50StPwlyYSmScxp+zJNncZaIpPEJiIsMSMGj1XJsS  
 x7K7bkYAbcxKe7z9/rQgKchyUBkJTQ1zcFYCBz0i8zv4uD1SgeABOmUN5poD5MRgDpa6  
 XwBMJDhG2BXAEpQdBwRfWPUJbDW/OCU2nKvSXNIMXAJxi7FfeEJuobL/ePrQyAEGQqfXh  
 PU7uBZ6Aw4Afb/AAzohj2P/ZgerIHWLrATUqu5ip3yTSz3mhsH7ZKQV0aXj+Pq0iFLnRU  
 QXsP/AL+lNNrv4Xdt6C+/OKfSQ57h9s6kMKHIYgVYY+drqmch7+ozUvG8riOVgG+Cf2GE  
 OqxW3rfn9EEQDEjTV9nzktS3ogdQ6HX74bmpu7YpMRFZTIEHCDG4SnwT+0xCBlzU/JCRM  
 8GgXw7+sknMXtn+v0eCEcmAAAAHADmr+3x/ww+Z/VGY3+F/StU4+s+U/Vyf7z9Lp9vrFG  
 PHCuue2e2e2FauV6Ronjp7TJLbLcbTHAVO/wD0yhEkAt6M/9k=\*ModBASE64Image"></  
 Immobilien>  
 </PARTNER>



	<p><b>As seguintes constelações relativas aos arquivos de imagem não são suportadas:</b></p> <ul style="list-style-type: none"> <li>• Inserção de uma string com código Base64 no diálogo para variáveis manuais, se a variável de imagem for definida como uma variável manual.</li> <li>• Inserção de uma string com código Base64 como um valor padrão para uma variável de imagem</li> </ul>
Valor padrão:	O valor usado para preencher a variável se nenhum campo de dados estiver definido.
Máscara de entrada:	<p>Formatação de regras para valores de formatação.</p> <p><b>Consulte também:</b></p> <p>Link TODO de máscara de entrada</p>
Ajuda de valor:	<p>Um valor pode ser atribuído a uma variável manual para auxiliar os usuários enquanto fazem entradas manuais.</p> <p><b>Consulte também:</b></p> <p>Link TODO de ajuda de valor</p>
Permissão de entrada manual:	<p>Especifica se uma variável de bloco de texto pode ser substituída ou se precisa estar preenchida fazendo a seleção de um valor pré-definido de uma ajuda de valor.</p> <p>A propriedade <b>IsFreeInput</b> é definida corretamente.</p>

- 2.
3. Clique no botão **Inserir**. A variável é inserida na posição do cursor correspondente no documento do Word.

Para alterar a ordem das variáveis, arraste e solte a variável na posição desejada na estrutura do bloco de texto ou altere a posição no do documento do Word.

### Funções no menu de contexto das variáveis

#### Redefinição de DataLink

Deleta o valor da propriedade **DataLink**.

#### Redefinição do nome do atributo

Deleta o valor da propriedade **AttributeName**.

#### Obrigatória

Altera o status da variável. O valor da propriedade [Tipo de variável](#) é definida para **obrigatória**.

#### Opcional

Altera o status da variável. O valor da propriedade [Tipo de variável](#) é definida para **opcional**.

#### Livre

Altera o status da variável. O valor da propriedade [Tipo de variável](#) é definida para **livre**.

### Conteúdo de variáveis especiais

#### Saltos de linha, tabulações, hífen incondicionais e espaço incondicional

As seguintes palavras chaves em conteúdo de variáveis permitem inserir saltos de linha, tabulações, hífen incondicionais e espaço incondicional

Conteúdo da variável	é convertido em
\line	Salto de linha <b>Observe:</b> O primeiro espaço após \line (ou seja, "\line ") é ignorado. Se for necessário um espaço após \line, devem ser digitados dois espaços ("\line ").
CRLF	Salto de linha
\tab	Tabulação
\~	Espaço incondicional
\_	Hífen incondicional

Esses conteúdos de variável também podem ser definidos na propriedade de variável **Modo de formatação** como **Formatação específica de tipo**.

**Execute as seguintes etapas para fazer isso:**

1. Selecione a variável
2. No Object Inspector, clique na propriedade **Modo de formatação** e em seguida no botão ao lado da caixa de entrada:
3. Na caixa de diálogo de **Formatação de saída**, selecione o modo de formatação **Formatação específica de tipo**
4. No campo **Regra**, defina o modo de formatação desejado, por exemplo, %s\\line força um salto de linha após o conteúdo da variável.

**Hyperlinks**

Se um conteúdo de variável padrão contiver um hyperlink, o conteúdo de variável deve ser formatado da seguinte forma:

**O conteúdo visível da variável no documento|<http://www.perceptivesoftware.de>**

O conteúdo da variável consiste em duas partes: o conteúdo visível e a URL. As duas partes são separadas por uma barra vertical ("|").

O link pode ser aberto no documento gerado pela combinação de teclas **CTRL + botão esquerdo do mouse** no conteúdo da variável.

**Hyperlinks em variável de imagem**

Se uma variável de imagem contiver um hyperlink, o conteúdo de variável deve ser formatado da seguinte maneira:

**C:\Logo.jpg|<http://www.perceptivesoftware.de>**

O conteúdo de variável também deve consistir em duas partes: o nome e o caminho do arquivo de imagem e a URL.

As duas partes são separadas por uma barra vertical ("|").

O link pode ser aberto no documento gerado pela combinação de teclas **CTRL + botão esquerdo do mouse** na imagem.

**Definição de regras de variáveis**

► [Mostrar tudo](#)

Você pode armazenar máscaras de entrada usadas com frequência em uma [tabela de pesquisa](#).

**Execute as seguintes etapas para fazer isso:**

1. Abra a [Administração do sistema](#).
2. Selecione o sistema para o qual as máscaras de entrada são definidas.
3. Atribua uma tabela de aliases para a propriedade **DefaultAliasTable** no object inspector se a propriedade ainda não estiver definida.
4. Salve e feche a administração do sistema.
5. Abra a tabela de aliases que foi atribuída como **DefaultAliasTable**.
6. Vá para a guia Tabela de aliases de pesquisa e clique no botão Nova chave.
7. Insira o nome da chave **variablestemplate**.

8. Clique na chave **variabletemplate** e, depois, no botão **Nova referência** na guia **Tabela de aliases de pesquisa**.
9. Selecione uma [LookupTable](#).  
Você pode ter que criar uma nova tabela de pesquisa antes.
10. Abra a tabela de pesquisa para a qual a chave **variabletemplate** foi atribuída.
11. Defina as regras de variáveis padrão da seguinte forma:

VarName=VarDefault|"VarInput".

**Exemplo:**

```
Var1=Value1|"!\(999 99\) 99999;1;_";
Var2=Value2|">cc-99999;1;_"
Var3=Value3|"000\ 000\ 00;1;_"
```

12. Agora, ao inserir uma nova variável e pressionar o botão para o **valor da variável padrão**  ao lado do campo **Nome da variável**, as entradas na tabela de pesquisa são exibidas para seleção.

## Máscara de entrada

[▶ Mostrar tudo](#)

Máscaras de entrada são usadas para definir formatos para variáveis, por exemplo, data.

Uma máscara de entrada é definida pelo uso de três parâmetros, separados uns dos outros por ponto e vírgula.

### Conclua as seguintes etapas:

1. O primeiro valor é usado para definir caracteres de formatação.  
A seguinte tabela é uma lista de caracteres que podem ser usados para definir uma máscara de entrada e seus significados.

**Caracteres de formatação**

Caractere	Significado na máscara de entrada
!	Especifica que os caracteres opcionais, como os espaços à esquerda, são exibidos. Se '!' não for especificado, caracteres opcionais são exibidos como espaços à direita na máscara.
>	Converte todos os seguintes caracteres em letras maiúsculas até que o fim da máscara seja alcançado ou o caractere '<' seja detectado.
<	Converte todos os seguintes caracteres em letras maiúsculas até que o fim da máscara seja alcançado ou o caractere '>' seja detectado.
<>	Se os dois sinais forem usados juntos, nenhuma verificação de diferenciação de maiúsculas e minúsculas é feita. O dado inserido é formatado exatamente da mesma maneira que o usuário o inseriu.
\	Depois o caractere „\" se torna o caractere a ser usado como um espaço reservado

	na máscara de entrada.
L l	Somente letras são entradas válidas (A-Z, a-z). Uma entrada não é obrigatória.
A a	Somente caracteres alfanuméricos são válidos (A-Z, a-z, 0-9). Uma entrada não é obrigatória.
C c	Qualquer caractere é permitido. Uma entrada não é obrigatória.
0 9	Somente um caractere numérico é permitido. Uma entrada não é obrigatória.
#	Somente um caractere numérico ou sinal de mais ou menos é permitido. Uma entrada não é obrigatória.
:	Uma vírgula é usada para separar horas, minutos e segundos na definição de hora. Se um caractere diferente for especificado nas configurações regionais do Windows, será exibido em vez da vírgula.
/	Uma barra é usada para separar as entradas de data, dia, mês e ano umas das outras. Se um caractere diferente for especificado nas configurações regionais do Windows, será exibido em vez da vírgula.
;	Um ponto e vírgula é usado para separar os parâmetros das três máscaras uns dos outros.
_	O caractere sublinhado _ insere automaticamente um espaço no texto. Os cursores passam por esses caracteres quando um dado é inserido.

- Qualquer caractere não listado na tabela acima pode ser usado como o primeiro parâmetro da máscara como espaços reservados para entradas.
- Esses caracteres são inseridos automaticamente e ignorados pelo cursor na entrada de um dado.
- Os caracteres de máscara especiais listados na tabela acima também podem ser usados como espaços reservados, desde de que o caractere „\" seja especificado primeiro.
- Na sequência, use o caractere '1' para especificar que qualquer separador definido, por exemplo, barra ou espaço, seja transferido para o documento.  
O uso de '0' significa que os separadores especificados na regra de formatação não são transferidos.
- Como o terceiro parâmetro, defina o caractere para ser usado em uma caixa de entrada para exibir espaços ou dados não inseridos.

## Modelos de máscara

Clique no botão à direita do campo **Máscara de entrada**  para selecionar um modelo de máscara de entrada pré-definido.

## Variáveis manuais

Uma caixa de diálogo de entrada é exibida para variáveis cuja propriedade [Input Kind](#) permite a entrada manual durante a geração de documentos.

Se for atribuída uma origem de dados, os valores lidos da origem são exibidos como um valor padrão.

## Propriedades da variável

### Propriedades gerais de variável

[▶ Mostrar tudo](#)

### Attribute name

Nome do atributo (campo de banco de dados ou variável de pool)

### Formatting Mode

Essa propriedade permite que você defina como a formatação das variáveis em um documento deve ser feita.

A formatação é definida antes da avaliação de uma condição ser executada.

Os dois tipos de formatação a seguir estão disponíveis:

#### Divisão de string

Use o modo de formatação **Divisão de string** para preencher o conteúdo de uma variável com caracteres de preenchimento, por exemplo, zeros à esquerda para atingir o comprimento necessário.

A posição inicial e o número de casas desempenham uma função, quando somente uma substring da variável deve ser usada.

Posição inicial=3 e número de casas=2 significam que são usados 2 caracteres a partir do terceiro caractere.

Defina um caractere como o caractere de preenchimento que será usado para preencher a string até o comprimento especificado.

O valor padrão de um caractere de preenchimento é um espaço.

#### Formatação Typespecific

A formatação específica do tipo permite que você defina regras que determinam como as variáveis são formatadas quando são inseridas em um documento.

Por exemplo, o valor 123 pode ser formatado a inserido como R\$ 123,00.

Use o **Tipo de dados** para definir o tipo de variável.

É possível usar o campo **Regra** para definir a regra de formatação.

A regra de formatação contém caracteres simples e identificadores de formato. Os caracteres simples são inseridos na string do resultado no estado em que estão. Os identificadores de formato começam com "%" e atribuem um formato específico ao resultado.

#### Esta é a sintaxe de um identificador de formato:

%[largura].[prec] tipo

Após o sinal de "%", é possível fazer estas definições:

- uma definição de largura opcional [largura],
- uma definição de precisão opcional, que é denotada por um ponto à esquerda [.prec] e
- um caractere que denota o tipo de conversão (tipo).

A largura e a precisão são denotadas por números (por exemplo "%.10d"). A especificação da largura denota o comprimento mínimo do campo para conversão.

Se a string do resultado for mais curta do que o comprimento mínimo, ela será preenchida com espaços. Como regra, são inseridos espaços antes do valor para produzir alinhamento correto.

Para todos os formatos de ponto flutuante, os caracteres dos separadores decimal e de milhar são retirados das configurações DecimalSeparator e ThousandSeparator no Painel de Controle do Windows.

#### Exemplo:

para **value = 100**

a regra de formatação **%.2m** retorna

**R\$ 100,00** como resultado.

#### Tipo de conversão

Os caracteres a seguir estão disponíveis para o tipo de conversão e independem de estarem em maiúsculas ou minúsculas.

#### Caracteres de conversão

Caractere	Significado
d	<p><b>Decimal</b></p> <p>O argumento deve ser um valor inteiro. O valor é convertido em uma string composta de números decimais.</p> <p>Se a Format-String contiver uma especificação de precisão, a string resultante deverá ter, pelo menos, o número mínimo de casas decimais.</p> <p>Se tiver menos, a string será preenchida à esquerda com zeros até atingir o comprimento mínimo.</p>
e	<p><b>Notação científica</b></p> <p>O argumento deve ser um valor de ponto flutuante (float).</p> <p>O valor é convertido em uma string na forma de "-d,ddd...E+ddd". Se o número for negativo, a string começará com um sinal de menos.</p> <p>Um número sempre é definido antes do separador de decimal.</p>

.	<p>O número total de casa na string do resultado (inclusive o número antes da vírgula decimal) é determinado pelo especificador de precisão em Format-String.</p> <p>Se não houver uma especificação, será usada a precisão padrão de 15 casas.</p> <p>Um sinal de mais ou de menos e, no mínimo, três casas decimais sempre seguem o expoente "E" em uma string.</p>
f	<p><b>Fixas</b></p> <p>O argumento deve ser um valor de ponto flutuante (float).</p> <p>O valor é convertido em uma string de acordo com esta convenção: "-ddd,ddd...".</p> <p>Se o número for negativo, a string começará com um sinal de menos.</p> <p>O número de casas decimais é determinado pela especificação de precisão em Format-String.</p> <p>Se não houver uma especificação, será usada a precisão padrão de 2 casas decimais.</p>
g	<p><b>Geral</b></p> <p>O argumento deve ser um valor de ponto flutuante (float).</p> <p>O valor é convertido na string decimal mais curta, de acordo com o formato de notação científica ou fixo.</p> <p>O número de casas significantes na string de resultado é definido pelo especificador de precisão em Format-String.</p> <p>Se não houver uma especificação, será usada a precisão padrão de 15 casas.</p> <p>Os zeros à esquerda são removidos da string de resultado.</p> <p>Um ponto decimal só será exibido quando necessário.</p> <p>O formato de ponto fixo é usado para a string resultante, se o número de dígitos antes do ponto decimal for menor ou igual à precisão especificado e o valor for maior ou igual a 0,00001.</p> <p>A notação científica é usada em todos os casos.</p>
n	<p><b>Número</b></p> <p>O argumento deve ser um valor de ponto flutuante (float).</p> <p>O valor é convertido em uma string de acordo com esta convenção: "-d.ddd.ddd,ddd...".</p> <p>O formato "n" é o mesmo que o formato "f", exceto que a string de resultado contém um separador de milhar.</p>
m	<p><b>Moeda</b></p> <p>O argumento deve ser um valor de ponto flutuante (float).</p> <p>O valor é convertido em uma string que representa um valor monetário.</p> <p>A conversão está baseada nas variáveis globais do Windows para CurrencyString,</p>



	<p>CurrencyFormat, NegCurrFormat, ThousandSeparator, DecimalSeparator e CurrencyDecimals.</p> <p>Essas variáveis são inicializadas com as configurações armazenadas no Painel de Controle do Windows, na guia Configurações Regionais, Moeda.</p> <p>N.B. Se Format-String contiver um especificador de precisão, ele substituirá o valor na variável global CurrencyDecimals.</p>
s	<p><b>String</b></p> <p>O argumento deve ser um caractere, uma string ou um valor PChar.</p> <p>A string ou, conforme o caso, o caractere é inserido em vez do identificador de formato.</p> <p>Se Format-String contiver um especificador de precisão, este determinará o comprimento máximo das strings resultantes.</p> <p>Se o argumento for uma string que excede o comprimento máximo, a string será truncada.</p>
x	<p><b>Hexadecimal</b></p> <p>O argumento deve ser um valor inteiro.</p> <p>O valor é convertido em uma string composta de dígitos hexadecimais.</p> <p>Se a Format-String contiver uma especificação de precisão, a string do resultado deverá ter, pelo menos, o número mínimo de dígitos.</p> <p>Se tiver menos dígitos, a string será preenchida à esquerda com zeros até atingir o comprimento mínimo.</p>

Defina a regra de formatação no campo **Computação** para inserir uma fórmula de computação, por exemplo, \*10.

O espaço reservado para o valor na fórmula de computação é sempre **%s**, porque é passado como uma string para a computação.

### DataLink

Caminho para o objeto de dados ou o nome de um pool de variável.

### DataSource

Valor	Significado
None	<p>Se a propriedade InputType especificar a entrada manual, a variável é preenchida com esse resultado.</p> <p>Se a propriedade InputType não especificar uma entrada manual, a variável é preenchida com esse resultado.</p>
Pool	A variável é preenchida pelo valor da variável de pool atribuída.

Data Model	A variável é preenchida pelo valor do modelo de dados atribuído.
XPath	A variável é preenchida pelo valor retornado pela expressão XPath definida na propriedade <b>XPathExpression</b> .

### Input Mask

Consulte [Máscara de entrada](#)

### Input Kind

Valor	Significado
None	Sem entrada manual
Always	A entrada manual é possível
DataConditional	Se o conteúdo da variável for vazio ainda que uma fonte de dados tenha sido atribuída, manual é possível.

Consulte também [Variáveis manuais](#).

### Prompt

Texto do prompt para variáveis manuais.

### Variable Type

Valor	Significado
Free	Variável livre Se essa variável não estiver preenchida, ela não possui relevância na geração do documento.
Required	A variável é obrigatória Se essa variável não estiver preenchida, a geração do documento é abortada. As variáveis obrigatórias com entrada manual são exibidas em vermelhos no diálogo de entrada.
Optional	A variável é opcional Se essa variável não estiver preenchida, uma entrada de log é feita para esse efeito.

### Default Value

Valor aplicado, se a variável não estiver preenchida.

### Valores padrão para um usuário ModusOne

Para preencher uma variável com os valores relativos ao usuário atual, use um valor padrão de acordo com a sintaxe a seguir:

```
[ModusUser.AttributeName]
```

Usando o valor [ModusUser.UserId], por exemplo, você pode tornar o valor padrão, para um campo de entrada, o nome do usuário conectado no momento.

Você pode encontrar uma visão geral dos atributos já disponíveis no diálogo de opções, atingidos por meio do **botão ModusOne - Opções - Dados do usuário**.

A propriedade **InputType** deve ser definida para o valor **Always**.

### Value Help

Consulte [Ajudas de valor](#)

### ValueHelpOverwritable

Essa propriedade é avaliada quando uma ajuda de valor é atribuída.

Valor	Significado
True	Um valor pode ser inserido manualmente ao invés de selecionar a partir da ajuda de valor.
False	Apenas um valor da ajuda de valor pode ser selecionado.

### XPathExpression

Clique no botão ao lado da propriedade e selecione um arquivo XML.

A janela Avaliador de XPath é aberta, permitindo que você insira o início e a expressão XPath e verifique o resultado.

#### Conclua as etapas a seguir para atribuir uma expressão XPath:

1. No Inspetor de objetos, clique na propriedade **Parâmetro inicial**.
2. Em seguida, clique no botão com os três pontos ao lado da propriedade.
3. Selecione um arquivo XML na caixa de diálogo seguinte.
4. Selecione a variável na estrutura XML que deve ser atribuída à variável de bloco de texto.
5. Selecione a função **Copiar XPath** no menu de contexto.
6. Clique no campo **Expressão de XPath**.
7. Selecione a função **Inserir** no menu de contexto.
8. Clique no botão **Avaliar**.
9. Verifique o resultado no campo **Resultado de XPath**.
10. Clique no botão **Aplicar**.

### Exemplos:

Adição de valores com XPath

A função Sum retorna a soma de todos os valores de um nó.

**Observe:**

O caractere de vírgula não é reconhecido, isto é, um ponto (ponto decimal) deve ser usado como o separador de decimal no arquivo XML (ortografia dos EUA).

**Sintaxe:**

Sum(PARTNER/ACCOUNTS/MOV\_assets)  
 sum(//ACCOUNT[1]/MOVEMENTS/MOV\_assets)

//	Somente procura o nó ACCOUNT independentemente do nó pai, isto é, procura o nó onde quer que seja encontrado. Não é necessário definir o caminho do nó.
ACCOUNT[1]	Somente o primeiro nó com o nome ACCOUNT é lido.
/MOVEMENTS	Nó filho
/MOV_assets	Elemento (com valores)

Existem várias instâncias do nó ACCOUNT, mas somente a soma dos valores de um elemento de nó para um nó ACCOUNT será calculado.

O nó ACCOUNT é executado apenas uma vez, mas todos os nós filhos (MOVEMENTS) são avaliados.

Os valores são lidos por um bloco de texto múltiplo (relacionamento 1:n) no modus.

Dessa forma, todos os nós têm a mesma estrutura e apenas valores diferentes.

O valor de retorno é um valor decimal normal (por exemplo, R\$ 100,33) e, portanto, do tipo de dados **Duplo**.

Outras funções úteis

Concat()	Permite combinar strings de acordo com regras definidas individualmente. Assim, o conteúdo lido de elementos pode ser combinado com caracteres à esquerda. <b>Exemplo:</b> concat ( Postbox , PARTNER/ADDRESS/ADD_POB ) <b>Valor de retorno:</b> Caixa postal 08 15
Count()	Conta o número de nós especificados encontrados no arquivo XML. Isso é útil para relacionamentos 1:n, nos quais são encontrados vários nós com o mesmo nome (blocos de texto múltiplos). <b>Exemplo:</b> Count ( PARTNER/ACCOUNTS )
Translat	Substitui todos os caracteres minúsculos por letras maiúsculas no conteúdo lido do nome

e()	do elemento. <b>Exemplo:</b> <code>Translate(//ADD_country_name,abcdefghijklmnopqrstuvwxyz,ABCDEFGHIJKLMN OPQRSTUVWXYZ)</code>
Caracter e: //	Lê o conteúdo de todos os elementos do nome especificado, independentemente de onde o elemento está no arquivo XML.  Não é necessário definir um caminho particular, pois somente o elemento é procurado. <b>Exemplo:</b> <code>//MOV_liabilities</code>

### Condição da variável

A condição deve retornar o resultado **true** para a variável a ser inserida no bloco de texto.

Você pode atribuir uma condição clicando no botão ao lado da propriedade ou arrastando o objeto de condição do navegador e soltando-o sobre a variável.

Consulte também:

[Atribuição de scripts e condições](#)

### Eventos da variável

#### AfterInsert

O script definido aqui é executado depois que a variável é inserida no bloco de texto.

Contexto: [TextVariable](#)

#### BeforeInsert

O script definido aqui é executado antes da variável ser inserida no bloco de texto.

Contexto: [TextVariable](#)

### Pesquisar

#### LookupFormattingType

Formatação de saída após avaliação de condição e tabela de pesquisa.

Os dois tipos de formatação a seguir estão disponíveis:

#### Divisão de string

Use o modo de formatação **Divisão de string** para preencher o conteúdo de uma variável com caracteres de preenchimento, por exemplo, zeros à esquerda para atingir o comprimento necessário.

A posição inicial e o número de casas desempenham uma função, quando somente uma substring da variável deve ser usada.

Posição inicial=3 e número de casas=2 significam que são usados 2 caracteres a partir do terceiro caractere.

Defina um caractere como o caractere de preenchimento que será usado para preencher a string até o comprimento especificado.

O valor padrão de um caractere de preenchimento é um espaço.

### Formatação específica do tipo

A formatação específica do tipo permite que você defina regras que determinam como as variáveis são formatadas quando são inseridas em um documento.

Por exemplo, o valor 123 pode ser formatado a inserido como R\$ 123,00.

Use o **Tipo de dados** para definir o tipo de variável.

É possível usar o campo **Regra** para definir a regra de formatação.

A regra de formatação contém caracteres simples e identificadores de formato. Os caracteres simples são inseridos na string do resultado no estado em que estão. Os identificadores de formato começam com "%" e atribuem um formato específico ao resultado.

### Esta é a sintaxe de um identificador de formato:

%[largura].[prec] tipo

Após o sinal de "%", é possível fazer estas definições:

- uma definição de largura opcional [largura],
- uma definição de precisão opcional, que é denotada por um ponto à esquerda [.prec] e
- um caractere que denota o tipo de conversão (tipo).

A largura e a precisão são denotadas por números (por exemplo "%.10d). A especificação da largura denota o comprimento mínimo do campo para conversão.

Se a string do resultado for mais curta do que o comprimento mínimo, ela será preenchida com espaços. Como regra, são inseridos espaços antes do valor para produzir alinhamento correto.

Para todos os formatos de ponto flutuante, os caracteres dos separadores decimal e de milhar são retirados das configurações DecimalSeparator e ThousandSeparator no Painel de Controle do Windows.

### Exemplo:

para **value = 100**

a regra de formatação **%.2m** retorna

**R\$ 100,00** como resultado.

### Tipo de conversão

Os caracteres a seguir estão disponíveis para o tipo de conversão e independem de estarem em maiúsculas ou minúsculas.

### Caracteres de conversão

Caractere	Significado
d	Decimal

	<p>O argumento deve ser um valor inteiro. O valor é convertido em uma string composta de números decimais.</p> <p>Se a Format-String contiver uma especificação de precisão, a string resultante deverá ter, pelo menos, o número mínimo de casas decimais.</p> <p>Se tiver menos, a string será preenchida à esquerda com zeros até atingir o comprimento mínimo.</p>
e	<p><b>Notação científica</b></p> <p>O argumento deve ser um valor de ponto flutuante (float).</p> <p>O valor é convertido em uma string na forma de "-d,ddd...E+ddd". Se o número for negativo, a string começará com um sinal de menos.</p> <p>Um número sempre é definido antes do separador de decimal.</p>
.	<p>O número total de casa na string do resultado (inclusive o número antes da vírgula decimal) é determinado pelo especificador de precisão em Format-String.</p> <p>Se não houver uma especificação, será usada a precisão padrão de 15 casas.</p> <p>Um sinal de mais ou de menos e, no mínimo, três casas decimais sempre seguem o expoente "E" em uma string.</p>
f	<p><b>Fixas</b></p> <p>O argumento deve ser um valor de ponto flutuante (float).</p> <p>O valor é convertido em uma string de acordo com esta convenção: "-ddd,ddd...".</p> <p>Se o número for negativo, a string começará com um sinal de menos.</p> <p>O número de casas decimais é determinado pela especificação de precisão em Format-String.</p> <p>Se não houver uma especificação, será usada a precisão padrão de 2 casas decimais.</p>
g	<p><b>Geral</b></p> <p>O argumento deve ser um valor de ponto flutuante (float).</p> <p>O valor é convertido na string decimal mais curta, de acordo com o formato de notação científica ou fixo.</p> <p>O número de casas significantes na string de resultado é definido pelo especificador de precisão em Format-String.</p> <p>Se não houver uma especificação, será usada a precisão padrão de 15 casas.</p> <p>Os zeros à esquerda são removidos da string de resultado.</p> <p>Um ponto decimal só será exibido quando necessário.</p> <p>O formato de ponto fixo é usado para a string resultante, se o número de dígitos antes do ponto decimal for menor ou igual à precisão especificado e o valor for maior ou igual a 0,00001.</p> <p>A notação científica é usada em todos os casos.</p>

n	<p><b>Número</b></p> <p>O argumento deve ser um valor de ponto flutuante (float).</p> <p>O valor é convertido em uma string de acordo com esta convenção: "-d.ddd.ddd,ddd...".</p> <p>O formato "n" é o mesmo que o formato "f", exceto que a string de resultado contém um separador de milhar.</p>
m	<p><b>Moeda</b></p> <p>O argumento deve ser um valor de ponto flutuante (float).</p> <p>O valor é convertido em uma string que representa um valor monetário.</p> <p>A conversão está baseada nas variáveis globais do Windows para CurrencyString, CurrencyFormat, NegCurrFormat, ThousandSeparator, DecimalSeparator e CurrencyDecimals.</p> <p>Essas variáveis são inicializadas com as configurações armazenadas no Painel de Controle do Windows, na guia Configurações Regionais, Moeda.</p> <p>N.B. Se Format-String contiver um especificador de precisão, ele substituirá o valor na variável global CurrencyDecimals.</p>
s	<p><b>String</b></p> <p>O argumento deve ser um caractere, uma string ou um valor PChar.</p> <p>A string ou, conforme o caso, o caractere é inserido em vez do identificador de formato.</p> <p>Se Format-String contiver um especificador de precisão, este determinará o comprimento máximo das strings resultantes.</p> <p>Se o argumento for uma string que excede o comprimento máximo, a string será truncada.</p>
x	<p><b>Hexadecimal</b></p> <p>O argumento deve ser um valor inteiro.</p> <p>O valor é convertido em uma string composta de dígitos hexadecimais.</p> <p>Se a Format-String contiver uma especificação de precisão, a string do resultado deverá ter, pelo menos, o número mínimo de dígitos.</p> <p>Se tiver menos dígitos, a string será preenchida à esquerda com zeros até atingir o comprimento mínimo.</p>

Defina a regra de formatação no campo **Computação** para inserir uma fórmula de computação, por exemplo, \*10.

O espaço reservado para o valor na fórmula de computação é sempre **%s**, porque é passado como uma string para a computação.

### LookupSource



Objeto a ser pesquisado.

### LookupDefault

Valor padrão quando nenhum valor é encontrado na tabela de pesquisa

### LookupKey

Valor de chave a ser passado. A inserção de %s passas o conteúdo da variável.

### LookupValueCondition

Condição avaliada após pesquisa

Se o resultado da condição for **False**, a variável não será inserida.

## O controle de conteúdo EmptyLine


Use o controle de conteúdo **EmptyLine** para definir uma área em um bloco de texto ou documento que será excluída durante a geração de documentos se as variáveis contida na seção não forem populadas com valores após a recuperação de dados.

Você pode aplicar essa prática a uma única linha, várias linhas ou apenas partes de uma linha.

Se o controle de conteúdo contiver um bloco de texto ativo, todo o conteúdo do controle de conteúdo é visível.

Se o bloco de texto contido no controle de conteúdo for desativado e nenhuma das variáveis tiver valores, todo o controle de conteúdo é removido do documento durante a geração.

### Execute as seguintes etapas para inserir um controle de conteúdo EmptyLine:

1. Posicione o cursor no Word no local onde o controle de conteúdo será inserido.
2. Vá para o item do ribbon Ferramentas do desenvolvedor - Elementos de controle e insira um elemento de rich text a partir do menu: 

#### Observação:

o item de menu Desenvolvedor pode ser ativado no menu principal do Word:

**Opções do Word - Frequentemente usados - Mostrar guia Desenvolver no Menu multifunção.**

3. Edite as propriedades do controle de conteúdo no item de menu **Ferramentas do desenvolvedor - Elementos de controle - Propriedades:**
  - o Insira um título.
  - o No campo Tag:, insira **mc:EmptyLine**.
  - o Feche a janela com o **botão OK**.
4. Arraste e solte uma ou mais variáveis para o controle de conteúdo.

### Método alternativo

1. Selecione um segmento de texto que deva ser incluído no controle de conteúdo (inclusive a marca de parágrafo final)
2. Crie o controle de conteúdo como descrito acima.

## Atribuir um modelo de dados

Atribua uma seleção ao bloco de texto para preencher as variáveis do bloco de texto com valores de uma tabela de banco de dados ou arquivo XML.

### Siga estas etapas:

1. Selecione a condição obrigatória no Navigator.
2. Arraste e solte a seleção do Navigator para a área à direita da janela do bloco de texto. A seleção agora é exibida na área superior direita da janela do bloco de texto.
3. Clique em um dos objetos de dados na seleção. As variáveis do objeto de dados agora são exibidas na janela inferior direita.
4. Atribua as variáveis do objeto de dados às variáveis do bloco de texto arrastando e soltando.

## Inserir um bloco de texto filho

Blocos de texto podem ser aninhados uns dentro dos outros em qualquer profundidade.

### Conclua as seguintes etapas para inserir um bloco de texto filho:

1. Abra o bloco de texto em que o bloco de texto filho deve ser inserido.
2. Selecione o bloco de texto filho obrigatório dos blocos de texto no Navigator.
3. Arraste o bloco de texto obrigatório do Navigator e solte-o na janela do editor. Se necessário, o mesmo bloco de texto pode ser inserido (aninhado) mais de uma vez.
4. Ative o bloco de texto filho usando o menu de contexto ou alterne a propriedade `TextInsertionRule`.

### Funções no menu de contexto dos blocos de texto filhos

#### Redefinir Data Link

Exclui o valor da propriedade `DataLink`.

#### Atributos de herança

Se essa função é ativada, o bloco de texto herda no documento as propriedades do bloco de texto original.

Consulte também:

[Herança de atributos de bloco de texto](#)

#### Obrigatório

Altera o status do bloco de texto. O valor da propriedade `TextInsertionRule` é definido como **Required**.

#### Ativar

Altera o status do bloco de texto. O valor da propriedade `TextInsertionRule` é definido como **Activated**.

#### Desativado

Altera o status do bloco de texto. O valor da propriedade `TextInsertionRule` é definido como **Deactivated**.

#### Desabilitado

Altera o status do bloco de texto. O valor da propriedade [TextInsertionRule](#) é definido como **Disabled**.

#### Selecionar no Word

Destaca as referências no bloco de texto selecionado no documento do Word.

#### Abrir bloco de texto

Abre o bloco de texto selecionado no modo de edição.


#### Remover

Remove o bloco de texto selecionado.

## Aplicar propriedade de bloco de texto e variável

Se você alterou as propriedades de um bloco de texto ou variável que já está em uso por um documento e agora deseja transferir essas alterações para todos os objetos, vá para as **Ferramentas do Modus Editor** e selecione o item do menu **Aplicar propriedades**.

**Para aplicar propriedades de objeto, execute estas etapas:**

1. Abra o bloco de texto cujas propriedades serão acumuladas para referências de bloco de texto.
2. Selecione o bloco de texto ou variável na janela do bloco de texto.
3. Vá para a guia **Ferramentas do Modus Editor** e clique no botão **Aplicar propriedades** 
4. Selecione as propriedades a serem acumuladas na janela de diálogo seguinte.
5. Clique no botão **Aplicar** para iniciar a transferência.

## Vários blocos de texto

Vários blocos de texto são usados para moldar 1:n relacionamentos em estruturas de dados nos blocos de texto.

Isso permite, por exemplo, inserir vários detalhes de contas de banco diferentes em um documento.

**Execute as etapas a seguir para definir vários blocos de texto:**

1. Arraste o 1:n nó desejado da estrutura de dados até o bloco de texto.
2. A propriedade **Multiple** altera para **True**.
3. O bloco de texto é inserido para cada conta achada na estrutura de dados.

#### Observe:

- Um bloco de texto variável não pode ser usado como vários blocos de texto. Mesmo com a propriedade **Multiple** definida como **True** o bloco de texto será apenas inserido uma vez na estrutura do documento.
- O XPath não é adequado para o usar com vários blocos de texto.

**Ao executar uma expressão XPath, o ModusOne apresenta o seguinte comportamento:**

Se houver vários nós no XML de dados que correspondem à expressão XPath especificada - como é geralmente o caso com vários blocos de texto - o valor do primeiro nó encontrado é sempre usado como o valor da variável, esteja ele na primeira ou enésima iteração.

**Solução:**

Para variáveis de vários blocos de texto, use a opção **DataModel** para a origem de dados e especifique o link de dados correspondente e nome do atributo.

**Caso especial: Várias inserções de um bloco de texto usando um script**

Um bloco de texto pode ser inserido como você sempre desejou usando um script. Isso é completamente independente do número de conjuntos de dados lidos na execução.

**Execute as seguintes etapas para fazer isso:**

1. Defina a propriedade **Multiple** do bloco de texto como **True**.
2. Defina a propriedade **DataSource** do bloco de texto como **None**.
3. Atribua um script ao Evento **BeforeInsert** ou **AfterInsert** do bloco de texto.
4. Nesse script, defina a variável **Blockpool.Break** como **True**, para sinalizar o fim da inserção de iterações.

**Observe:**

Se a variável **Blockpool.Break** for definida como **False** em um script, ela deve ser definida como **True** posteriormente.

Se não for, o script será executado em um loop infinito.

## Pools de variável

Os pools de variável podem ser usados, por exemplo, para armazenar os resultados de cálculos.

Eles também podem ser usados para passar variáveis para sistemas conectados como o Odin.

- O pool **MODUSER** fornece informações específicas de usuário.
- O contextpool **BLOCKPOOL** fornece acesso a variáveis de bloco de texto.
- O **DOCUMENTPOOL** fornece acesso a variáveis de documento.
- O **SYSTEMPOOL** está disponível em pacotes.

A inserção de **Poolname.NewVariablename** em um script cria uma nova variável.

## Objetos de variável

Com o uso de scripts e condições, você pode controlar se um bloco de texto é inserido em um documento ou se um documento é inserido em um pacote.

Em alguns casos, como termos gerais e condições relacionadas a apólices de seguros, o objeto a ser inserido pode ser vinculado diretamente a um valor nos dados (por exemplo, valor = número do documento interno).

Dessa forma, nem todos os objetos (que podem ser solicitados) precisam ser predefinidos na estrutura do documento e em seguida ativados ou desativados por meio de condições. É possível definir um objeto de espaço reservado que, dependendo do conteúdo dos dados durante a geração do documento, pode ser substituído pelo objeto solicitado.

**Execute as seguintes etapas para definir um objeto como um objeto de variável:**

1. Selecione a propriedade **VariableObjectDefinition** no item **Criar** do menu de contexto. Opcionalmente, a **VariableObjectDefinition** pode ser criada arrastando e soltando uma tabela de

aliasas sobre o objeto.

Como resultado, as seguintes propriedades adicionais serão visíveis abaixo da definição de objeto usada para determinar o objeto a ser inserido.

- `AttributeName`
- `DataLink`
- `DataSource`
- `LookupSource`
- `LookupKey`
- `XPathExpression`

As descrições dessas propriedades podem ser encontradas em Propriedades de variável.

2. As propriedades **AttributeName** e **DataLink** ou **XPathExpression** são usadas para definir o campo de dados que contem o nome do objeto a ser inserido.
3. Além disso, você pode usar a propriedade **LookupSource** para definir uma tabela de aliasas a ser usada para procurar o valor recuperado.

#### Observe:

- As variáveis manuais não devem ser usadas em blocos de texto de variável, pois a caixa de diálogo de entrada para entradas manuais não é exibida quando o objeto de variável é inserido no documento.
- Você pode usar a propriedade de pacote **MissedObjects**, cujo conteúdo é apagado antes de cada geração de documento, para detectar os objetos de variável não encontrados.
- Em geral, os scripts do objeto de espaço reservado de variável são executados para os eventos **BeforeInsert** e **AfterInsert**.
- Se não houver scripts atribuídos ao objeto de espaço reservado para esses eventos, os scripts (se atribuídos) do objeto inserido é executado.

## Herança de atributos de bloco de texto

Modificações em um bloco de texto não têm efeito nas propriedades do mesmo bloco de texto quando ele é usado como um bloco de texto filho em um documento ou um bloco principal.

Portanto, um bloco de texto pode ter propriedades diferentes para cada uso diferente.

Somente a estrutura do bloco de texto é reutilizada (ou seja, as variáveis de bloco de texto e blocos de texto aninhados).

Os blocos de texto que herdam suas propriedades são a exceção à essa regra.

#### Conclua as seguintes etapas para ativar a herança:

1. Abra o documento ou o bloco de texto que contém o bloco de texto filho.
2. Selecione o bloco de texto que deve herdar as propriedades.
3. Ative o item de menu de contexto **Herdar atributos**.  
A cor do símbolo do bloco de texto altera para vermelho.

As propriedades do bloco de texto não pode mais ser alterado. As propriedades do bloco de texto original são sempre válidas.

## Transferir propriedades

A transferência de propriedades entre blocos de texto e variáveis é um procedimento simples.

### Execute as seguintes etapas para fazer isso:

1. Pressione a tecla CTRL e clique no objeto cujas propriedades serão transferidas.
2. Arraste o objeto de origem para o objeto de destino. O símbolo do cursor muda para um pequeno sinal de adição.

Os possíveis objetos de destino são qualquer objeto do mesmo tipo em qualquer nível. Você pode transferir entre objetos dentro da janela de objetos atual ou entre qualquer outra janela de objetos aberta.

Novamente, os objetos devem ser do mesmo tipo.

Ao transferir entre blocos de texto ou variável com IDs de objeto diferentes, a ação é concluída sem qualquer ação adicional.

Se os blocos de texto tiverem a mesma ID de objeto, uma nova janela será exibida com três opções:

- Somente propriedades de bloco de texto
- Somente propriedades das variáveis de bloco de texto
- Propriedades do bloco de texto e suas variáveis

3. Selecione a opção desejada e clique no botão **Aplicar**.

## Propriedades de bloco de texto

### Geral

#### AttributeInheritance

Essa propriedade pode ser alterada somente pelo menu de contexto de um bloco de texto durante a edição de documentos.

#### Consulte também:

[Herança de atributos de bloco de texto](#)

#### Expansível

Define se o usuário pode ver a estrutura do bloco de texto e tem permissão para ativar ou desativar seus objetos.

#### DataLink

Caminho para o objeto de dados.

#### DataSource

Valor	Significado
None	Nenhum dado atribuído.

Pool	Dados são recuperados da variável pool atribuída.
Data Model	A variável é preenchida pelo valor do modelo de dados atribuído. Vincular um objeto de dados também controla o número de vezes que o objeto é inserido. Se nenhum dados for encontrado no endereço especificado, o bloco de texto não é inserido.
XPath	Os dados são recuperados da expressão definida na propriedade <b>XPathExpression</b> . Isso retorna um número de nós que determina a frequência de inserção do bloco de texto.

### Multiple

É definido automaticamente, dependendo do objeto de dados atribuído.

Se a relação for 1:1, a propriedade é False. Para uma relação 1:N, a propriedade é True.

### Consulte também:

[Diversos blocos de texto](#)

### Visible

Define se o bloco de texto fica visível aos usuários finais.

### TextComponentProtection

Se a propriedade for True, os usuários não podem fazer quaisquer alterações ao bloco de texto.

### TextInsertionRule

Valor	Significado
Activated	Se o bloco de texto não puder ser inserido durante a geração de documento, a anulação não é forçada.
Deactivated	O bloco de texto é desativado e não pode ser ativado por um usuário.
Disabled	O bloco de texto não será inserido no documento.
Required	A geração do documento é abortada se o bloco de texto não puder ser inserido.

### Title

O valor definido aqui é o nome do bloco de texto que o usuário vê no cliente.

### Default inheritance value

Valor	Significado
True	O bloco de texto é inserido no documento ou outro bloco de texto com a propriedade <b>AttributeInheritance=True</b> As propriedades do bloco de texto inserido sempre correspondem às propriedades do bloco de texto original e não podem ser alteradas.
False	O bloco de texto é inserido no documento ou outro bloco de texto com a propriedade <b>AttributeInheritance=False</b> As propriedades do bloco de texto inserido podem ser alteradas.

**Consulte também:**

[Herança de atributos de bloco de texto](#)

**XPathExpression**

Clique no botão ao lado da propriedade e selecione um arquivo XML.

A janela Avaliador de XPath é aberta, permitindo que você insira o início e a expressão XPath e verifique o resultado.

**Conclua as etapas a seguir para atribuir uma expressão XPath:**

1. No Inspetor de objetos, clique na propriedade **Parâmetro inicial**.
2. Em seguida, clique no botão com os três pontos ao lado da propriedade.
3. Selecione um arquivo XML na caixa de diálogo seguinte.
4. Selecione a variável na estrutura XML que deve ser atribuída à variável de bloco de texto.
5. Selecione a função **Copiar XPath** no menu de contexto.
6. Clique no campo **Expressão de XPath**.
7. Selecione a função **Inserir** no menu de contexto.
8. Clique no botão **Avaliar**.
9. Verifique o resultado no campo **Resultado de XPath**.
10. Clique no botão **Aplicar**.

**Exemplos:**

Adição de valores com XPath

A função Sum retorna a soma de todos os valores de um nó.

**Observe:**

O caractere de vírgula não é reconhecido, isto é, um ponto (ponto decimal) deve ser usado como o separador de decimal no arquivo XML (ortografia dos EUA).

**Sintaxe:**

```
Sum(PARTNER/ACCOUNTS/MOV_assets)
sum(//ACCOUNT[1]/MOVEMENTS/MOV_assets)
```



//	Somente procura o nó ACCOUNT independentemente do nó pai, isto é, procura o nó onde quer que seja encontrado. Não é necessário definir o caminho do nó.
ACCOUNT[1]	Somente o primeiro nó com o nome ACCOUNT é lido.
/MOVEMENTS	Nó filho
/MOV_assets	Elemento (com valores)

Existem várias instâncias do nó ACCOUNT, mas somente a soma dos valores de um elemento de nó para um nó ACCOUNT será calculado.

O nó ACCOUNT é executado apenas uma vez, mas todos os nós filhos (MOVEMENTS) são avaliados.

Os valores são lidos por um bloco de texto múltiplo (relacionamento 1:n) no modus.

Dessa forma, todos os nós têm a mesma estrutura e apenas valores diferentes.

O valor de retorno é um valor decimal normal (por exemplo, R\$ 100,33) e, portanto, do tipo de dados **Duplo**.

Outras funções úteis

Concat()	Permite combinar strings de acordo com regras definidas individualmente. Assim, o conteúdo lido de elementos pode ser combinado com caracteres à esquerda. <b>Exemplo:</b> <code>concat (Postbox , PARTNER/ADDRESS/ADD_POB)</code> <b>Valor de retorno:</b> Caixa postal 08 15
Count()	Conta o número de nós especificados encontrados no arquivo XML. Isso é útil para relacionamentos 1:n, nos quais são encontrados vários nós com o mesmo nome (blocos de texto múltiplos). <b>Exemplo:</b> <code>Count ( PARTNER/ACCOUNTS )</code>
Translate()	Substitui todos os caracteres minúsculos por letras maiúsculas no conteúdo lido do nome do elemento. <b>Exemplo:</b> <code>Translate ( //ADD_country_name , abcdefghijklmnopqrstuvwxyz , ABCDEFGHIJKLM NOPQRSTUVWXYZ )</code>
Caractere://	Lê o conteúdo de todos os elementos do nome especificado, independentemente de onde o elemento está no arquivo XML. Não é necessário definir um caminho particular, pois somente o elemento é procurado.

**Exemplo:**

```
//MOV_liabilities
```

## Condições

### Condições

A condição deve retornar o resultado 'true' para o bloco de texto a ser inserido no documento.

Você pode atribuir a condição clicando no botão ao lado da propriedade, ou arrastando um objeto de condição do navigator e soltando-o em um bloco de texto.

### Consulte também:

[Atribuição de scripts e condições](#)

## Modelo de dados

### Selection

Essa propriedade armazena a seleção associada.

## Eventos de bloco de texto

### AfterInsert

O script definido aqui é executado depois que o bloco de texto é inserido no documento.

Contexto: [TextComponent](#)

### BeforeInsert

O script definido aqui é executado antes que o bloco de texto seja inserido no documento.

Contexto: [TextComponent](#)

## Definições de objetos

### VariableObjectDefinition

Essa propriedade contém as configurações para blocos de texto variáveis.

Se uma tabela de alias for solta em um bloco de texto, ou a VariableObjectDefinition for criada de um menu de contexto, propriedades adicionais são exibidas.

Estas propriedades determinam qual bloco de texto será realmente inserido.

- Nome do atributo
- DataLink
- DataSource

- LookupSource
- LookupKey
- XPathExpression

**Consulte também:**

[Objetos de variável](#)

# Documentos

## Introdução

Como uma regra, documentos dinâmicos ou modelos são compostos de um ou mais blocos de texto. Esses blocos de texto também podem ser aninhados um dentro do outro.

Documentos estáticos são documentos que, quando criados como um objeto, foram importados como documentos concluídos no ModusOne e não podem ser editados.

Os documentos estáticos se referem aos documentos do Word, PDF, TIFF e XPS

## Documentos dinâmicos

### Criar um documento dinâmico

Documentos dinâmicos são compostos por blocos de texto.

**Para criar um documento dinâmico, execute estas etapas:**

1. Selecione **Novo -> Documento** do menu de contexto do Navigator.
2. Digite o nome do documento na caixa de diálogo seguinte.
3. Em seguida, selecione o tipo de documento **Documento do Word dinâmico**.  
O novo documento é aberto na janela do editor.

**Agora é possível:**

- [inserir os blocos de texto obrigatórios](#)
- [atribuir um modelo de dados](#)
- gravar texto adicional no documento do Word
- executar qualquer formatação necessária
- [definir as propriedades](#) do documento
- [criar variáveis globais](#)
- [definir grupos e ações](#)
- [usar a visualização de documento](#)

## Menu de contexto da janela do editor

### Menu de contexto de um documento

#### Obrigatório

Altera o status do documento. O valor da propriedade [TextInsertionRule](#) é definido como **required**.

#### Ativar

Altera o status do bloco de texto. O valor da propriedade [TextInsertionRule](#) é definido como **activated**.

#### Desativado

Altera o status do bloco de texto. O valor da propriedade [TextInsertionRule](#) é definido como **deactivated**.

#### **Desabilitado**

Altera o status do bloco de texto. O valor da propriedade [TextInsertionRule](#) é definido como **disabled**.

#### **Menu de contexto de blocos de texto**

##### **Remove DataLink**

Exclui o valor da propriedade DataLink.

##### **Herdar atributos**

Se essa função estiver ativada, o bloco de texto no documento herda as propriedades do bloco de texto original.

Consulte também [Herdar atributos do bloco de texto](#)

##### **Obrigatório/Ativar/ Desativado/Desabilitado**

Altera o status do bloco de texto (propriedade TextInsertionRule)

##### **Selecionar no Word**

Realça a referência ao bloco de texto selecionado no documento do Word.

##### **Abrir bloco de texto**

Abre o bloco de texto selecionado no modo de edição.

##### **Remover**

Remove o bloco de texto selecionado.

#### **Menu de contexto das variáveis**

##### **Redefinir DataLink**

Exclui o valor da propriedade DataLink.

##### **Redefinir nome do atributo**

Exclui o valor da propriedade AttributeName.

##### **Obrigatório**

Altera o status da variável. O valor da propriedade Variable Type é definido como **required**.

##### **Opcional**

Altera o status da variável. O valor da propriedade Variable Type é definido como **optional**.

##### **Livre**

Altera o status da variável. O valor da propriedade Variable Type é definido como **free**.

#### **Funções da ferramenta do editor**

**Atualizar** 

Atualiza o documento se os blocos de texto filhos ou seleções foram modificados com o documento ainda aberto em modo de edição.

Se as modificações foram feitas ao documento, a função se mantém desativada. A atualização é feita no salvamento.

#### Abrir bloco de texto

Abre o bloco de texto selecionado.

#### Definir novo grupo

Consulte [Grupos](#)

#### Mostrar documento do Word

Traz o documento do Word do documento para o primeiro plano.

#### Alternar visualização

Divide a tela para mostrar as janelas do ModusOne e Word.

#### Compilar

Compila o documento.

#### ModusOne Client

Testa o documento usando o ModusOne Client integrado.

## Documentos estáticos

### Criar um documento estático

Documentos estáticos são documentos que não podem ser alterados e devem ser carregados de uma fonte externa.

#### Para criar um documento estático, execute estas etapas:

1. Selecione **Novo -> Documento** do menu de contexto do Navigator.
2. Digite o nome do documento na caixa de diálogo seguinte.
3. Depois, selecione os tipos de documentos estáticos.

#### Para documentos estáticos, agora você pode:

- Usar as ferramentas de edito para carregar o documento necessário de uma unidade local ou unidade de rede e salvar um documento já carregado em um arquivo.
- Criar [variáveis globais](#)

### Carregar documentos de fontes externas

Se, primeiramente, você deseja carregar o documento estático de uma fonte externa durante a geração de pacote, você não precisa salvar o documento nesse momento.

Em vez disso, atribua um script ao evento **Load**, que contém um comando **LoadExternalDocument** para carregar o documento necessário.

#### Consulte também:

[Eventos do documento](#)

### Documentos TIFF estáticos

É possível definir configurações específicas relacionadas a largura de página, altura de página, posições X e Y e dimensionamento de documentos TIFF estáticos no nível da propriedade de pacote.

Isso é feito usando o método [SetArchiveAttribute](#) para definir a variável de arquivo **tifflayout**.

A variável é passada para a coluna **ODJ\_TIFFLAYOUT** na tabela **ODIN\_JOB** para processamento posterior no Odin.

O componente Odin-Rendition usa as configurações ao converter para XPS.

#### Sintaxe

```
SetArchiveAttribute("tifflayout", "Page Width/Page Height/X-Position/Y-Position/Scaling Factor")
```

#### Parâmetro

Parâmetro	Significado
Page Width	Largura em pixels
Page Height	Altura em pixels
X-Position	Posição X do TIFF
Y-Position	Posição Y do TIFF
Scaling Factor	Especificação opcional do fator de dimensionamento definido em porcentagem. Se não for feita qualquer especificação, o fator padrão de 100 é usado.

#### Exemplo

```
SetArchiveAttribute("tifflayout", "794.0/1123.0/0.0/0.0/90")
```

### Funções da ferramenta do editor



Carregar documento

Abre uma caixa de diálogo para a seleção de um documento finalizado.




### Salvar documento em arquivo

Permite que você salve um documento já carregado como arquivo.

## Criar o controle de conteúdo 'CopyLabel'

O texto especificado na propriedade **CopyLabel** de um modelo de documento é o texto inserido no controle de conteúdo **CopyLabel** quando uma cópia do documento é gerada.

**Para inserir um controle de conteúdo CopyLabel, execute estas etapas:**

1. Posicione o cursor no Word no local onde o controle de conteúdo será inserido.
2. Vá para o item de ribbon **Ferramentas do desenvolvedor - Elementos de controle** e insira um elemento rich text do menu .

#### Observação:

o item de menu Desenvolvedor pode ser ativado no menu principal do Word:

**Opções do Word - Usadas com frequência - Mostrar guia de desenvolvimento no menu multifunção.**

3. Edite as propriedades do controle de conteúdo pelo item do menu **Ferramentas do desenvolvedor - Elementos de controle - Propriedades:**
  - Insira um título.
  - No campo **Tag:**, insira **mp:CopyLabel**
  - Feche a janela com o botão OK.
4. Posicione o controle no documento conforme necessário.

## Inserir um bloco de texto

**Conclua as seguintes etapas para inserir um bloco de texto em um documento:**

1. Arraste o bloco de texto obrigatório do Navigator e solte-o até a janela do editor do documento. O bloco de texto é inserido na parte inferior da estrutura.
2. É possível alterar a ordem dos blocos de texto arrastando e soltando. Ou, como alternativa, é possível alterar a ordem dos blocos de texto no Word.

### Inserir imagens

As imagens devem ser sempre inseridas como um link para os documentos ou blocos de texto.

Certifique-se de que o arquivo de imagem tem o tamanho desejado antes de inseri-lo no documento.

## Atribuir um modelo de dados

Arraste a seleção necessária do Navigator e solte-a na janela do editor do documento.

Se o documento já tiver uma seleção definida, essa operação sobrescreve o link.



## Objetos de variável

Com o uso de scripts e condições, você pode controlar se um bloco de texto é inserido em um documento ou se um documento é inserido em um pacote.

Em alguns casos, como termos gerais e condições relacionadas a apólices de seguros, o objeto a ser inserido pode ser vinculado diretamente a um valor nos dados (por exemplo, valor = número do documento interno).

Dessa forma, nem todos os objetos (que podem ser solicitados) precisam ser predefinidos na estrutura do documento e em seguida ativados ou desativados por meio de condições. É possível definir um objeto de espaço reservado que, dependendo do conteúdo dos dados durante a geração do documento, pode ser substituído pelo objeto solicitado.

### Execute as seguintes etapas para definir um objeto como um objeto de variável:

1. Selecione a propriedade **VariableObjectDefinition** no item **Criar** do menu de contexto. Opcionalmente, a **VariableObjectDefinition** pode ser criada arrastando e soltando uma tabela de aliases sobre o objeto. Como resultado, as seguintes propriedades adicionais podem ficar visíveis abaixo da definição do objeto. Elas serão usadas para determinar o objeto a ser inserido:
  - **AttributeName**
  - **DataLink**
  - **DataSource**
  - **LookupSource**
  - **LookupKey**
  - **XPathExpression**

As descrições dessas propriedades podem ser encontradas em Propriedades de variável.

2. As propriedades **AttributeName** e **DataLink** ou **XPathExpression** são usadas para definir o campo de dados que contem o nome do objeto a ser inserido.
3. Além disso, você pode usar a propriedade **LookupSource** para definir uma tabela de aliases a ser usada para procurar o valor recuperado.

### Observe:

- As variáveis manuais não devem ser usadas em blocos de texto de variável, pois a caixa de diálogo de entrada para entradas manuais não é exibida quando o objeto de variável é inserido no documento.
- Você pode usar a propriedade de pacote **MissedObjects**, cujo conteúdo é apagado antes de cada geração de documento, para detectar os objetos de variável não encontrados.
- Em geral, os scripts do objeto de espaço reservado de variável são executados para os eventos **BeforeInsert** e **AfterInsert**.
- Se não houver scripts atribuídos ao objeto de espaço reservado para esses eventos, os scripts (se atribuídos) do objeto inserido é executado.

## Visualização do documento

Para usar a visualização de documento, execute estas etapas:

1. Vá para a guia **Ferramentas do Modus Editor - Teste** e clique no botão **ModusOne Client**.

2. Vá para a guia **Processos** e clique no botão **Criar documento**.
3. Clique no documento e selecione **Exibir documento no Word** do menu de contexto. O documento é exibido no Word.
4. Volte para o ModusOne Studio e volte para a janela do editor do documento.
5. Vá para a guia de visualização do documento no lado direito do painel. **Na visualização do documento**, clique no botão **Atualizar**.
6. Selecione o bloco de texto ou variável na estrutura do documento. O texto correspondente será realçado por um plano de fundo colorido na visualização do documento.

**Observação:**

O documento na janela de visualização do documento também pode ser impresso para uma impressora local.

Use o botão **Abrir arquivo** para carregar documentos externo do Word na janela de visualização.

## Variáveis globais

### Variáveis globais

Variáveis de documento globais estão disponíveis para todos os scripts que tiverem acesso ao pool **DOCUMENTPOOL**.

**Para editar variáveis globais, execute estas etapas:**

1. Mude para a guia **Variáveis globais** na parte inferior da barra da janela do editor.
2. Aqui, você pode usar o menu de contexto para criar novas variáveis ou excluir variáveis existentes.

### Propriedades de variáveis globais

**Valor padrão**

Valor aplicado se a variável não for preenchida.

**Manual**

Entrada manual True / False

**Tipo de variável****Valores possíveis:**

- Booleano
- Flutuante
- Longo
- String

## Grupos e ações

### Introdução

[Grupos](#) e [Ações](#) restringem ações de usuários de negócios por meio de regras pré-definidas.

### Grupos

Um grupo consiste de dois ou mais blocos de texto. O tipo de relacionamento define quantos blocos de texto no grupo podem ser ativados.

**Para criar um novo grupo, execute estas etapas:**

1. Abra um modelo de documento.
2. Clique na guia na região inferior direita da janela chamada Grupos/ações.
3. Na janela superior direita, selecione a função 'Novo grupo' do menu de contexto.
4. Use esse menu de contexto do grupo para definir o tipo de relacionamento:

Mínimo de um bloco de texto	Ao menos um bloco de texto no grupo deve ser ativado.
Exatamente um bloco de texto	Apenas um bloco de texto no grupo pode ser ativado.
Máximo de um bloco de texto	Apenas um ou nenhum bloco de texto no grupo pode ser ativado.

5. Atribua blocos de texto ao grupo arrastando e soltando.

### Observe:

Os usuários devem certificar-se de que o uso de diversos grupos não resultará em exclusão mútua.

Esse é particularmente o caso ao usar o mesmo objeto em diversos grupos.

A aplicação de uma regra de grupo não deve comprometer outra regra.

### Ações

As ações definem relacionamentos entre um bloco de texto principal e um ou mais blocos de texto secundários.

O status dos blocos de texto secundários depende do status do bloco de texto principal.

**Para criar uma nova ação, execute estas etapas:**

1. Abra um modelo de documento.
2. Clique na guia na região inferior direita da janela chamada **Grupos/ações**.
3. Arraste um bloco de texto do modelo de documento e solte-o na janela inferior direita.
4. Arraste os blocos de texto que dependem do primeiro bloco de texto.

- Use o menu de contexto para especificar o status dos blocos de texto.

**Exemplo:**

O bloco de texto principal recebe o status **Ativado**

O bloco de texto secundário recebe o status **Desativado**

Se o usuário mudar o bloco de texto principal para ativado, o bloco de texto secundário será definido automaticamente como desativado.

**Observe:**

Os usuários devem ter certeza de que as ações definidas, ou qualquer combinação delas, não comprometam a lógica de grupo predominante.

## Transferir propriedades

A transferência de propriedades entre blocos de texto e variáveis é um procedimento simples.

**Execute as seguintes etapas para fazer isso:**

- Pressione a tecla CTRL e clique no objeto cujas propriedades serão transferidas.
- Arraste o objeto de origem para o objeto de destino. O símbolo do cursor muda para um pequeno sinal de adição.

Os possíveis objetos de destino são qualquer objeto do mesmo tipo em qualquer nível. Você pode transferir entre objetos dentro da janela de objetos atual ou entre qualquer outra janela de objetos aberta.

Novamente, os objetos devem ser do mesmo tipo.

Ao transferir entre blocos de texto ou variável com IDs de objeto diferentes, a ação é concluída sem qualquer ação adicional.


Se os blocos de texto tiverem a mesma ID de objeto, uma nova janela será exibida com três opções:


- Somente propriedades de bloco de texto
- Somente propriedades das variáveis de bloco de texto
- Propriedades do bloco de texto e suas variáveis

- Selecione a opção desejada e clique no botão **Aplicar**.

## Teste de documentos

**Execute as seguintes etapas para testar um documento:**

- Abra um modelo de documento.
- Vá para a guia **Ferramentas do Modus Editor - Teste** e clique no botão **Compilar**. Mensagens de falha ou êxito são registradas na Janela de saída.
- Se o documento foi compilado com êxito, vá para a guia **Ferramentas do Modus Editor - Teste** e clique no botão **ModusOne Client**. 
- O documento é aberto como um teste de pacotes em uma nova guia.

5. Com a ajuda do menu de contexto, é possível alternar o status dos blocos de texto, inserir blocos de texto adicionais ou excluí-los.
6. Clique na guia **Processos** e no botão **Criar documento**. 
7. Se o documento foi criado com êxito, clique no documento e selecione **Exibir documento no Word** no menu de contexto.  
O documento é exibido no Word.  
Opcionalmente, o documento também pode ser aberto no Word simplesmente clicando duas vezes sobre ele.

## Propriedades do documento

### Geral

#### **NumberOfCopies**

Número de cópias a serem feitas desse documento.

Essa propriedade está disponível para documentos em pacotes.

#### **Archive**

Especifica se o documento deve ser arquivado após a criação.

Essa propriedade está disponível para documentos em pacotes.

#### **Expandable**

Define se o usuário pode ver a estrutura do documento e tem permissão para ativar ou desativar seus objetos.

#### **CustomIndex**

O valor pode ser definido livremente e usado para consultas subsequentes.

Essa propriedade está disponível para documentos em pacotes.

#### **CustomUIReference**

A interface de usuário do Word atribuída.

Essa propriedade está disponível para documentos em pacotes.

#### **DataID**

Essa propriedade armazena a seleção associada

Essa propriedade está disponível para documentos em pacotes.

#### **DocumentReference**

Nome do documento.

#### **PrintDefinition**

Use isso para definir a impressora, formulário e hora da impressão do documento.

A especificação de impressora e formulário é opcional.

Se os campos não estiverem configurados, eles devem ser definidos pelo usuário corporativo no cliente.

Nesse caso, ative a opção **Alterações permitidas** e também o tempo de impressão **desconhecido**.

Essa propriedade está disponível para documentos em pacotes.

### Password

Se uma senha for definida, os usuários podem apenas editar um bloco de texto com a propriedade **TextComponentProtection=TRUE** se eles souberem a senha.

Essa propriedade está disponível para documentos em pacotes.

### CopyLabel

O texto definido aqui é inserido no controle de contexto **CopyLabel** de uma cópia do documento.

Essa propriedade está disponível para cópias de documentos em pacotes.

Consulte também:

[Criar o controle de conteúdo 'CopyLabel'](#)

### OriginalID

ID exclusiva do objeto.

Essa propriedade está disponível para cópias de documentos em pacotes.

### Visible

Define se o documento fica visível aos usuários finais.

### Documento TextInsertionRule

Regra	Efeito
Activated	Se o documento não puder ser inserido durante a geração, a anulação não é forçada.
Deactivated	O documento é desativado e não pode ser ativado por um usuário.
Disabled	O documento não será inserido no pacote.
Required	A geração do pacote é abortada se o documento não puder ser inserido.

### Title

O valor definido aqui é o nome do documento que o usuário vê no cliente.

### Condições

O script definido aqui é interpretado após o script **BeforeInsert** ser executado.

A condição deve retornar o resultado **true** para o documento a ser inserido no pacote.

A condição tem acesso aos dados de documentos coletados pela `DataReference` correspondente.

**Consulte também:**

[Atribuição de scripts e condições](#)

## Modelo de dados

### Selection

Essa propriedade armazena a seleção associada.

## Eventos do documento

### Unload

Este script é executado após a inserção de todos os blocos de texto e antes do evento **DefinitionSorting**.

**Contexto:** [TextDocument](#)

### Load

Este script é executado quando um documento é carregado.

**Contexto:** [TextDocument](#)

### DefinitionSorting

Esse evento é definido para referências de documento em um pacote e é executado após os documentos serem criados.

O script é usado para definir atributos de classificação. Um critério de classificação é definido aqui como uma string de, no máximo, 256 caracteres.

Depois, isso é passado para o Odin como o atributo de classificação para cada trabalho.

Duas maneiras alternativas de definir a string de classificação são suportadas.

#### Alternativa 1

A string de classificação a ser usada é atribuída para a variável **Content**.

A propriedade **Sorting** do contexto **TextDocumentReference** é atribuída ao valor correspondente.

#### Exemplo:

```
Content = FormatStr("%s%s", "mySorting: ", SelVariable("PARTNER", "PAR_Vornam
e", ""))
```

#### Alternativa 2

Todos os scripts executados no contexto **TextDocumentReference** têm acesso de leitura e gravação à propriedade **Sorting**.

O conteúdo dessa propriedade é passado para o Odin como o atributo **Sorting** para cada trabalho.

Como na Alternativa 1, pode ser possível estender ou substituir um critério de classificação pré-definido nos eventos subsequentes.

**Exemplo de extensão do critério de classificação:**

```
dummy_Sort = Sorting
Sorting = FormatStr("%s%s", "MySortingAdd + ", dummy_Sort)
```

Contexto: [TextDocumentReference](#)

**DefinitionGrouping**

Esse evento é definido para referências de documento em um pacote e é executado após o evento **DefinitionSorting** .

O script é usado para definir atributos de agrupamento. Um critério de agrupamento é definido aqui como uma string de, no máximo, 256 caracteres.

Depois, isso é passado para o Odin como o atributo **Grouping** para cada trabalho.

Duas maneiras alternativas de definir a string de agrupamento são suportadas.

**Alternativa 1**

A string de agrupamento a ser usada é atribuída para a variável **Content**.

A propriedade **Grouping** do contexto **TextDocumentReference** é atribuída ao valor correspondente.

**Exemplo:**

```
Content = FormatStr("%s%s", "myGrouping: ", SelVariable("PARTNER", "PAR_Vorname", ""))
```

**Alternativa 2**

Todos os scripts executados no contexto **TextDocumentReference** têm acesso de leitura e gravação à propriedade **Grouping**.

O conteúdo dessa propriedade é passado para o Odin como o atributo de agrupamento para cada trabalho.

Como na Alternativa 1, pode ser possível estender ou substituir um critério de agrupamento pré-definido nos eventos subsequentes.

**Exemplo de extensão do critério de agrupamento:**

```
dummy_Group = Grouping
Grouping = FormatStr("%s%s", "MyGroupingAdd + ", dummy_Group)
```

Contexto: [TextDocumentReference](#)

**DefinitionArchiving**

Esse evento é definido para referências de documento em um pacote e é executado após o evento **DefinitionGrouping**.

O script é usado para definir atributos de arquivamento. O método `SetArchiveAttribute(name, value)` pode ser usado para definir atributos de arquivo 1-n que, depois, são passados para o Odin para cada trabalho.



Duas maneiras alternativas de definir os atributos de arquivo são suportadas.

### Alternativa 1

Os atributos a serem usados são atribuídos em uma string na forma de

Name=Value <CR><LF>Name=Value

por exemplo, criando uma lista de string e convertendo-a para uma string como uma string de atributo de arquivo atribuída à variável **Content**.

Ao fazê-lo, cada atributo individual é inserido no dicionário de arquivo.

#### Exemplo:

```
StringlistAdd("List", "ArchiveAttribute1=Value1")
StringlistAdd("List", FormatStr("%s%s", "ArchivCity=", SYSTEM.CITY))
Content = String("List")
```

### Alternativa 2

Todos os scripts executados no contexto **TextDocumentReference** podem chamar o método **SetArchiveAttribute** para inserir atributos no dicionário de arquivo.

Os atributos de cada dicionário de arquivo são passados para o Odin para cada trabalho.

#### Exemplo:

```
SetArchiveAttribute("MyArchiveVar1", "Value1")
SetArchiveAttribute("MyArchiveVar2", "Value2")
```

**Contexto:** [TextDocumentReference](#)

**Consulte também:**

[Caso especial: atributos de arquivamento idênticos para todos os documentos em um pacote](#)

### AfterInsert

Esse evento é definido para as referências de documento em um pacote.

Esse script é executado após o documento ser inserido no pacote.

**Contexto:** [TextDocumentReference](#)

### BeforeInsert

Esse evento é definido para as referências de documento em um pacote.

Esse script é executado antes do documento ser inserido no pacote.

Se InsertObject for True após o script ser executado, o processamento é continuado com a avaliação da condição.

**Contexto:** [TextDocumentReference](#)

## Definições de objetos

### VariableObjectDefinition

Essa propriedade contém as configurações para documentos variáveis.

Se uma tabela de alias for solta em um documento, ou a `VariableObjectDefinition` for criada de um menu de contexto, propriedades adicionais são exibidas.

Estas propriedades determinam qual documento será realmente inserido.

- Nome do atributo
- `DataLink`
- `DataSource`
- `LookupSource`
- `LookupKey`
- `XPathExpression`

**Consulte também:**

[Objetos de variável](#)

# Pacotes

## Introdução

Encaminhamentos são agrupamentos lógicos de documentos relacionados.

Agrupar documentos relacionados a processos de negócios em um único objeto permite a geração e o arquivamento de documentos em uma etapa de processamento comum.

Além disso, é possível processar cópias dos documentos originais na mesma etapa de processamento único mencionada acima.

## Criar um pacote

**Para criar um pacote, execute estas etapas:**

1. Selecione **Novo -> Pacote** do menu de contexto do Navigator.
2. Digite o nome do pacote na caixa de diálogo seguinte.
3. Arraste os documentos necessários do Navigator e solte-os na janela do pacote.

Se os modelos de dados já tiverem sido atribuídos aos documentos em um pacote, eles são listados no painel inferior.

Arraste e solte uma seleção do Navigator em um documento para alterar a seleção atribuída. Cada documento pode ter sua própria seleção atribuída.

### Menu de contexto

O menu de contexto de um documento disponibiliza diversas funções:

#### Remover DataID

Exclui o valor da propriedade DataID.

#### Obrigatório/Ativar/ Desativado/Desabilitado

Alterar o status do documento (propriedade TextInsertionRule)

#### Criar cópia

Insere uma cópia do documento selecionado.

#### Abrir modelo de documento

Abre o modelo de documento para edição.





#### Remover

Remove o modelo do documento selecionado.

**As ferramentas do editor fornecem as seguintes funções:**



Abre modelo do documento selecionado.

	Inserir uma cópia do modelo de documento selecionado.
	Definir um novo grupo. Consulte <a href="#">Grupos</a>
	Compilar um pacote
	Testar o documento no <b>ModusOne Client</b> integrado.

## Cópias de documento

Para criar uma cópia de documento em um pacote, execute estas etapas:

1. Selecione o documento.
2. Selecione a opção **Criar cópia** das Ferramentas do Modus Editor. Como alternativa, selecione essa função do menu de contexto do documento.

Uma cópia do documento é inserida e marcada com um C.

## Atribuir um modelo de dados

Um pacote pode usar diversas seleções diferentes.

Se os modelos de dados já tiverem sido atribuídos aos documentos em um pacote, eles são listados no painel inferior.

Se você precisar de seleções adicionais, pode arrastá-las do Navigator e soltá-las no painel, ou diretamente em um documento.

Se o documento já tiver uma seleção definida, essa operação sobrescreve o link.

## Encaminhar um pacote

Caso um usuário não deva ter permissão para imprimir um pacote criado por ele mesmo, defina a propriedade **BlockBundle** como True em um script de pacote.

Ao clicar no botão **Imprimir e arquivar** em um **ModusOne Client**, a caixa de diálogo para encaminhamento será exibida.

Um grupo de usuários e/ou um único usuário podem ser selecionados como destino para o pacote.

Se um grupo de usuários e um usuário forem selecionados, o pacote será encaminhado para o grupo e o usuário individual.

Como outra opção, você pode usar o script de pacote para definir as propriedades de **DestinationGroup** e/ou **DestinationUser**.

Nesse caso, uma caixa de mensagem de informação é exibida e o processo é encaminhado automaticamente para o grupo e/ou usuário especificado.

**Script de exemplo**

```
BlockBundle = true
DestinationGroup = "Adminrole"
DestinationUser = "Administrator"
```

## Grupos e ações

### Introdução

[Grupos](#) e [Ações](#) restringem ações de usuários de negócios por meio de regras pré-definidas.

### Grupos

Um grupo consiste de dois ou mais documentos. O tipo de relacionamento define quantos documentos no grupo podem ser ativados.

**Para criar um novo grupo, execute estas etapas:**

1. Abra um pacote.
2. Clique na guia na região inferior direita da janela chamada Grupos/ações.
3. Na janela superior direita, selecione a função 'Novo grupo' do menu de contexto.
4. Use esse menu de contexto do grupo para definir o tipo de relacionamento:

Ao menos um documento:	Ao menos um documento no grupo deve ser ativado.
Exatamente um documento:	Apenas um documento no grupo pode ser ativado.
Máximo de um documento:	Apenas um ou nenhum documento no grupo pode ser ativado.

5. Atribua documentos ao grupo arrastando e soltando.

**Observe:**

Os usuários devem certificar-se de que o uso de diversos grupos não resultará em exclusão mútua.

Esse é particularmente o caso ao usar o mesmo objeto em diversos grupos.

A aplicação de uma regra de grupo não deve comprometer outra regra.

### Ações

As ações definem relacionamentos entre um documento principal e um ou mais documentos secundários. O status dos documentos secundários depende do status do documento principal.

**Para criar uma nova ação, execute estas etapas:**

1. Abra um pacote.
2. Clique na guia na região inferior direita da janela chamada **Grupos/ações**.

3. Arraste um documento do pacote e solte-o na janela inferior direita.
4. Arraste os documentos que dependem do primeiro documento.
5. Use o menu de contexto para especificar o status do documento.

**Exemplo:**

O documento principal recebe o status **Ativado**

O documento secundário é definido automaticamente com o status **Desativado**.

Caso um usuário mude o documento principal para ativado, o documento secundário será definido automaticamente como desativado.

**Observe:**

Os usuários devem ter certeza de que as ações definidas, ou qualquer combinação delas, não comprometam a lógica de grupo predominante.

## Pacotes dinâmicos



Em um pacote dinâmico, os documentos a serem inseridos em um pacote e suas propriedades respectivas são definidos na etapa de criação do pacote real.

**Para criar um pacote dinâmico, execute estas etapas:**

1. Crie um novo script com o contexto [TextDocumentCollection](#).  
Esse contexto fornece as funções necessárias para inserir documentos e cópias de documentos em um pacote.
2. Use as instruções [AddDocumentByName](#) ou [AddDocumentById](#) para especificar os documentos a serem inseridos no pacote.
3. Abra o pacote no qual os documentos serão inseridos.
4. Atribua a propriedade **AfterDataSelection** do pacote ao script recém criado.

## Teste de pacotes

**Execute as seguintes etapas para testar um documento:**

1. Abra o pacote.
2. Vá para a guia **Ferramentas do Modus Editor - Teste** e clique no botão **Compilar**. Mensagens de falha ou êxito são registradas na Janela de saída.
3. Se o pacote foi compilado com êxito, vá para a guia **Ferramentas do Modus Editor - Teste** e clique no botão **ModusOne Client** .
4. O pacote é aberto em uma nova guia.
5. Com a ajuda do menu de contexto, você pode alternar o status dos documentos ou inserir documentos adicionais.  
O status dos blocos de texto pode ser alternado no painel direito.
6. Vá para a guia **Processos** e clique no botão **Criar documentos** .  
A geração do pacote cria todos os documentos originais ativados.  
As cópias não são criadas até que os documentos sejam impressos ou enviados para o Odin.

7. Se os documentos foram criados com êxito, clique em um documento e selecione **Exibir documento no Word** no menu de contexto.  
O documento é exibido no Word.  
Opcionalmente, o documento também pode ser aberto no Word simplesmente clicando duas vezes sobre ele.

### **Menu de contexto de um documento**

O menu de contexto de um documento oferece as seguintes funções:

#### **Alternar status**

Ativar/desativar o documento.

#### **Exibir documento no Word**

Abre um documento gerado no Word.

#### **Salvar documento como...**

Abre uma janela de diálogo para permitir que você salve o documento gerado em um diretório específico.

#### **Excluir documento**

Remove o documento do pacote.

#### **Atribuir documento externo**

Esse item de menu está disponível para documentos estáticos ativos e é usado para substituir documentos no pacote com documentos externos.

Os documentos em um pacote podem ser substituídos por documentos externos do mesmo tipo.

Para os documentos externos, o texto (documento externo) é exibido.

É possível remover o link com o item de menu

#### **Remover documento externo**

### **Observação:**

O uso do método **LoadExternalDocument** em um script sobrescreve os valores definidos nessa função para o objeto correspondente.

### **Menu de contexto de um bloco de texto**

O menu de contexto de um bloco de texto oferece as seguintes funções:

#### **Alternar status**

Ativar/desativar o documento.

#### **Excluir bloco de texto**

Remove o bloco de texto do documento.

#### **Atribuir documento externo**

Esse item de menu está disponível para blocos de texto ativos e é usado para substituir os blocos de texto existentes em um documento por documentos externos.

Os blocos de texto podem ser substituídos por documentos do Word.

Para os documentos externos, o texto (documento externo) é exibido.

É possível remover o link com o item de menu

### Remover documento externo

#### Observação:

O uso do método **LoadExternalDocument** em um script sobrescreve os valores definidos nessa função para o objeto correspondente.

## Caso especial: Atributos de arquivo idênticos para todos os documentos de um pacote

Se todos os documentos de um pacote receberem os mesmos atributos de arquivo, é possível usar o evento de pacote [InitialisePrintAndArchive](#) para fazer isso em uma única etapa.

Os atributos de arquivo 1:n definido nele são passados para o Odin para cada um dos trabalhos no pacote.

Dois modos alternativos de definir os atributos de arquivo para todos os documentos são suportados.

#### Alternativa 1

Os atributos a serem usados são definidos como uma string em um script atribuído ao evento de pacote [InitialisePrintAndArchive](#) (corresponde a AdditionalInformationScript no modus 4), no formato

```
Name=Value <CR><LF>Name=Value
```

que em seguida é atribuído como a string de atributo de arquivo da variável ArchiveData(Content).

Ao fazer isso, cada atributo individual é inserido nos dicionários de arquivos de todos os documentos.

A string pode ser criada, por exemplo, criando uma lista de strings e convertendo-a em uma string.

#### Exemplo:

```
StringlistAdd("List", FormatStr("%s%s", "CreatorUser=", ModusUser.UserId)
StringlistAdd("List", FormatStr("%s%s", " MyArchiveVar2=", "Value2")
ArchiveData = String("List")
```

#### Alternativa 2

Todos os scripts executados no contexto [TextDocumentCollection](#) podem chamar o método **SetArchiveAttribute** para inserir atributos nos dicionários de arquivos de todos os documentos.

Os atributos de arquivo são passados em seguida para o Odin para cada um dos trabalhos no pacote.

O evento [BeforePrintingDocument](#) não é adequado a esse procedimento porque, diferente de outros eventos de pacote, esse evento não é executado apenas uma vez mas para cada documento individual no pacote.

#### Exemplo:

```
SetArchiveAttribute("CreatorUser", ModusUser.UserId)
SetArchiveAttribute("MyArchiveVar2", "Value2")
```



## Transferir propriedades

A transferência de propriedades entre documentos é um procedimento simples.

### Execute as seguintes etapas para fazer isso:

1. Pressione a tecla **CTRL** e clique no objeto cujas propriedades serão transferidas.
2. Arraste o objeto de origem para o objeto de destino. O símbolo do cursor muda para um pequeno sinal de adição.

## Propriedades do pacote

### Geral

#### DataID

Seleção atribuída ao pacote.

Scripts de pacote podem acessar os dados coletados pela seleção.

### Eventos de pacote

Nestes scripts, você pode acessar os dados das referências de dados atribuídas ao pacote e as propriedades e métodos do contexto **TextDocumentCollection**.

#### InitializeDocumentCreation

Esse evento é executado após o evento **BeforeCreatingDocuments** e antes da geração do documento.

Se, após a execução do script, a propriedade **InsertObject for True**, o processamento é continuado com a etapa de geração de documentos.

Se for **False**, o processamento do pacote é encerrado sem a criação de documentos.

Contexto: [TextDocumentCollection](#)

#### InitializePrintAndArchive

O script definido aqui é executado uma vez após o evento **BeforePrintingAndArchiving** e adiciona informações de arquivamento definidas na variável Content pelo script para todos os documentos contidos no pacote.

Contexto: [TextDocumentCollection](#)

#### Consulte também:

[Caso especial: atributos de arquivamento idênticos para todos os documentos em um pacote](#)

#### AfterDataRetrieval

O script definido aqui é executado após a recuperação de dados.

Contexto: [TextDocumentCollection](#)

#### AfterCreateDocuments

Esse evento é executado após a criação de todos os documentos, ao final do processamento do pacote.

**Contexto:** [TextDocumentCollection](#)

### **BeforePrintingDocument**

O evento é executado antes da impressão de cada documento individual.

**Contexto:** [TextDocumentReference](#)

### **BeforeCreatingDocuments**

Esse evento é executado uma vez, no início de um pacote.

Se, após a execução do script, a propriedade **InsertObject** for **True**, o processamento do pacote é continuado com o evento **InitializeDocumentCreation**.

Se for **False**, o processamento do pacote é encerrado sem a criação de documentos.

**Contexto:** [TextDocumentCollection](#)

### **BeforePrintingAndArchiving**

Esse script é executado uma vez, antes do início da impressão e arquivamento do pacote.

Se, após a execução do script, a propriedade **AllowPrint** for **False**, um erro lógico predefinido é lançado.

**Contexto:** [TextDocumentCollection](#)

# Scripts e condições

## Introdução

### Scripts

Objetos para a resolução de tarefas específicas.

A linguagem do script MonaLisa oferece uma vasta gama de possibilidades para herdar o escopo funcional do ModusOne.

Os scripts podem ser usados para implementar requisitos que o uso de objetos padrão autônomos não pode cobrir totalmente e podem ser usados para ganhar acesso total a todas as lógicas de negócio disponíveis armazenadas nos vários pools de variáveis.

A linguagem de script abrange as funções de chamadas para vários contextos e oferece estruturas de controle como loops e ramificações.

Para obter mais ramificações na sintaxe de linguagem do script, consulte o capítulo Funções MonaLisa.

### Condições

Um script que em conexão com o processo de geração do documento verifica a validade de uma instrução.

O resultado que passou para a variável **Resultado** é **Verdadeiro** ou **Falso**.

As condições são usadas para ativar ou desativar variáveis, blocos de texto ou documentos de acordo com o valor retornado.

## Criar scripts e condições

### Criar um script

**Para criar um script, execute estas etapas:**

1. Selecione **Novo -> Script** do menu de contexto do Navigator.
2. Digite o nome do script na caixa de diálogo seguinte.
3. Defina o contexto obrigatório nas propriedades.  
Isso determina os objetos aos quais o script pode ser atribuído e quais propriedades e métodos estão disponíveis.
4. Digite o script, ou seja, as instruções para resolver um trabalho específico na janela do editor. O escopo completo da linguagem de script MonaLisa está disponível para o uso.
5. Compile o script com o botão **Compilar**  na guia **Ferramentas do editor de observações de script - Controle**.  
Mensagens de êxito ou falha são registradas na Janela de saída.

### Criar uma condição

**Para criar uma nova condição, execute estas etapas:**

1. Selecione **Novo -> Condição** do menu de contexto do Navigator.
2. Digite o nome da condição na caixa de diálogo seguinte.
3. Defina o contexto obrigatório nas propriedades.  
Isso determina os objetos aos quais a condição pode ser atribuída e quais propriedades e métodos estão disponíveis.
4. Digite o código da condição na janela do editor. Uma condição simples verifica, por exemplo, se um campo contém um valor específico.  
O escopo completo da linguagem de script MonaLisa está disponível para o uso.  
Use o [Browser de seleção](#) para acessar as variáveis do modelo de dados.
5. Atribua um valor booleano à variável **Result** como resultado de uma instrução condicional.  
**Exemplo:**  
Result = true
6. Compile a condição pressionando o botão **Compilar**  na aba **Ferramentas do editor de observações de script - Controle**.  
Mensagens de êxito ou falha são registradas na Janela de saída.

### Sintaxe

Você pode usar operadores para vincular diversas expressões lógicas juntas em uma condição.

Expressões combinadas devem estar entre colchetes para manter sua finalidade lógica.

Expressões lógicas aninhadas também são possíveis.

### Exemplo:

```
Result = ( ( SelVariable("PARTNER", "PAR_TITLECODE") = "1" ) OR (
SelVariable("PARTNER", "PAR_TITLECODE") = "2" ) )
```

### Operadores

Os seguintes operadores estão disponíveis:

Operador	Definição
=	O atributo deve ser igual ao valor
<>	O atributo não deve ser igual ao valor
<	O atributo deve ser menor que o valor
>	O atributo deve ser maior que o valor
<=	O atributo deve ser menor ou igual ao valor
>=	O atributo deve ser maior ou igual ao valor
<>"	O atributo não pode estar vazio
AND()	AND lógico

OR()

OR lógico

## Conclusão do código

Para facilitar o uso, combinações de teclas de atalho podem ser usadas durante a escrita de código.

### Funções padrão

A combinação de teclas **CTRL + Espaço** abre uma lista de todas as funções padrão Monalisa.

Essa lista também mostra informações sobre a função da instrução e seus parâmetros.

Usando enter ou um clique duplo, você pode inserir o comando no editor de script.

### Propriedades, métodos e pools

A combinação de teclas **CTRL + ALT + Espaço** abre uma lista de propriedades, métodos e pools disponíveis no contexto atual.

### Estruturas de controle

As estruturas de controle podem ser acessadas pela combinação de teclas **CTRL + K + X**.

Alternativamente, você pode pressionar a tecla Tab para uma das palavras-chave listadas para exibir a estrutura de comando correspondente.

Para obter mais informações, consulte o capítulo [Estruturas de controle](#).

### Outras combinações de tecla

Atalho	Efeito
Alt + seta para a direita ou CTRL + espaço	Mostra a conclusão da palavra.
CTRL + K + C	Marca a linha atual do código como comentário.
CTRL + K + U	Remove a sintaxe do comentário da linha de código atual.
CTRL + K + F	Recua a linha do código selecionada de forma igual às linhas de código em volta.
SHIFT + ALT + T	Move uma linha com o cursor para baixo da próxima linha.
CTRL + U	Converte o texto selecionado em minúscula.
CTRL + SHIFT + U	Converte o texto selecionado em maiúscula.

CTRL + SHIFT + espaço	Mostra as informações sobre o parâmetro atual.
CTRL + DEL	Exclui a palavra à direita do cursor.
CTRL + Backspace	Exclui a palavra à esquerda do cursor.
CTRL + F	Mostra a caixa de diálogo de pesquisa.
CTRL + H	Mostra a caixa de diálogo de pesquisa e substituição.
F3	Continua a pesquisa para baixo do texto.
SHIFT + F3	Continua a pesquisa para cima do texto.
CTRL + A	Seleciona todo o conteúdo.
CTRL + W	Seleciona a palavra onde o cursor está posicionado ou a palavra à direita do cursor.

## Definir bookmarks

Para ajudá-lo a ir direto a linhas específicas do código fonte mais rapidamente, recomendamos o uso de **Bookmarks**.

- A combinação de teclas **CTRL + K + Número** define um bookmark na linha atual do editor de script.
- A mesma combinação de teclas o exclui.
- Usando a combinação de teclas **CTRL + Número** você pode ir diretamente a um bookmark definido.

## Informações de contexto

Mude para a guia abaixo da janela do editor de script chamada **Informações de contexto**.

As informações de contexto mostram quais pools de variáveis e componentes de processo estão disponíveis de acordo com o contexto especificado.

## Parâmetros do script

É possível especificar parâmetros e seus valores padrão para um script quando o script é chamado de outro script com a função passagem de parâmetro.

Altere para a guia abaixo do editor de script chamada **Parâmetro**.

Ao pressionar o botão **Novo parâmetro**, você pode inserir novos parâmetros.

Consulte também: [CallSystem](#)

## Comentários

A janela de comentários é usada para inserir quaisquer comentários sobre o script.

## Ferramentas do editor

A barra de ferramentas oferece diversas funções para processamento de scripts e condições.

### Exportar script

Permite que você salve um script em um arquivo.

### Serviço de depuração

Essa função permite que você adicione o script aberto ao serviço de depuração, ou remova-o, se necessário.

### Compilar

Scripts e condições devem ser compilados. O botão correspondente está disponível na barra de ferramentas.

### Executar

Para executar um script ou condição, clique no botão **Executar**.

O código é executado em segundo plano.

### Depurar



Os scripts que contêm apenas as funções padrão do MonaLisa podem ser executados diretamente no modo de depuração, se o contexto **None** for atribuído.

Para obter mais informações, consulte o capítulo

[Testar scripts e condições](#)

## Browser de seleção

**Para usar o Browser de seleção, execute estas etapas:**

1. Clique no botão Browser de seleção  nas Ferramentas do editor de observações de script. Outra janela é aberta no editor, na qual você pode escolher as seleções disponíveis no sistema.
2. Clique no botão  ao lado da seleção para listar seus objetos de dados.
3. Clique em um objeto de dados para listar suas variáveis no painel inferior.
4. Você pode arrastar e soltar variáveis daqui para a janela do editor. No edito, o comando **SelVariable** é criado.

**Exemplo:**

```
SelVariable("Partner", "PAR_Titlecode", "")
```

**Criptografar senha**

Essa função abre outra janela de diálogo, onde você pode criptografar uma senha e copiá-la para a área de transferência.

**Editar**

Fornece o número de funções para a edição do código do script.

**Saída**

Mensagens de êxito ou falha são registradas na janela de saída.

## Atribuição de scripts e condições

[▶ Mostrar tudo](#)

Scripts e condições podem ser atribuídos a uma variável, bloco de texto, documento e eventos de pacote, além de blocos e componentes de designer de processo.

Há duas maneiras de fazer isso.

**Scripts****Conclua as etapas a seguir para atribuir um script do Navigator:**

1. Abra o objeto ao qual o script será atribuído.
2. Selecione o script no Navigator.
3. Arraste e solte o script do Navigator no nome do objeto na janela do editor. Isso é possível apenas quando o contexto do script corresponde ao do objeto selecionado.

**Exemplo:**

um script com o contexto **TextComponent** pode ser atribuído a um bloco de texto.

4. Na consulte de evento exibida, defina se o script pode ser executado antes ou depois da inserção de um objeto. Essa atribuição está disponível para variáveis, blocos de texto, documentos e pacotes.

**Conclua as etapas a seguir para atribuir um script do Object Inspector:**

1. Abra o objeto ao qual o script será atribuído.
2. Selecione o evento ao qual o script será atribuído no Object Inspector.
3. Clique no botão com reticências.
4. Selecione o script necessário na janela de diálogo seguinte. Apenas os scripts com o contexto adequado estão disponíveis para seleção.



Na janela do editor, você verá uma linha acima da caixa, ao lado do objeto quando um script

**BeforeInsert** foi atribuído:



Uma linha abaixo da caixa sinaliza que um script **AfterInsert** foi atribuído.

## Condições

### Conclua as etapas a seguir para atribuir uma condição do Navigator:

1. Abra o objeto ao qual a condição será atribuída.
2. Selecione a condição no Navigator.
3. Arraste e solte a condição do Navigator no nome do objeto na janela do editor. Isso é possível apenas quando o contexto da condição corresponde ao do objeto selecionado.

#### Exemplo:

uma condição com o contexto **TextComponent** pode ser atribuída a um bloco de texto.

4. Na janela do editor, você verá ponto de interrogação ao lado objeto quando uma condição foi atribuída.

### Conclua as etapas a seguir para atribuir um script do Object Inspector:

1. Abra o objeto ao qual a condição será atribuída.
2. Selecione a condição de propriedade no Object Inspector.
3. Clique no botão com reticências.
4. Selecione a condição necessária na janela de diálogo seguinte. Apenas as condições com o contexto adequado estão disponíveis para seleção.

#### Observação:

- Um clique duplo em um evento ou condição abre o script correspondente.
- Se um script não foi atribuído, um novo script é criado. Ao fazer isso, o contexto correspondente ao evento é selecionado automaticamente.
- Você pode ver quais scripts ou condições são atribuídos a um objeto no Object Inspector. A função **Reset (null)** no menu de contexto revoga a atribuição.


## Scripts de teste e condições

### Teste de scripts e condições que não precisam de contexto para execução

Scripts e condições que não usam métodos de qualquer contexto específico podem ser executados durante a edição no modo de depuração.

#### Execute as seguintes etapas:

1. Abra o script ou a condição.
2. No Object Inspector, defina a propriedade **Context** com o valor **None**.

3. Clique no botão **Executar no modo de depuração**  em **Ferramentas do editor de observações de script** ou pressione a tecla **F9**.  
O script em seguida é aberto em uma nova janela do modo de depuração.
4. Execute o script usando os botões nas ferramentas do Depurador.

### Teste de scripts e condições que precisam de contexto para execução

Scripts e condições que usam métodos de contextos específicos não podem ser executados durante a edição no modo de depuração. Em vez disso, eles precisam ser executados no contexto correspondente.

#### Execute as seguintes etapas:

1. Selecione o script a ser depurado no Navigator e use o item **Adicionar a serviço de depuração** do menu de contexto;
2. Abra o documento no processo Modus, no qual o script é executado e inicia a criação de documentação ou do processo, conforme o caso.
3. Assim que o script é executado, a janela do depurador é aberta e o script pode ser testado passo a passo usando as **Ferramentas do depurador**.
4. Se o script não precisar mais ser executado no modo de depuração, selecione-o no Navigator e escolha o item **Remover de serviço de depuração** do menu de contexto.  
Opcionalmente, você pode selecionar o item **Gerenciador de depuração** do menu principal do Studio para abrir o gerenciador de depuração e remover os scripts.

### Ferramentas do depurador

#### Executar (CTRL + F9)

Executa o script completo e fecha o Depurador.

Não leva em consideração os pontos de ruptura.

#### Executar (F9)

Executa o script completo, mas não fecha o Depurador.

Se houver pontos de ruptura definidos, o script irá parar na linha de código correspondente e pode ser reiniciado nessa posição.

#### Definir/remover ponto de ruptura (F5)

Define ou remove um ponto de ruptura na posição atual.

#### Depuração completa (F7)

Executa a linha de código atual.

Se essa instrução for [CallSystem](#), o Depurador se ramifica no script sendo chamado.

#### Etapa de procedimento (F8)

Executa a linha de código atual.

### **Parar (CTRL + F2)**

Encerra o script e fecha o Depurador.

### **Monitoramento rápido**

Mostra os valores de variáveis atuais

## Funções MonaLisa

### **Migração de scripts do modus 4**

Se você importou scripts do modus4, os ajustes devem ser feitos para os seguintes comandos:

#### **Função OnError - Aumentar**

Essa função deve ser substituída por uma das estruturas de controle **Tentar – Finalmente**.

**Consulte também:**

[Tratamento de exceção](#)

#### **Análise de função**

A análise de informação deve ser incluída entre aspas.

**Consulte também:**

[Analisar](#)

#### **Verificação de função**

A verificação variável sempre pode ser especificada sem aspas.

**Consulte também:**

[Estruturas de controle](#)

#### **Função GetLookUpValue**

O número de parâmetros foi alterado.

**Consulte também:**

[GetLookupValue](#)

#### **Inicializa eventos, InitializeGUI e OnProcessEnd**

Estes eventos não estão mais disponíveis.

#### **Funções de entrada, MessageDialog e DropDownbox**

Estas funções **FatClient** não estão mais disponíveis.

#### **Função GoTo**

Esta função não está mais disponível.

## Tipos de dados

O MonaLisa funciona com os tipos de dados Integer, BigInteger, Float e String.

### Intervalos de valores dos tipos de dados

Tipo de dados		Mín.	Máx.
I	Integer (int32)	-2147483648	2147483647
L	BigInteger (int64/long)	-9223372036854775808	9223372036854775807
F	Float (double)	-1.7976931348623157E+308	1,7976931348623157E+308
S	String		

### Tipos de dados de resultado

A tabela a seguir lista os tipos de dados de resultado de acordo com as operações de dados.

Operandos		Tipo de dados de resultado			
Operando 1	Operando 2	Operador +	Operador -	Operador *	Operador /
S	S	S	Erro	Erro	Erro
S	I	S	Erro	Erro	Erro
S	L	S	Erro	Erro	Erro
S	F	S	Erro	Erro	Erro
I	S	S	Erro	Erro	Erro
I	I	I / L	I / L	I / L	F
I	L	L	L	L	F
I	F	F	F	F	F
L	S	S	Erro	Erro	Erro
L	I	L	L	L	F
L	L	L	L	L	F
L	F	F	F	F	F

F	S	S	Erro	Erro	Erro
F	I	F	F	F	F
F	L	F	F	F	F
F	F	F	F	F	F

## Estruturas de controle

### Geral

Use as estruturas de controle para controlar o fluxo de scripts e condições, por exemplo, para responder a conteúdos de dados diferentes.

### CheckEvery

Verifica a validade da variável. Se o primeiro parâmetro passado for **EVERY**, mais de um acordo é possível.

A variável a ser verificada é passada como o segundo parâmetro.

- O valor verifica a validade do valor passado.
- Usar a combinação de chaves **when** permite que você verifique mais condições.
- As instruções após **when-any** são executadas quando uma ou mais condições são atendidas.
- As instruções após **when-all** são executadas quando todas as condições são atendidas.
- As instruções após **when-none** são executadas quando nenhuma das condições é atendida.

### Exemplo:

```
a = 11
erg = 0
c = false
;
Check("every", a)
  value(1)
    erg = erg + 1 ; if a is equal to 1
  when(c = true)
    erg= erg + 1 ; if c is TRUE
  When-Any
    message = "any" ; if a is equal to 1 OR c is TRUE
  When-All
    message = "all" ; if a is equal to 1 AND c is TRUE
  When-None
    message = "none" ; if a is not equal to 1 AND c is not TRUE
end-check
```

## CheckFirst

Verifica a validade da variável. Se o primeiro parâmetro passado for **FIRST**, apenas as instruções após o primeiro acordo são executadas.

A variável a ser verificada é passada como o segundo parâmetro.

- O valor verifica a validade do valor passado.
- Usar **when** permite que você verifique mais condições.
- As instruções após **when-any** são executadas quando uma ou mais condições são atendidas.
- As instruções após **when-all** são executadas quando todas as condições são atendidas.
- As instruções após **when-none** são executadas quando nenhuma das condições é atendida.

### Exemplo:

```
a = 11
erg = 0
c = false
Check("FIRST", a)
  value(1)
    erg = 1    ; if a is equal to 1
  when(c = true)
    erg=2      ; if a is not equal to 1 AND c is TRUE
  When-Any
    Erg=3      ; if a is equal to 1 OR c is TRUE
  When-All
    Erg=4      ; if a is equal to 1 AND c is TRUE
  When-None
    Erg=5      ; if a is not equal to 1 AND c is not TRUE
end-check
```

## Condição If

Verifica se uma condição entre colchetes é válida. Se sim, os comandos a seguir, até **End-if** são executados.

Se a condição entre colchetes não for válida, o script é continuado após o comando **End-if**.

Cada condição individual deve estar entre colchetes.

Diversas condições, vinculadas por **and** ou **or** devem estar entre colchetes adicionais.

### Exemplos

```
if (a < 10)
  b = 0
end-if
if ( (a = 10) and (b = 30) )
  c = 0
end-if
```

## Condição IfElse

Verifica se a condição entre parênteses é válida. Se for, os seguintes comandos até **Else** são executados.

Em seguida, o script continua com as instruções seguindo o respectivo comando **End-If**.

Se a condição entre parênteses não for válida, os comandos entre **Else** e o respectivo **End-If** são executados.

### Exemplo:

```
if (a < 10)
  b = 0
else
  b = 1
end-if
```

## Loops While

Executa as instruções em loop enquanto a condição entre parênteses for **VERDADEIRA**.

A condição é avaliada antes de as instruções serem executadas.

### Exemplo:

```
while (a < 100)
  a = a + 1
end-while
```

## Region

Define uma seção de um script que pode ser exibida ou oculta usando os símbolos + / -.

Isso ajuda a gerenciar melhor scripts grandes.

### Exemplo:

```
#region Info Text
...
...
#endregion
```

## Tratamento de exceção

### Introdução

Exceções ocorridas durante a execução do script podem ser detectadas usando o controle **Try - Finally**.

### TryFinally

Se ocorrer um erro durante a execução de uma instrução, o script normalmente terminará com uma exceção.

Se a instrução estiver dentro de um try-block, as instruções no finally-block serão executadas antes do script ser concluído.

**Exemplo:**

```
try
  InitDatabase("ODIN")
finally
  Protocol("Inside Finally-block", 9)
end-try
```

**TryOnError**

Se ocorrer um erro durante a execução de uma instrução em um bloco try, use o comando **Raise** para ignorar o rótulo de erro definido em **OnError**.

Apenas o primeiro bloco OnError correspondente em um try-block será executado.

Selecione a ordem correta dos blocos OnError de maneira que os blocos OnError específicos sejam executados antes dos blocos mais gerais.

**Exemplo:**

Consulte [TryOnErrorFinally](#)

**TryOnErrorFinally**

Se ocorrer um erro durante a execução de uma instrução em um bloco try, use o comando **Raise** para ignorar o rótulo de erro definido em **OnError**.

O código no bloco finally é executado depois que o bloco OnError foi processado.

**Exemplo:**

```
;the special object "Error" contains information about the last error
;If no error occurred - all attributes contain an empty string
;
;Error.Category: the category identifies the sub-systems that threw the error
(for example: "OdinBusinessFacade")
;Error.Token: the token is a one-word identifier for the error. Can be
specified if you use the Raise method
;Error.Message: contains the error message
Protocol("before try: category '{0}'; error token: '{1}'; error message:
'{2}'; ", 0, Error.Category, Error.Token, Error.Message)
;
;modify this value to play with the different OnError code-blocks
a = 10

try
  if(a < 10)
    ;This exception will be caught in the OnError("BUSINESS_ERR") block
    Raise("BUSINESS_ERR","a cannot be less than 10 sein")
  end-if
;
;This exception will be caught in the OnError() block
;because there is no OnError("Err") block
```



```

    Raise("Err", "Test-Raise")
;
;
OnError("BUSINESS_ERR")
    ;this code is executed if the exception token is "BUSINESS_ERR"
    ;the exception token is the first parameter of the Raise method
    Protocol("BUSINESS_ERR: {0}", 0, Error.Message)
;
    ;you can also rethrow the error
    ReRaise()
;
OnErrorCategory("OdinBusinessFacade")
    ;this code is executed if the category of the exception is
    "OdinBusinessFacade"
    Protocol("odin error caught", 0)
;
OnError()
    ;this code is executed if no matching OnError() was found up to this line
    ;information about the exception can be retrieved from the Error object
    Protocol("error category: {0}; error token: '{1}'; error message: '{2}';
", 0, Error.Category, Error.Token, Error.Message)
;
Finally
;
    ;code in the finally block is ALWAYS executed
    ;this is the right place to free resources (database connections, file
handles)
    Protocol("Inside Finally-block", 9)
End-try
;
Protocol("after finally: error category: {0}; error token: '{1}'; error
message: '{2}'; ", 0, Error.Category, Error.Token, Error.Message)

```

## Funções padrão

### Acrescentar

Insere um conjunto de dados e preenche os campos com valores.

### Sintaxe

```
Append(datasetName: String ,fields: String[])
```

### Parâmetro

Parâmetro	Significado
datasetName	Nome do conjunto de dados

fields	Valores de campo na ordem correta
--------	-----------------------------------

**Valor de retorno**

-

**Exemplo**

```
Append( "Table" , "Name" , "3" , "26.10.2011" )
```

**Beep**

Envia um tom de sinal padrão do sistema.

**Sintaxe**

```
Beep(frequency: Integer)
```

**Parâmetro**

Parâmetro	Significado
frequency	Frequência do sinal

**Valor de retorno**

-

**Exemplo**

```
Beep(100)
```

**BigInteger**

Converte um valor para um valor BigInteger. O conteúdo da variável deve estar em formato numérico.

**Sintaxe**

```
BigInteger(valueToConvert: Variable): Integer
```

**Parâmetro**

Parâmetro	Significado
valueToConvert	A variável a ser convertida. Definir uma string vazia como valor do parâmetro resultará em uma exceção no tempo de execução!

**Valor de retorno**

Valor BigInteger da variável retornada

**Exemplo**

```
Var = 100
Var = BigInteger(Var)
```

## Bof

Determina se o cursor está no início de um conjunto de dados.

### Sintaxe

Bof(datasetName: String): Boolean

### Parâmetro

Parâmetro	Significado
datasetName	Nome do conjunto de dados

### Valor de retorno

True / False

### Exemplo

```
Beginn = Bof("Table")
```

## CallSystem

Chama um script do mesmo sistema.

### Sintaxe

CallSystem(scriptName: String)

CallSystem(scriptName: String ,inputVariables: String)

CallSystem(scriptName: String ,inputVariables: String ,outputVariables: String)

### Observação sobre inVars/outvars:

Apenas essas variáveis que realmente existem são passadas.

### Parâmetro

Parâmetro	Significado
scriptName	Nome do script a ser executado
inputVariables	Lista separada por vírgulas com os nomes das variáveis a serem passadas (chamadas por valor)
outputVariables	Lista separada por vírgulas com os nomes das variáveis a serem passadas (chamadas por valor). Essas variáveis também são retornadas.

Esses parâmetros devem ser definidos no script de chamada. Consulte também [Parâmetros do script](#)

**Valor de retorno**

outputVariables, se definido.

**Exemplos****Chamado um sem passagem de variável:**

```
CallSystem("callTestScript")
```

**Chamando um script passando inputVariables:**

```
Amount1 = 1
Amount2 = 2
CallSystem("callTestScript", "Amount1,Amount2")
```

Aqui, o script **callTestScript** é chamado do sistema e as variáveis Amount1 e Amount2 estão disponíveis no script chamado.

**Chamando um script passando inputVariables e definindo variáveis de destino:**

```
Amount1 = 1
Amount2 = 2
CallSystem("callTestScript", "Amount1=Number1,Amount2=Number2")
```

Aqui, o script **callTestScript** é chamado do sistema e as variáveis Amount1 e Amount2 estão disponíveis como as variáveis Number1 e Number2 no script chamado.

**Chamando um script passando as variáveis *inVars* e *outVars*:**

```
Amount1 = 1
Amount2 = 2
CallSystem("callTestScript", "Amount1,Amount2", "Sum")
Protocol("Sum of Amount1 and Amount2: {0} ", 0, Sum)
```

Aqui, o script **callTestScript** é chamado do sistema e as variáveis Amount1 e Amount2 estão disponíveis no script chamado.

Após o script ser chamado, a variável Sum do script chamado é passada de volta para o script de chamada e está disponível para processamento adicional.

**Fechar**

Fecha a conexão do banco de dados.

**Sintaxe**

```
Close(datasetName: String)
```

**Parâmetro**

Parâmetro	Significado
datasetName	Nome do conjunto de dados

**Valor de retorno**

-

**Exemplo**

```
Close("DSName")
```

**CloseDatabase**

Fecha a conexão com um banco de dados.

**Sintaxe**

```
CloseDatabase(aliasName: String)
```

**Parâmetro**

Parâmetro	Significado
aliasName	Nome do alias do banco de dados

**Valor de retorno**

-

**Exemplo**

```
CloseDatabase("DBAliasName")
```

**CloseSQL**

Fecha um conjunto de dados aberto.

**Sintaxe**

```
CloseSQL(datasetName: String)
```

**Parâmetro**

Parâmetro	Significado
datasetName	Nome do conjunto de dados

**Valor de retorno**

-

**Exemplo**

```
CloseSQL("DSName")
```

## Confirmar

Executar o comando confirmar armazena permanentemente todas as alterações aos dados feitas na transação atual.

Então, a transação atual é fechada.

A transação atual é a última transação iniciada chamando a função **StartTransaction**.

### Sintaxe

Commit(datasetName: String)

### Parâmetro

Parâmetro	Significado
datasetName	Nome do conjunto de dados

### Valor de retorno

-

### Exemplo

```
Commit("DSName")
```

## Contém

Verifica se a string de pesquisa é encontrada na string de origem.

### Sintaxe

Contains(sourceString: String ,searchString: String): Boolean

### Parâmetro

Parâmetro	Significado
sourceString	A string a ser pesquisada
searchString	A string sendo pesquisada

### Valor de retorno

True / False

### Exemplo

```
source_string = "Perceptive Software"
search_string = "Soft"
ok=Contains(source_string ,search_string)
```

## Contains\_Exactly

Verificação com distinção de maiúsculas e minúsculas se a string de pesquisa foi encontrada na string de origem.

### Sintaxe

Contains\_Exactly(sourceString: String ,searchString: String): Boolean

### Parâmetro

Parâmetro	Significado
sourceString	A string a ser pesquisada
searchString	A string sendo pesquisada

### Valor de retorno

True / False

### Exemplo

```
source_string = "Perceptive Software"
search_string = "ware"
ok=Contains_Exactly(source_string ,search_string)
```

## Contagem

Determina o número de registros em um conjunto de dados.

### Sintaxe

Count(datasetName: String): Integer

### Parâmetro

Parâmetro	Significado
datasetName	Nome do conjunto de dados

### Valor de retorno

Número de conjunto de dados.

### Exemplo

```
num = Count("DSName")
```

## CreatePool

Cria um novo pool de variáveis MonaLisa.

**Sintaxe**

```
CreatePool(poolName: String)
```

**Parâmetro**

Parâmetro	Significado
poolName	Nome do pool

**Valor de retorno**

-

**Exemplo**

```
CreatePool("MyPool")
```

**Decimal**

Converte o valor para um valor de ponto flutuante (decimal).

**Observe:**

Se o parâmetro para Decimal() for uma string ou uma constante, o ponto é sempre o separador decimal.

O separador de milhares é sempre uma vírgula.

Se, por outro lado, uma variável do MonaLisa for passada para Decimal(), o conteúdo é interpretado de acordo com as configurações regionais do Windows para DecimalSeparator e ThousandSeparator.

**Sintaxe**

```
Decimal(valueToConvert: Variable): Decimal
```

**Parâmetro**

Parâmetro	Significado
valueToConvert	O valor a ser convertido. Definir uma string vazia como valor do parâmetro resultará em uma exceção no tempo de execução!

**Valor de retorno**

Valor de ponto flutuante

**Exemplo**

```
b = Decimal(a)
```

**Consulte também:**

[Flutuante](#)



## Excluir

Exclui o registro com o número de registro especificado.

### Sintaxe

```
Delete(datasetName: String ,rowNum: Integer)
```

### Parâmetro

Parâmetro	Significado
datasetName	Nome do conjunto de dados
rowNum	Número de linha do registro a ser excluído.

### Valor de retorno

-

### Exemplo

```
Delete( "DSName" , 1 )
```

## DeletePool

Exclui o pool MonaLisa especificado.

Os seguintes pools não podem ser excluídos:

- SYSTEMPOOL
- SYSTEM
- ZUSATZVARIABLEN (não aplicável na versão em Inglês)
- MODUSER
- BAUSTEIN (não aplicável na versão em Inglês)
- BLOCKPOOL
- DOKUMENT (não aplicável na versão em Inglês)
- DOCUMENTPOOL

### Sintaxe

```
DeletePool(string poolName)
```

### Parâmetro

Parâmetro	Significado
poolName	Nome do pool

### Valor de retorno

-

**Exemplo**

```
DeletePool("MyPool")
```

**Eof**

Determina se o cursor está no final de um conjunto de dados.

**Sintaxe**

Eof(datasetName: String): Boolean

**Parâmetro**

Parâmetro	Significado
datasetName	Nome do conjunto de dados

**Valor de retorno**

True / False

**Exemplo**

```
End = Eof("DSName")
```

**Execute**

Executa um programa.

**Observação para os sistemas operacionais de 64 bits**

O ModusOne é um aplicativo de 32 bits

Se o comando **ExecuteSynchron** for usado para iniciar um programa do diretório

`%windir%\System32` ou

`C:\windows\System32`

o diretório

`%windir%\Sysnative` ou

`c:\windows\Sysnative`

deve ser definido.

Para obter mais informações, use este link:

[http://msdn.microsoft.com/en-us/library/aa384187\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/aa384187(v=vs.85).aspx)

**Sintaxe**

Execute(commandLine: String ,commandShow: Integer): Integer

**Parâmetro**

Parâmetro	Significado
commandLine	Caminho
commandShow	Parâmetro de API do Win32, definindo como a janela deve ser aberta.

**Valor de retorno**

ProgramHandle: integer

**Exemplo**

```
PrgHandle = Execute("C:\Tools\ProgramName.EXE",1)
```

**Encontrar**

Pesquisa o índice atual por um registro que corresponda exatamente aos valores especificados.

**Sintaxe**

Find(datasetName: String ,idxFields: String[]): Boolean

**Parâmetro**

Parâmetro	Significado
datasetName	Nome do conjunto de dados
idxFields	Conteúdo do campo do índice atual.

**Valor de retorno**

Conjunto de dados encontrado True / False

**Exemplo**

```
found = Find("odin", "3899")
```

**FindValue**

Leitura de valores de campos de um conjunto de dados de acordo com um filtro definido.

**Sintaxe**

FindValue(datasetName: String ,filter: String ,formattingRule: String ,resultFields: String[]): String

**Parâmetro**

Parâmetro	Significado
datasetName	Nome do conjunto de dados

filter	<p>Critério de filtro como string separada por vírgula.. Duas alternativas são suportadas:</p> <p><b>Com definição do FILTRO da string:</b></p> <p>Se o filtro contém o FILTRO de string como um primeiro valor, o restante dos valores é interpretado como uma expressão e avaliado corretamente.</p> <p>Um exemplo de um expressão possível é KTO_Par_Num = '0815'.</p> <p><b>Sem definição do FILTRO da string:</b></p> <p>Aqui, os valores do filtro são inseridos em um filterString separado por vírgula.</p> <p>Os nomes da coluna de nomes correspondente são derivados do conjunto de dados.</p> <p>A ordem das colunas no conjunto de dados deve ser observada quando os valores do filtro são definidos.</p>
formattingRule	<p>Regra de formatação, por exemplo, "PartnerNr: %s, AccountType = %s, Balance = %s"</p> <p>Se nenhuma regra de formatação for usada, os valores do resultado das colunas são armazenados sem separadores na string de resultado.</p>
resultFields	Definição dos nomes de colunas cujos valores da string de resultado irá conter.

### Valor de retorno

Valores encontrados.

### Exemplo

```

InitDatabase("DataDbAlias")
InitDataSet("testSet", "DataDbAlias", "dbo.Konten")
;
;Alternativel: with definition of FILTER --> independent of column order
;
StringListAdd("filterString", "FILTER")
StringListAdd("filterString", "KTO_Par_Num = '0815' ") ; Column name with
search value for 1. column of the dataset
StringListAdd("filterString", "AND ")
StringListAdd("filterString", "KTO_Typ = 'N'") ; Column name with
search value for 5. column
filter = String("filterString")
;
; no FormatRule
result = findValue("testSet", filter, "", "KTO_Par_Num", "KTO_Typ",
"KTO_Guthaben")
Protocol(result, 10, "") ; should be : 0815N24,34 --> Value from the first
dataset found

```

```

;
; with FormatRule
formatRule = "PartnerNr: %s, AccountType: %s, Balance: %s"
result = FindValue("testSet", filter, formatRule, "KTO_Par_Num", "KTO_Typ",
"KTO_Guthaben")
Protocol(result, 10, "") ; should be : PartnerNr: 0815, AccountType: N,
Balance: 24,34 --> Values from the first dataset found
;
;Alternative2: no definiton of FILTER --> column order must be adhered to in
filter
;
StringListClear("filterString")
StringListAdd("filterString", "0815") ; Filter value for 1. column of the
dataset
StringListAdd("filterString", "2") ; Filter value for 2. column
StringListAdd("filterString", "55378924") ; Filter value for 3. column
filter = String("filterString")
;
; no FormatRule
result = findValue("testSet", filter, "", "KTO_Par_Num", "KTO_Typ",
"KTO_Guthaben")
Protocol(result, 10, "") ; should be : 0815N24,34 --> Value from the first
dataset found
;
; with FormatRule
formatRule = "PartnerNr: %s, AccountType: %s, Balance: %s"
result = findValue("testSet", filter, formatRule, "KTO_Par_Num", "KTO_Typ",
"KTO_Guthaben")
Protocol(result, 10, "") ; should be : PartnerNr: 0815, AccountType: N,
Balance: 24,34 --> Values from the first dataset found

```

## First

Posiciona o cursor do conjunto de dados no primeiro registro no conjunto de dados.

### Sintaxe

First(datasetName: String)

### Parâmetro

Parâmetro	Significado
datasetName	Nome do conjunto de dados

### Valor de retorno

-

**Exemplo**

```
First("DSName")
```

**Flutuante**

Converte o calor para uma variável de ponto flutuante. O valor usado deve estar em formato numérico.

**Observe:**

Se o parâmetro para Float() for uma string ou uma constante, o ponto é sempre o separador decimal. O separador de milhares é sempre uma vírgula.

Se, por outro lado, uma variável do Monalisa for passada para Float(), o conteúdo é interpretado de acordo com as configurações regionais do Windows para DecimalSeparator e ThousandSeparator.

**Sintaxe**

```
Float(valueToConvert: Variable): Double
```

**Parâmetro**

Parâmetro	Significado
valueToConvert	A variável a ser convertida. Definir uma string vazia como valor do parâmetro resultará em uma exceção no tempo de execução!

**Valor de retorno**

Variável de ponto flutuante

**Exemplo**

```
FloatVar = Float("9.15")           ;FloatVar = 9.15
FloatVar = Float("9.000,15")       ;FloatVar = 9000,15
FloatVar = Float("11,0,15")        ;FloatVar = 110,15
;
Test = "9.15"
FloatVar = Float(Test)              ;result depends on the windows regional
settings, in the United States FloatVar = 915
;
Test = "9,15"
FloatVar = Float(Test)              ;result depends on the windows regional
settings, in the United States FloatVar = 9,15
```

**Formatar**

Permite que você combine cadeias e, ao mesmo tempo, extraia uma subcadeia específica de uma cadeia. Depois, o resultado pode ser formatado para um comprimento específico e preenchido com os caracteres apropriados.

**Sintaxe**

Format(startPosition: Integer ,numberCharacters: Integer ,resultLength: Integer ,formatRule: String ,fillCharacter: String ,formatParams: Variable,...): String

### Parâmetro

Parâmetro	Significado
startPosition	Posição inicial (começando em 1)
numberCharacters	Número de caracteres da posição inicial a ser cortado.
resultLength	Comprimento máximo da cadeia resultante. 0 significa que a cadeia resultante não tem restrição de comprimento.
formatRule	L = alinhamento à esquerda R = alinhamento à direita Z = centralizado
fillCharacter	Caractere de preenchimento (caractere de preenchimento) se a cadeia resultante não tiver o comprimento obrigatório.
formatParams	Cadeias de dados a serem vinculadas juntas

### Valor de retorno

Cadeia formatada

### Exemplo

```
Date = "1.7.2011"
```

```
Day = Format(1, 1, 2, "R", "0",Date) ;      Day    ==> "01"
```

```
Month = Format(3, 1,2, 2, "R", "0",Date) ;      Month   ==> "07"
```

```
Year = Format(5, 4, 4, "R", "0",Date) ;      Year    ==> "2011"
```

```
-----  
Name = "Maximilian Max Mustermann"
```

```
; subst without padding
```

```
Name1 = Format(12, 50, 0, "L", "_", Name); Name1 = "Max Mustermann"
```

```
Name2 = Format( 1, 15, 0, "L", "_", Name); Name2 = "Maximilian Max"
```

```
Name3 = Format(12,  3, 0, "L", "_", Name); Name3 = "Max"
```

```
; subst with padding, text left aligned
```

```
Namela = Format(12, 50, 20, "L", "_", Name); Namela = "Max Mustermann_____"
```

```
; subst with padding, text right aligned
Name1b = Format(12, 50, 20, "R", "_", Name); Name1b = "_____Max Mustermann"
```

```
; subst with padding, text centered
Name1c = Format(12, 50, 20, "Z", "_", Name); Name1c = "___Max Mustermann___"
```

## FormatStr

Formata uma string concatenando as strings variable1 até variableN de acordo com a regra de formatação definida.

Caso o resultado contenha um caractere '%', ele deve ser definido como '%%' no parâmetro formatRule (consulte o exemplo).

### Sintaxe

```
FormatStr(formatRule: String ,formatParams: Variable1, Variable2, VariableN...): String
```

### Parâmetro

Parâmetro	Significado
formatRule	Regra de formatação
formatParams	Strings a serem formatadas

### Valor de retorno

String formatada

### Exemplo

```
d = "01"
m = "02"
dm = FormatStr("Day: %s, Month: %s", d, m)
;Result: dm = "Day: 01, Month: 02"
;
y = "2011"
Date = FormatStr("%s.%s.%s", d, m, y)
;Result Date = "01.02.2011"

;-----
; Percent sign in the result string:
PSatz = "5"
Res = FormatStr("The percentage is %s %% p.a.", PSatz)
; Res = The percentage is 5% p.a.
```

## FreeObject

Libera uma instância de objeto OLE de tempo de execução.



**Sintaxe**

```
FreeObject(varName: String)
```

**Parâmetro**

Parâmetro	Significado
varName	Nome da instância

**Valor de retorno**

-

**Exemplo**

```
FreeObject("Reg")
```

```
GetBlobField
```

**Sintaxe****Parâmetro**

Parâmetro	Significado

**Valor de retorno**

-

**Exemplo**

```
GetConnection
```

Retorna o objeto de conexão do banco de dados em uso no momento.

**Sintaxe**

```
GetConnection(varName: String ,aliasName: String)
```

**Parâmetro**

Parâmetro	Significado
varName	Nome da variável na qual o objeto de conexão do banco de dados está armazenado.

aliasName	Nome do alias do banco de dados
-----------	---------------------------------

**Valor de retorno**

-

**Exemplo**

```
InitDatabase("DataDbAlias")
GetConnection("con", "DataDbAlias")
con_s = con.ConnectionString
```

**GetField**

Lê um valor de campo.

**Sintaxe**

GetField(datasetName: String, fieldName: String): MonalisaVariable

GetField(datasetName: String, fieldName: String, destList: String, dataType: String): MonalisaVariable

**Parâmetro**

Parâmetro	Significado
datasetName	Nome do conjunto de dados
fieldName	Nome do campo
destList	Nome da lista de string de destino para o valor lido, se o tipo de dados for "L".
dataType	DataType S: String I: Número inteiro F: Flutuante L: StringList

**Valor de retorno**

Valor do campo do tipo String, Número inteiro ou Flutuante, dependendo do tipo de dados.

**Exemplo**

```
VN = GetField("DSName", "Forename")
VN = GetField("DSName", "Forename", SListe, "L")
```

**GetLookupValue**

Permite que você leia valores em uma tabela de pesquisa.

**Sintaxe**

GetLookupValue(objectName: String ,key: String): String

**Parâmetro**

Parâmetro	Significado
objectName	Nome da tabela de pesquisa
key	Chave de pesquisa

**Valor de retorno**

Valor de pesquisa

**Exemplo**

```
VarContent="001"
res = GetLookupValue("LookupTab", VarContent)
```

**GetObject**

Cria uma instância do objeto especificado.

**Sintaxe**

GetObject(varName: String ,className: String)

**Parâmetro**

Parâmetro	Significado
varName	Nome da instância
className	Nome da classe / DLL

**Valor de retorno**

-

**Exemplo**

```
GetObject("Reg", "MSLReg_Ini_Lib.MSLRegistry")
```

**GetVariable**

Lê o conteúdo de um campo BLOB.

**Sintaxe**

GetBlobField(datasetName: String ,fieldName: String): Byte[]

**Parâmetro**

Parâmetro	Significado
datasetName	Nome do conjunto de dados
fieldName	Nome do campo

**Valor de retorno**

Conteúdo do campo BLOB

**Exemplo**

```
;Read BLOB field "Data" using the method GetBlobField() and save to file
System.IO.File.WriteAllBytes("D:\Temp\SystemRequirements_modus_Version_5_DB2.
rtf", GetBlobField("dataset", "Data"))
```

**GetSystemOld**

Retorna a SystemOID do sistema no qual o script é executado.

**Sintaxe**

GetSystemOld()

**Parâmetro**

-

**Valor de retorno**

SystemOID

**Exemplo**

```
MyOID = GetSystemOID()
```

**Índice**

Cria um índice no conjunto de dados.

**Sintaxe**

Index(datasetName: String ,idx: String)

**Parâmetro**

Parâmetro	Significado
datasetName	Nome do conjunto de dados
idx	Lista de campos separada por vírgula usada no índice

**Valor de retorno**

-

**Exemplo**

```
Index( "DSName" , "KTO_Konto_Nr ,KTO_BLZ"
```

**InitDataBase**

Cria uma conexão com o banco de dados.

**Sintaxe**

```
InitDatabase(aliasName: String): Boolean
```

**Parâmetro**

Parâmetro	Significado
aliasName	Nome do alias do banco de dados

**Valor de retorno**

Conexão estabelecida True / False

**Exemplo**

```
OK = InitDatabase("odinTest")
```

**InitDataSet**

Cria um novo conjunto de dados para a instância especificada.

**Sintaxe**

```
InitDataset(datasetName: String ,aliasName: String)
```

```
InitDataset(datasetName: String ,aliasName: String ,tableName: String)
```

**Parâmetro**

Parâmetro	Significado
datasetName	Nome do conjunto de dados
aliasName	Nome do alias do banco de dados
tableName	Nome da tabela do banco de dados Se o parâmetro for definido, a tabela é lida (usando 'Selecionar*').

**Valor de retorno**

-

**Exemplo**

```
InitDataSet("modusorder", "ORDERDATA")
InitDataSet("modusorder", "ORDERDATA", "modus_xml_job")
```

**Inserir**

Insere um conjunto de dados e preenche os campos com valores.

**Sintaxe**

```
Insert(datasetName: String ,fields: String[])
```

**Parâmetro**

Parâmetro	Significado
datasetName	Nome do conjunto de dados
fields	Valores de campo na ordem correta

**Valor de retorno**

-

**Exemplo**

```
Insert("DSName", "Perceptive Software", "3", "11,11.2011")
```

**Integer**

Converte uma string em uma variável inteira. O conteúdo da variável deve ser um formato de número.

**Sintaxe**

```
Integer(valueToConvert: Variable): Integer
```

**Parâmetro**

Parâmetro	Significado
valueToConvert	A variável a ser convertida. Definir uma string vazia como valor do parâmetro resultará em uma exceção no tempo de execução!

**Valor de retorno**

Valor inteiro da variável

**Exemplo**

```
Var = "10"
Var = Integer(Var)
```

## Last

Posiciona o cursor do conjunto de dados no último registro do banco de dados.

### Sintaxe

Last(datasetName: String)

### Parâmetro

Parâmetro	Significado
datasetName	Nome do conjunto de dados

### Valor de retorno

-

### Exemplo

```
Last ( "DSName" )
```

## Max

Retorna o maior valor encontrado em uma lista de strings ou lista de variáveis.

### Sintaxe

Max(stringList: String): MonalisaVariable

Max(values: Variable,...): MonalisaVariable

### Parâmetro

Parâmetro	Significado
stringList	Nome da lista de strings
values	Valores a serem verificados

### Valor de retorno

Maior valor

### Exemplo

```
MaxVarL=Max( StrListe )
```

```
v1=10
```

```
v2=30
```

```
v3=50
```

```
MaxVar=Max( v1 , v2 , v3 )
```

## Min

Retorna o menor valor encontrado em uma lista de strings ou lista de variáveis.

### Sintaxe

Min(stringList: String): MonalisaVariable

Min(values: Variable,...): MonalisaVariable

### Parâmetro

Parâmetro	Significado
stringList	Nome da lista de strings
values	Valores a serem verificados

### Valor de retorno

Menor valor

### Exemplo

```

MinVarL=Min(StrListe)
v1=10
v2=30
v3=50
MinVar=Min(v1,v2,v3)

```

## Move

Move o cursor do conjunto de dados em N etapas da posição atual.

### Sintaxe

Move(datasetName: String ,relativeIndex: Integer)

### Parâmetro

Parâmetro	Significado
datasetName	Nome do conjunto de dados
relativeIndex	Posição relativa da posição atual

### Valor de retorno

-

### Exemplo

```
Move("DSName", 3)
```



```
Move ( "DSName" , -3 )
```

### Próximo

Posiciona o cursor do conjunto de dados no próximo registro no conjunto de dados.

### Sintaxe

```
Next(datasetName: String)
```

### Parâmetro

Parâmetro	Significado
datasetName	Nome do conjunto de dados

### Valor de retorno

-

### Exemplo

```
Next ( "DSName" )
```

### OpenSQL

Executa a instrução definida na SetSQL.

### Sintaxe

```
OpenSQL(datasetName: String)
```

### Parâmetro

Parâmetro	Significado
datasetName	Nome do conjunto de dados

### Valor de retorno

-

### Exemplo

```
OpenSQL ( "DSName" )
```

### Analisar

isso permite processar textos com funções especiais, por exemplo, data. Data e hora, calendário ou valores hexadecimais.

### Sintaxe

```
Parse(formattedContent: String): String
```

**Parâmetro**

Parâmetro	Significado
formattedContent	A string a ser analisada

**Valor de retorno**

Texto formatado

**Exemplo****Função especial ^D[mm,dd,yy,yyyy] para inserir a data****Parâmetro**

Parâmetro	Significado
mm	Mês atual
dd	Dia atual
yy	Ano atual (dois algarismos)
yyyy	Ano atual (quatro algarismos)

**Exemplo**

```
Test = Parse( "^Dmm.dd.yyyy;" )
;-->Test = "08,05.2011"
;
Test = Parse( "Current date: ^Dmm.dd.yy;" )
;-->Test = "Current date: 08,05.08 "
```

**Função especial ^U[hh,mm,ss] para inserir data e hora**

Parâmetro	Significado
HH	Hora atual, formato de 24 horas
hh	Hora atual, formato de 12 horas
mm	Minuto atual
ss	Segundo atual

**Exemplo**

```
Test = Parse( "^Uhh:mm:ss;" )
```

```

;-->Test = "10:23:55"
;
Test = Parse("Date: ^Dmm.dd.yy; Time: ^UHH:mm:ss;")
;-->Test = "Date: 04,21.09 Time: 19:19:48"

```

### Função especial ^K para inserir o número de semanas do calendário

#### Exemplo

```

Test = Parse("^K;")
;-->Test = "32"

```

### Função especial ^H[Hexvalue] para inserir valores hexadecimais

#### Parâmetro

Valor hexadecimal

#### Exemplo

```

Test = Parse("^H3F;")
;-->Test = "?"

```

### PrepareSQL

Preenche uma consulta com parâmetros

#### Sintaxe

PrepareSQL(datasetName: String ,paramDescls: String)

#### Parâmetro

Parâmetro	Significado
datasetName	Nome do conjunto de dados
paramDescls	<p>Nome da lista de strings que contém a descrição dos parâmetros SQL.</p> <p>Os parâmetros usados na instrução SQL são descritos nessa lista de strings.</p> <p>As descrições de parâmetros devem ser inseridas na mesma ordem em que estão na instrução SQL.</p> <p>Cada parâmetro deve existir no formato ParameterName=ParameterType.</p> <p>Deve existir uma variável MonaLisa local correspondente com o valor do parâmetro para cada ParameterName definido.</p> <p>Para o parâmetro do tipo <b>b</b> (binário), a variável deve conter o caminho e o nome de arquivo dos dados a serem inserido.</p> <p>O tipo de parâmetro consiste em dois caracteres. O primeiro caractere determina a direção do parâmetro.</p> <p><b>Os seguintes caracteres são suportados:</b></p>

Parâmetro	Significado
b	Bi-direcional (parâmetro de entrada/saída)
i	Parâmetro de entrada
o	Parâmetro de saída
r	Valor de retorno (procedimento armazenado)

Se for definido um caractere sem suporte, o parâmetro 'i' deve ser usado.

O segundo caractere determina o formato do parâmetro.

**Os seguintes caracteres são suportados:**

Parâmetro	Significado
b	Binário (campo BLOB)
d	Data
f	Flutuante
i	Número inteiro
s	String
t	Hora
@	Data/hora

Se for definido um caractere sem suporte, o parâmetro 's' deve ser usado.

### Valor de retorno

-

### Exemplo

```
StringListClear ("ParamDescList")
StringListSetValue("ParamDescList", "MyParam", "is")
SetSQL(myDataSet, SQL)
PrepareSQL(myDataSet, "ParamDescList" )
```

### Anterior

Posiciona o cursor do conjunto de dados no registro anterior no conjunto de dados.

**Sintaxe**

```
Prev(datasetName: String)
```

**Parâmetro**

Parâmetro	Significado
datasetName	Nome do conjunto de dados

**Valor de retorno**

-

**Exemplo**

```
Prev("DSName")
```

**Protocol**

Grava uma entrada no arquivo de rastreamento 'ModusSuite\_Log.txt' no diretório bin.

**Sintaxe**

```
Protocol(traceMessage: String ,traceLevel: Integer ,formatParams: Variable,...)
```

**Parâmetro**

Parâmetro	Significado
traceMessage	Mensagem de rastreamento
traceLevel	<b>Nível de rastreamento</b> 0 = Erro 1-4 = Aviso 5-9 = Informação 10 = Detalhamento
formatParams	Parâmetros definidos para cada espaço reservado (espaços reservados são {0},{1} etc.)

**Valor de retorno**

-

**Exemplo**

```
ErrorNumber = "001"  
Protocol("Error message {0} ", 0, ErrorNumber)
```

## PutField

Insere um valor em um campo.

### Sintaxe

```
PutField(datasetName: String ,fieldName: String ,fieldValue: Variable)
```

### Parâmetro

Parâmetro	Significado
datasetName	Nome do conjunto de dados
fieldName	Nome do campo
fieldValue	Novo conteúdo do campo

### Valor de retorno

-

### Exemplo

```
PutField("DSName", "CompanyName", "Perceptive Software")
```

## Raise

Aciona um erro e ignora o rótulo de erro definido em OnError.

### Sintaxe

```
Raise(errorTypeToken: String ,errorMessage: String)
```

### Parâmetro

Parâmetro	Significado
errorTypeToken	O valor passado à variável LastError quando a função Raise é executada.
errorMessage	Mensagem de erro detalhada

### Valor de retorno

-

### Exemplo

```
Raise("ERR1", "Error 1 thrown")
```

## Leitura

Lê um conjunto de dados completo e retorna o resultado.

**Sintaxe**

Leitura (datasetName: String,fieldList: String,destList: String ,delimiter: String ,quote: String): String

**Parâmetro**

Parâmetro	Significado
datasetName	Nome do conjunto de dados
fieldList	O nome da lista de string contendo os nomes das colunas a serem lidas. Se a lista de string estiver vazia, todos os campos são lidos. A lista de string deve existir.
destList	O nome da lista, o resultado será armazenado dentro. Se a lista de string especificada não for encontrada, ela é criada. Se uma string vazia for passada, apenas a string de resultados é retornada.
delimiter	Contém o separador para os conteúdos de campo. Uma vírgula é usada por uma string vazia.
quote	Define o caractere que inclui o valor de resultado. Se uma string vazia for definida, nenhum dos valores de resultado individual é incluído.

**Valor de retorno**

Campos de dados lidos em uma string.

**Exemplo**

```

InitDatabase("DataDbAlias")
InitDataSet("testSet", "DataDbAlias", "dbo.Konten")
;
; fieldList Alternative 1: fieldList = Empty List --> all column values are
in the result
StringListClear("fieldList") ; clears the string list or creates a new one if
not found
;
dummy_s = Read("testSet", "fieldList", "", ",", "'') ;no destList defined -->
only result string
Protocol("fieldList Alternative 1: fieldList = Empty List; Set1",10,"")
Protocol(dummy_s, 10, "'') ;should be :
'0815', '1', '69924420', '60050000', 'H', 'Hauptkonto', '1,23'
;
Next("testSet")
;

```

```

dummy_s = Read("testSet", "fieldList", "", "", "", "") ;no destList defined -->
only result string
Protocol("fieldList Alternative 1: fieldList = Empty List; Set2",10,"")
Protocol(dummy_s, 10, "") ;should be :
'0815','2','55378924','62045000','N','Nebenkonto','24,34'
;
First("testSet") ;move to first record of dataset
;
;fieldList Alternative 2: fieldList contains the columns wanted in the result
StringListAdd("fieldList", "KTO_Par_Num")
StringListAdd("fieldList", "KTO_Typ")
StringListAdd("fieldList", "KTO_Guthaben")
dummy_s = Read("testSet", "fieldList", "", "", "", "") ;no destList defined -->
only result string
Protocol("fieldList Alternative 2: fieldList contains the columns wanted in
the result; Set1",20,"")
Protocol(dummy_s, 10, "") ;should be : '0815','H','1,23'
;
Next("testSet")
;
dummy_s = Read("testSet", "fieldList", "", "", "", "") ;no destList defined -->
only result string
Protocol("fieldList Alternative 2: fieldList contains the columns wanted in
the result; Set2",20,"")
Protocol(dummy_s, 10, "") ;should be : '0815','N','24,34'
;
First("testSet") ;move to first record of dataset
;
;destList Alternative 1: only destList-Name defined --> List must be created
automatically
dummy_s = Read("testSet", "fieldList", "destList", "", "", "")
Protocol("destList Alternative 1: only destList-Name defined;
Returnstring",10,"")
Protocol(dummy_s, 10, "") ;should be : '0815','H','1,23'
Protocol("destList Alternative 1: only destList-Name defined; StringList
1.Zeile",10,"")
Protocol(StringListLine("destList", 0), 10, "")
;
Next("testSet")
;destList Alternative 2: destList already exists with entry --> List must be
automatically appended
dummy_s = Read("testSet", "fieldList", "destList", "", "", "")
destListString = String("destList")
Protocol("destList Alternative 2: extend existing destList;
Returnstring",10,"")
Protocol(dummy_s, 10, "") ;should be : '0815','N','24,34'

```



```

Protocol("destList Alternative 2: extend existing destList; StringList
1.Row",10,"")
Protocol(StringListLine("destList", 0), 10, "")
Protocol("destList Alternative 2: extend existing destList; StringList
2.Row",10,"")
Protocol(StringListLine("destList", 1), 10, "")

```

## RegisterLib

Registra uma biblioteca de funções (DLL) do sistema.

### Sintaxe

RegisterLib(dllFileName: String)

### Parâmetro

Parâmetro	Significado
dllFileName	Nome e caminho da DLL

### Valor de retorno

-

### Exemplo

```
RegisterLib("C:\Winnt\System\MSLDB_LIB.DLL")
```

## ReRaise

Pode ser usado dentro de um bloco OnError para passar a exceção de volta ao contexto que o chamou.

### Sintaxe

ReRaise()

### Parâmetro

-

### Valor de retorno

-

### Exemplo

```
ReRaise()
```

## Rollback

Descarta todas as alterações, insere e exclui ações da transação atual e fecha a transação.

### Sintaxe

Rollback(datasetName: String)

**Parâmetro**

Parâmetro	Significado
datasetName	Nome do conjunto de dados

**Valor de retorno**

-

**Exemplo**

```
Rollback("DSName")
```

**Round**

Arredonda um número de ponto flutuante para um valor inteiro.

**Sintaxe**

```
Round(value: Double): Integer
```

**Parâmetro**

Parâmetro	Significado
value	O número de ponto flutuante a ser arredondado.

**Valor de retorno**

Valor arredondado

**Exemplo**

```
v1=10.45
VarInt=Round(v1)
```

**Search**

Pesquisa por um conjunto de dados que corresponda aos critérios definidos no parâmetro.

**Sintaxe**

```
Search(datasetName: String ,expression: String): Boolean
```

**Parâmetro**

Parâmetro	Significado
datasetName	Nome do conjunto de dados
expression	Conteúdo do campo do índice atual.

**Valor de retorno**

True / False

**Exemplo**

```
found = Search("odin","ODJ_ID=3899")
```

**SelFirstRecord**

Posiciona o cursor no primeiro registro do objeto de dados.

**Sintaxe**

SelFirstRecord(nodeName: Variable): Boolean

**Parâmetro**

Parâmetro	Significado
nodeName	Nó no objeto de dados, por exemplo VSH.RI.ADR

**Valor de retorno**

-

**Exemplo**

```
SelFirstRecord("VSH.RI.ADR")
```

**SelNextRecord**

Posiciona o cursor um registro além em um objeto de dados.

**Sintaxe**

SelNextRecord(nodeName: Variable): Boolean

**Parâmetro**

Parâmetro	Significado
nodeName	Nó no objeto de dados, por exemplo VSH.RI.ADR

**Valor de retorno**

-

**Exemplo**

```
SelNextRecord("VSH.RI.ADR")
```

**SelRecordCount**

Retorna o número de registros em um objeto de dados.

**Sintaxe**

`SelRecordCount(nodeName: Variable): Integer`

**Parâmetro**

Parâmetro	Significado
nodeName	Nó no objeto de dados, por exemplo VSH.RI.ADR

**Valor de retorno**

Número de conjunto de dados

**Exemplo**

```
NumberOfRecords=SelRecordCount ( "PARTNER.ACCOUNTS" )
```

**SelRecordNumber**

Retorna o número de registro atual de um objeto de dados.

**Sintaxe**

`SelRecordNumber(nodeName: Variable): Integer`

**Parâmetro**

Parâmetro	Significado
nodeName	Nó no objeto de dados, por exemplo VSH.RI.ADR

**Valor de retorno**

Número de registro do objeto de dados

**Exemplo**

```
RecordNumber=SelRecordNumber ( "PARTNER.KONTEN" )
```

**SelVariable**

Determina o valor da variável em um objeto de dados da seleção atual.

**Sintaxe**

`SelVariable(dataObject: String ,dataAttribute: String): MonalisaVariable`

`SelVariable(dataObject: String ,dataAttribute: String ,defaultValue: String): MonalisaVariable`

**Parâmetro**

Parâmetro	Significado
-----------	-------------

dataObject	Caminho completo do objeto de dados
dataAttribute	Nome do atributo
defaultValue	Valor de retorno padrão se o objeto de dados ou o atributo não forem encontrados.

### Valor de retorno

Valor da variável como uma string.

### Exemplo

```
STR = SelVariable("ADRESSE","STREET,"")
```

Se o objeto de dados tiver um objeto de dados pai, como no exemplo a seguir, o caminho completo deve ser especificado:

```
ExRate = SelVariable("PARTNER.CONTRACT.CURRENCY_CODE", "CUR_EXCHANGE_RATE",
"")
```

### SelVariableEx

Determina o valor da variável em um objeto de dados da seleção atual.

### Sintaxe

SelVariableEx(dataObject: String ,dataAttribute: String): MonalisaVariable

SelVariableEx(dataObject: String ,dataAttribute: String ,defaultValue: String): MonalisaVariable

### Parâmetro

Parâmetro	Significado
dataObject	Caminho completo do objeto de dados
dataAttribute	Nome do atributo
defaultValue	Valor de retorno padrão se o objeto de dados ou o atributo não forem encontrados.

### Valor de retorno

Para determinar o tipo dos dados de retorno, as seguintes verificações são realizadas. A primeira verificação aplicada determina o tipo.

Se o valor lido for um número inteiro válido, o resultado é uma variável Monalisa do tipo Inteiro.

Se o valor lido for um número de ponto flutuante válido, o resultado é uma variável Monalisa do tipo Flutuante.

Em outros casos, o resultado é uma variável Monalisa do tipo String.

### Exemplo

```
STR = SelVariableEx("ADDRESS","STREET,"")
```

Se o objeto de dados tiver um objeto de dados pai, como no exemplo a seguir, o caminho completo deve ser especificado:



```
ExRate = SelVariableEx("PARTNER.CONTRACT.CURRENCY_CODE", "CUR_EXCHANGE_RATE",
" ")
```

## SetExecSQL

[▶ Mostrar tudo](#)

Executa um comando SQL sem um resultado.

### Sintaxe

```
SetExecSQL(datasetName: String ,sqlList: String)
```

```
SetExecSQL(datasetName: String ,sqlList: String ,paramDescs: String)
```

### Parâmetro

Parâmetro	Significado				
datasetName	Nome do conjunto de dados				
sqlList	Nome da lista de cadeias de caracteres que contém a instrução SQL.				
paramDescs	<p>Nome da lista de cadeias de caracteres que contém a descrição dos parâmetros SQL.</p> <p>Os parâmetros usados na instrução SQL são descritos nessa lista de cadeias de caracteres.</p> <p>As descrições de parâmetros devem ser inseridas na mesma ordem em que estão na instrução SQL.</p> <p>Cada parâmetro deve existir no formato NomeDoParâmetro=TipoDoParâmetro.</p> <p>Deve existir uma variável MonaLisa local correspondente com o valor do parâmetro para cada NomeDeParâmetro definido.</p> <p>Para o parâmetro do tipo <b>b</b> (binário), a variável deve conter o caminho e o nome de arquivo dos dados a serem inserido.</p> <p>O tipo de parâmetro consiste em dois caracteres. O primeiro caractere determina a direção do parâmetro.</p> <p><b>Os seguintes caracteres são suportados:</b></p> <table border="1"> <thead> <tr> <th>Parâmetro</th> <th>Significado</th> </tr> </thead> <tbody> <tr> <td>b</td> <td>Bi-direcional (parâmetro de entrada/saída)</td> </tr> </tbody> </table>	Parâmetro	Significado	b	Bi-direcional (parâmetro de entrada/saída)
Parâmetro	Significado				
b	Bi-direcional (parâmetro de entrada/saída)				

i	Parâmetro de entrada
o	Parâmetro de saída
r	Valor de retorno (procedimento armazenado)

Se for definido um caractere sem suporte, o parâmetro 'i' deve ser usado.

O segundo caractere determina o formato do parâmetro.

**Os seguintes caracteres são suportados:**

Parâmetro	Significado
b	Binário (campo BLOB)
d	Data
f	Flutuante
i	Inteiro
s	Cadeia de caracteres
t	Hora
@	Data/hora

Se for definido um caractere sem suporte, o parâmetro 's' deve ser usado.

Se a instrução SQL contiver parâmetros (%s ou :Param), ele só será executada se esses parâmetros puderem ser resolvidos ou reconhecidos.

Isso é feito através da lista paramsDescls.

Se a lista já tiver sido passada com SetExecSql, o comando é executado.

Se não, os parâmetros devem ser resolvidos ou disponibilizados chamando PrepareSQL e passando a lista.

Em seguida, a função ExecSql deve ser executada.

### Valor de retorno

-

### Exemplo

```
InitDatabase("DataDbAlias")
InitDataSet("testSet", "DataDbAlias")
#region Testpreparation
```

```

;first delete any existing test datasets
StringListClear("SqlDefList")
StringListAdd("SqlDefList", "Delete from dbo.Accounts")
StringListAdd("SqlDefList", "where (ACC_Par_Num = '6666' ) ")
StringListAdd("SqlDefList", "OR (ACC_Par_Num = '9999' ) ")
SetExecSQL("testSet", "SqlDefList")
;
#region create Testddataset1
StringListClear("SqlDefList")
StringListAdd("SqlDefList", "INSERT INTO dbo.Accounts")
StringListAdd("SqlDefList", "(ACC_Par_Num, ")
StringListAdd("SqlDefList", "ACC_Seq_Num, ")
StringListAdd("SqlDefList", "ACC_Account number, ")
StringListAdd("SqlDefList", "ACC_POSTCODE, ")
StringListAdd("SqlDefList", "ACC_Type, ")
StringListAdd("SqlDefList", "ACC_Name, ")
StringListAdd("SqlDefList", "Acc_Credit)")
StringListAdd("SqlDefList", "VALUES")
StringListAdd("SqlDefList", "(6666, ")
StringListAdd("SqlDefList", "1, ")
StringListAdd("SqlDefList", "066066066, ")
StringListAdd("SqlDefList", "62093210, ")
StringListAdd("SqlDefList", "'H', ")
StringListAdd("SqlDefList", "'MyAccount', ")
StringListAdd("SqlDefList", "66.66)")
SetExecSQL("testSet", "SqlDefList")
#endregion
;
#region create Testddataset2
StringListClear("SqlDefList")
StringListAdd("SqlDefList", "INSERT INTO dbo.Accounts")
StringListAdd("SqlDefList", "(ACC_Par_Num, ")
StringListAdd("SqlDefList", "ACC_Seq_Num, ")
StringListAdd("SqlDefList", "ACC_Account number, ")
StringListAdd("SqlDefList", "ACC_POSTCODE, ")
StringListAdd("SqlDefList", "ACC_Type, ")
StringListAdd("SqlDefList", "ACC_Name, ")
StringListAdd("SqlDefList", "Acc_Credit)")
StringListAdd("SqlDefList", "VALUES")
StringListAdd("SqlDefList", "(9999, ")
StringListAdd("SqlDefList", "1, ")
StringListAdd("SqlDefList", "099099099, ")
StringListAdd("SqlDefList", "62999999, ")
StringListAdd("SqlDefList", "'H', ")
StringListAdd("SqlDefList", "'MyAccount2', ")
StringListAdd("SqlDefList", "99.99)")

```



```

SetExecSQL("testSet", "SqlDefList")
#endregion
#endregion
;
;-----

;-----
StringListClear("SqlDefList")
StringListAdd("SqlDefList", "Delete from dbo.Accounts")
StringListAdd("SqlDefList", "where (ACC_Par_Num = %s) ")
StringListAdd("SqlDefList", "AND (ACC_Type = :MyTyp) ")
;Store parameter descriptions in a string list
StringListClear("ParamDescList")
StringListSetValue("ParamDescList", "MyVar", "is") ; Alternative to StringListAdd("ParamDescList", "MyVar=is")
StringListSetValue("ParamDescList", "MyTyp", "") ; Empty string as Value: by default takes 'is'
;
;-----
-----
;Alternative1: Call SetExecSQL passing the paramsDescList: Delete test dataset1
;-----
-----
MyVar = "6666"
MyTyp = "H"
SetExecSQL("testSet", "SqlDefList", "ParamDescList")
;Store SqlStatement in a string list
StringListClear("SqlDefList")
StringListAdd("SqlDefList", "Delete from dbo.Accounts")
StringListAdd("SqlDefList", "where (ACC_Type = :MyTyp) ")
StringListAdd("SqlDefList", "AND (ACC_Par_Num = %s) ")
;
;Store parameter descriptions in a string list
StringListClear("ParamDescList")
StringListSetValue("ParamDescList", "MyTyp", "") ; Order of parameters must be identical to the order
StringListSetValue("ParamDescList", "MyVar", "is") ; in the SQL statement!
;-----
-----
;Alternative2: Alternative ExecSQL call
For this, SetExecSQL without passing the paramsList (or SetSql) and PrepareSQL calls must already have been completed:
Delete test dataset2
;-----
-----
MyVar="9999"

```

```
SetExecSQL("testSet", "SqlDefList")
PrepareSQL("testSet", "ParamDescList")
ExecSQL("testSet")
```

### Exemplo: Inserir arquivos RTF em um campo BLOB

```
InitDatabase("Odin")
InitDataSet("dataSet", "Odin")

StringListClear("SqlDefList")
StringListAdd("SqlDefList", "Insert into dbo.Data (AutoIndex, Datatype, Data)
  Values (:AutoIndex, :Datatype, :Data)")

StringListClear("ParamDescList")
StringListSetValue("ParamDescList", "AutoIndex", "ii") ; integer
StringListSetValue("ParamDescList", "Datatype", "ii") ; integer
StringListSetValue("ParamDescList", "Data", "ib") ; binary

AutoIndex = 4711
Datatype = 1
Data = "D:\Temp\SystemRequirements_modus_Version_5.rtf"
SetExecSQL("dataSet", "SqlDefList", "ParamDescList")
```

### Exemplo: Ler o campo BLOB e salvar em arquivo (lista de strings)

```
InitDatabase("Odin")
InitDataSet("dataset", "Odin")

StringListClear("SqlDefList")
StringListAdd("SqlDefList", "Select Data from Data where Autoindex = 4711")
SetSQL("dataset", "SqlDefList")

;read BLOB field "Data" and save to file
GetField("dataset", "Data", "D:\Temp\SystemRequirements_modus_Version_5_DB.rtf", "")

; read blob-field "Data" into stringlist "test"
GetField("dataset", "Data", "test", "L")

;read BLOB field "Data" using the method GetBlobField() and save to file
System.IO.File.WriteAllBytes("D:\Temp\SystemRequirements_modus_Version_5_DB2.rtf", GetBlobField("dataset", "Data"))
```

### SetPoolVariable

Define o valor de uma variável de pool.

Se a variável não existir, ela será criada.

Se a variável já existir, ela será substituída.

**Sintaxe**

```
SetPoolVariable(string poolName, string varName, string Value)
```

**Parâmetro**

Parâmetro	Significado
poolName	Nome do pool
varName	Nome da variável
Value	O valor a ser definido

**Valor de retorno**

-

**Exemplo**

```
VarName = "Var_"
i = 10
while (i < 10)
    PoolVarName = VarName + i
    SetPoolVariable("System", PoolVarName, i)
    i = i + 1
end-while
```

**SetSQL**

Executa um comando SQL.

**Sintaxe**

```
SetSQL(datasetName: String ,sqlList: String)
```

**Parâmetro**

Parâmetro	Significado
datasetName	Nome do conjunto de dados
sqlList	Nome da lista de strings que contêm a definição SQL.

**Valor de retorno**

-

**Exemplo**

```
StringlistAdd("SQL","Delete from dfd.db")
```

```
InitDataset("Table","AllSQL")
SetSQL("Table", "SQL")
```

## Sleep

Pausa a execução do script pelo período especificado.

### Sintaxe

```
Sleep(sleepTime: Integer)
```

### Parâmetro

Parâmetro	Significado
sleepTime	Pausa em milissegundos

### Valor de retorno

-

### Exemplo

```
Sleep(1000)
```

## SplitStr

Obtém uma expressão parcial específica a partir de uma string.

### Sintaxe

```
SplitStr(data: String ,splitedStr: Integer ,seperator: String ,errorText: String): String
```

### Parâmetro

Parâmetro	Significado
data	String
splitedStr	Número da expressão parcial começando por 1
seperator	Delimitador para expressões parciais individuais
errorText	String de resultado retornada quando uma expressão parcial não é encontrada.

### Valor de retorno

Expressão parcial da string.

### Exemplo

```
dbvar = "T06M07J08"
s1 = SplitStr(dbvar, 1, "TMJX", "")
```

```

;S1 = "06"
;
S2 = SplitStr(dbvar, 2, "TMJX", "")
;S2 = "07"
;
S3 = SplitStr(dbvar, 3, "TMJX", "")
;S3 = "08"
;
S4 = SplitStr(dbvar, 4, "TMJX", "-")
;S4 = "-"

```

### StartTransaction

Inicia uma nova transação.

#### Sintaxe

StartTransaction(datasetName: String)

#### Parâmetro

Parâmetro	Significado
datasetName	Nome do conjunto de dados

#### Valor de retorno

-

#### Exemplo

```
StartTransaction("DSName")
```

### StoredProcedure

Executa um procedimento armazenado.

#### Sintaxe

StoredProcedure(aliasName: String ,procName: String)

StoredProcedure(aliasName: String ,procName: String ,paramDescls: String)

#### Parâmetro

Parâmetro	Significado
aliasName	Nome do alias do banco de dados
procName	Nome do procedimento armazenado
paramDescls	Parâmetro do procedimento armazenado na ordem correta

Consulte: [SetExecSQL](#)

### Valor de retorno

-

### Exemplo

```
StoredProcedure( "DSName", "ProcedureName", "Hans", "Mustermann" )
```

### StrDelChar

Exclui um caractere específico de uma string.

### Sintaxe

```
StrDelChar(inString: String ,delChar: String): String
```

### Parâmetro

Parâmetro	Significado
inString	String ou variável
delChar	Caractere a ser excluído

### Valor de retorno

String de resultado

### Exemplo

```
var1="a b c d"
var2=StrDelChar(var1, " ")
;var2=abcd
```

### String

Converte uma variável para o formato de string.

Essa função pode ser usada para reverter uma lista de strings de volta para uma string.

Os elementos da lista são separados por [CR+LF].

### Sintaxe

```
String(valueToConvert: Variable): String
```

### Parâmetro

Parâmetro	Significado
valueToConvert	A variável a ser convertida (alfanumérica, numérica ou uma lista de strings)

**Valor de retorno**

String convertida

**Exemplo**

```
i=5
s=String(i)
;
StringlistAdd("StrListe","Perceptive Software","Konrad-Zuse-Str. 9","74172
Neckarsulm")
Address=String("StrListe")
```

**StringListAdd**

Adiciona strings a uma lista de strings.

**Sintaxe**

StringListAdd(stringList: String ,stringParameters: Variable,...)

**Parâmetro**

Parâmetro	Significado
stringList	Nome da lista de strings
stringParameters	Strings a serem adicionadas

**Valor de retorno**

-

**Exemplo**

```
Street = "Konrad-Zuse-Str. 9"
Code = "74172"
City = "Neckarsulm"
;
StringlistAdd(SListe, Street, Code, City)
Address = String(SListe)
;-->Address = "Konrad-Zuse-Str. 9,74172,Neckarsulm"
```

**StringListClear**

Remove todos os elementos de uma lista de strings.

Se a lista de strings especificada não for encontrada, ela será criada.

**Sintaxe**

StringListClear(stringList: String)

**Parâmetro**

Parâmetro	Significado
stringList	Nome da lista de strings

**Valor de retorno**

-

**Exemplo**

```
StringListClear(SListe)
```

**StringListCount**

Conta as linhas em uma lista de strings.

**Sintaxe**

```
StringListCount(stringList: String): Integer
```

**Parâmetro**

Parâmetro	Significado
stringList	Nome da lista de strings

**Valor de retorno**

Número de linhas

**Exemplo**

```
i = StringListCount(SListe)
```

**StringListGetCommaText**

Retorna todas as linhas de uma lista de strings como uma string separada por vírgulas.

**Sintaxe**

```
StringListGetCommaText(stringList: String): String
```

**Parâmetro**

Parâmetro	Significado
stringList	Nome da lista de strings

**Valor de retorno**

Conteúdo da lista de strings



**Exemplo**

```
Commatext = StringListGetCommatext (SListe)
```

**StringListGetText**

Retorna o texto de uma lista de strings. Cada linha termina com CRLF.

**Sintaxe**

```
StringListGetText(stringList: String): String
```

**Parâmetro**

Parâmetro	Significado
stringList	Nome da lista de strings

**Valor de retorno**

Conteúdo da lista de strings

**Exemplo**

```
SLText = StringListGetText(SListe)
```

**StringListGetValue**

Lê o valor de um nome dentro de uma lista de strings no formato: "Name=Value".

**Sintaxe**

```
StringListGetValue(stringList: String ,aName: String): String
```

**Parâmetro**

Parâmetro	Significado
stringList	Nome da lista de strings
aName	Identificador do valor a ser lido.

**Valor de retorno**

O valor atribuído ao identificador

**Exemplo**

```
SLWert = StringListGetValue(SListe,Name)
```

**StringListIndexOf**

Retorna a posição do primeiro elemento na linha da lista de strings que contém o valor especificado.

**Sintaxe**

StringListIndexOf(stringList: String ,aLine: String): Integer

### Parâmetro

Parâmetro	Significado
stringList	Nome da lista de strings
aLine	Valor de pesquisa

### Valor de retorno

Posição do elemento

### Exemplo

```
StringlistAdd("Title", "Mr.", "Mrs.", "Company", "Family")
Pos = StringListIndexOf ("Title", "Company")
;-->Pos=2
```

### StringListItemValues

Lê o valor após o índice especificado e grava o texto separado por vírgula na lista de strings de destino.

### Sintaxe

StringListItemValues(stringList: String ,lineIndex: Integer ,destList: String): String

### Parâmetro

Parâmetro	Significado
stringList	Nome da lista de strings
lineIndex	Índice de linha da lista de strings começando por 0
destList	Nome da lista de strings destino

### Valor de retorno

Nome do valor

### Exemplo

```
StringlistAdd("test", "Valuelist=1,2,3,4,5")
name = StringListItemValues("test", 0, "Values")
;--> Name = "Valuelist"
;
value = StringListLine("Values", 0)
;--> value = "1 "
;
```

```
value = StringListLine("Values",1)
;--> value = "2 "
```

### StringListLine

Lê o conteúdo de uma linha em uma lista de strings.

#### Sintaxe

```
StringListLine(stringList: String ,lineIndex: Integer): String
```

#### Parâmetro

Parâmetro	Significado
stringList	Nome da lista de strings
lineIndex	Número da linha a ser lida. As linhas iniciam em 0.

#### Valor de retorno

Conteúdo da linha

#### Exemplo

```
line = StringListLine("SListe",0)
```

### StringListLoad

Carrega uma lista de strings a partir de um arquivo.

#### Sintaxe

```
StringListLoad(stringList: String ,fileName: String)
```

```
StringListLoad(stringList: String ,fileName: String ,sectionName: String)
```

#### Parâmetro

Parâmetro	Significado
stringList	Nome da lista de strings
fileName	Nome do arquivo a partir do qual a lista será carregada.
sectionName	Nome da seção de onde os dados devem ser carregados (parâmetro opcional)

#### Valor de retorno

-

#### Exemplo

```
StringlistLoad("SListe", "c:\Temp\Test.lst")
StringlistLoad("SListe", "c:\Temp\Test.lst", "Section1")
```

## StringListParams

Configura os parâmetros da lista de strings.

### Sintaxe

```
StringListParams(stringList: String ,sortList: Boolean ,duplicateState: Integer)
```

### Parâmetro

Parâmetro	Significado
stringList	Nome da lista de strings
sortList	Especifica se a lista de strings será classificada.
duplicateState	0 = Ignorar - ignorar duplicatas 1 = Aceitar - aceitar duplicatas 2 = Erro - mensagem de erro ao inserir duplicatas. A definição de parâmetros para duplicatas não tem qualquer efeito sobre as strings duplas já incluídas na lista.

### Valor de retorno

-

### Exemplo

```
StringlistAdd("SListe", "Mr.", "Mrs.", "Company", "Family")
StringListParams("SListe", True, 2)
StringlistAdd("SListe", "Company")
```

## StringListSave

[▶ Mostrar tudo](#)

Salva uma lista de strings em um arquivo.

### Sintaxe

```
StringListSave(string stringList, string fileName)
StringListSave(string stringList, string fileName, string encoding)
```

### Parâmetro

Parâmetro	Significado
stringList	Nome da lista de strings

fileName	Nome do arquivo no qual a lista será salva.																																								
encoding	<p>Os seguintes podem ser usados como parâmetro de codificação:</p> <ul style="list-style-type: none"> <li>• String vazia. Nesse caso, a codificação padrão (página de código ANSI do sistema operacional) é usada.</li> <li>• página de código</li> <li>• nome</li> <li>• nome amigável</li> </ul> <p>Exemplo:</p> <ul style="list-style-type: none"> <li>• "1252"</li> <li>• "windows-1252"</li> <li>• "Europeu Ocidental" (Windows)</li> </ul> <p>Parâmetros de codificação possíveis</p> <table border="1"> <thead> <tr> <th>Página de código</th> <th>Nome</th> <th>Nome de exibição</th> </tr> </thead> <tbody> <tr> <td>37</td> <td>IBM037</td> <td>IBM EBCDIC (EUA-Canadá)</td> </tr> <tr> <td>437</td> <td>IBM437</td> <td>OEM EUA</td> </tr> <tr> <td>500</td> <td>IBM500</td> <td>IBM EBCDIC (Internacional)</td> </tr> <tr> <td>708</td> <td>ASMO-708</td> <td>Árabe (ASMO 708)</td> </tr> <tr> <td>720</td> <td>DOS-720</td> <td>Árabe (DOS)</td> </tr> <tr> <td>737</td> <td>ibm737</td> <td>Grego (DOS)</td> </tr> <tr> <td>775</td> <td>ibm775</td> <td>Báltico (DOS)</td> </tr> <tr> <td>850</td> <td>ibm850</td> <td>Europeu Ocidental (DOS)</td> </tr> <tr> <td>852</td> <td>ibm852</td> <td>Europeu Oriental (DOS)</td> </tr> <tr> <td>855</td> <td>IBM855</td> <td>OEM Cirílico</td> </tr> <tr> <td>857</td> <td>ibm857</td> <td>Turco (DOS)</td> </tr> <tr> <td>858</td> <td>IBM00858</td> <td>OEM Multilíngue Latino I</td> </tr> </tbody> </table>		Página de código	Nome	Nome de exibição	37	IBM037	IBM EBCDIC (EUA-Canadá)	437	IBM437	OEM EUA	500	IBM500	IBM EBCDIC (Internacional)	708	ASMO-708	Árabe (ASMO 708)	720	DOS-720	Árabe (DOS)	737	ibm737	Grego (DOS)	775	ibm775	Báltico (DOS)	850	ibm850	Europeu Ocidental (DOS)	852	ibm852	Europeu Oriental (DOS)	855	IBM855	OEM Cirílico	857	ibm857	Turco (DOS)	858	IBM00858	OEM Multilíngue Latino I
Página de código	Nome	Nome de exibição																																							
37	IBM037	IBM EBCDIC (EUA-Canadá)																																							
437	IBM437	OEM EUA																																							
500	IBM500	IBM EBCDIC (Internacional)																																							
708	ASMO-708	Árabe (ASMO 708)																																							
720	DOS-720	Árabe (DOS)																																							
737	ibm737	Grego (DOS)																																							
775	ibm775	Báltico (DOS)																																							
850	ibm850	Europeu Ocidental (DOS)																																							
852	ibm852	Europeu Oriental (DOS)																																							
855	IBM855	OEM Cirílico																																							
857	ibm857	Turco (DOS)																																							
858	IBM00858	OEM Multilíngue Latino I																																							

860	IBM860	Português (DOS)
861	ibm861	Islandês (DOS)
862	DOS-862	Hebraico (DOS)
863	IBM863	Francês do Canadá (DOS)
864	IBM864	Árabe (864)
865	IBM865	Nórdico (DOS)
866	cp866	Cirílico (DOS)
869	ibm869	Grego moderno (DOS)
870	IBM870	IBM EBCDIC (Multilíngue Latino 2)
874	windows-874	Tailandês (Windows)
875	cp875	IBM EBCDIC (Grego moderno)
932	shift_jis	Japonês (Shift-JIS)
936	gb2312	Chinês simplificado (GB2312)
949	ks_c_5601-1987	Coreano
950	big5	Chinês (tradicional) (Big5)
1026	IBM1026	IBM EBCDIC (Turco, Latino 5)
1047	IBM01047	IBM Latino 1
1140	IBM01140	IBM EBCDIC (EUA-Canadá-Europeu)
1141	IBM01141	IBM EBCDIC (Alemanha-Europeu)

1142	IBM01142	IBM EBCDIC (Dinamarca-Noruega-Europeu)
1143	IBM01143	IBM EBCDIC (Finlândia-Suécia-Europeu)
1144	IBM01144	IBM EBCDIC (Italiano-Europeu)
1145	IBM01145	IBM EBCDIC (Espanha-Europeu)
1146	IBM01146	IBM EBCDIC (Grã-Bretanha-Europeu)
1147	IBM01147	IBM EBCDIC (França-Europeu)
1148	IBM01148	IBM EBCDIC (Internacional-Europeu)
1149	IBM01149	IBM EBCDIC (Islandês-Europeu)
1200	utf-16	Unicode
1201	unicodeFFFE	Unicode (Big Endian)
1250	windows-1250	Centro-europeu (Windows)
1251	windows-1251	Cirílico (Windows)
1252	Windows-1252	Europeu Ocidental (Windows)
1253	windows-1253	Grego (Windows)
1254	windows-1254	Turco (Windows)
1255	windows-1255	Hebraico (Windows)
1256	windows-1256	Árabe (Windows)
1257	windows-1257	Báltico (Windows)

1258	windows-1258	Vietnamita (Windows)
1361	Johab	Coreano (Johab)
10000	macintosh	Europeu Ocidental (Mac)
10001	x-mac-japanese	Japonês (Mac)
10002	x-mac-chinesetrad	Chinês tradicional (Mac)
10003	x-mac-korean	Coreano (Mac)
10004	x mac-arabic	Árabe (Mac)
10005	x-mac-hebrew	Hebraico (Mac)
10006	x-mac-greek	Grego (Mac)
10007	x-mac-cyrillic	Cirílico (Mac)
10008	x-mac-chinesesimp	Chinês simplificado (Mac)
10010	x-mac-romanian	Romeno (Mac)
10017	x-mac-ukrainian	Ucraniano (Mac)
10021	x-mac-thai	Tailandês (Mac)
10029	x-mac-ce	Centro-europeu (Mac)
10079	x-mac-icelandic	Islandês (Mac)
10081	x-mac-turkish	Turco (Mac)
10082	x-mac-croatian	Croata (Mac)
12000	utf-32	Unicode (UTF-32)
12001	utf-32BE	Unicode (UTF-32-Big Endian)
20000	x-Chinese-CNS	Chinês tradicional (CNS)



20001	x-cp20001	TCA Taiwan
20002	x-Chinese-Eten	Chinês tradicional (Eten)
20003	x-cp20003	IBM5550 Taiwan
20004	x-cp20004	TeleText Taiwan
20005	x-cp20005	Wang Taiwan
20105	x-IA5	Europeu Ocidental (IA5)
20106	x-IA5-German	Alemão (IA5)
20107	x-IA5-Swedish	Sueco (IA5)
20108	x-IA5-Norwegian	Norueguês (IA5)
20127	us-ascii	US-ASCII
20261	x-cp20261	T.61
20269	x-cp20269	ISO-6937
20273	IBM273	IBM EBCDIC (Alemanha)
20277	IBM277	IBM EBCDIC (Dinamarca-Noruega)
20278	IBM278	IBM EBCDIC (Finlândia-Suécia)
20280	IBM280	IBM EBCDIC (Itália)
20284	IBM284	IBM EBCDIC (Espanha)
20285	IBM285	IBM EBCDIC (Grã-Bretanha)
20290	IBM290	IBM EBCDIC (Japonês katakana)
20297	IBM297	IBM EBCDIC (França)

20420	IBM420	IBM EBCDIC (Árabe)
20423	IBM423	IBM EBCDIC (Grego)
20424	IBM424	IBM EBCDIC (Hebraico)
20833	x-EBCDIC-KoreanExtended	IBM EBCDIC (Coreano, estendido)
20838	IBM-Thai	IBM EBCDIC (Tailandês)
20866	koi8-r	Cirílico (KOI8-R)
20871	IBM871	IBM EBCDIC (Islandês)
20880	IBM880	IBM EBCDIC (Cirílico, russo)
20905	IBM905	IBM EBCDIC (Turco)
20924	IBM00924	IBM Latino 1
20932	EUC-JP	Japonês (JIS 0208-1990 e 0212-1990)
20936	x-cp20936	GB2312-80 Chinês (simplificado)
20949	x-cp20949	Coreano Wansung
21025	cp1025	IBM EBCDIC (Cirílico, Servo-Búlgaro)
21866	koi8-u	Cirílico (KOI8-U)
28591	iso-8859-1	Europeu Ocidental (ISO)
28592	iso-8859-2	Centro-europeu (ISO)
28593	iso-8859-3	Latino 3 (ISO)
28594	iso-8859-4	Báltico (ISO)
28595	iso-8859-5	Cirílico (ISO)

28596	iso-8859-6	Árabe (ISO)
28597	iso-8859-7	Grego (ISO)
28598	iso-8859-8	Hebraico (ISO-Visual)
28599	iso-8859-9	Turco (ISO)
28603	iso-8859-13	Estoniano (ISO)
28605	iso-8859-15	Latino 9 (ISO)
29001	x-Europa	Europa
38598	iso-8859-8-i	Hebraico (ISO-Logical)
50220	iso-2022-jp	Japonês (JIS)
50221	csISO2022JP	Japonês (JIS, 1 byte kana permitido)
50222	iso-2022-jp	Japonês (JIS, 1 byte kana permitido-SO/SI)
50225	iso-2022-kr	Coreano (ISO)
50227	x-cp50227	ISO-2022 Chinês (simplificado)
51932	euc-jp	Japonês (EUC)
51936	EUC-CN	Chinês simplificado (EUC)
51949	euc-kr	Coreano (EUC)
52936	hz-gb-2312	Chinês simplificado (HZ)
54936	GB18030	GB18030 Chinês simplificado
57002	x-iscii-de	ISCII Devanágari
57003	x-iscii-be	ISCII Bengalês

57004	x-iscii-ta	ISCII Tâmil
57005	x-iscii-te	ISCII Télego
57006	x-iscii-as	ISCII Assamês
57007	x-iscii-or	ISCII Oriá
57008	x-iscii-ka	ISCII Kannada
57009	x-iscii-ma	ISCII Malaiala
57010	x-iscii-gu	ISCII Guzerate
57011	x-iscii-pa	ISCII Punjabi
65000	utf-7	Unicode (UTF-7)
65001	utf-8	Unicode (UTF-8)

**Valor de retorno**

-

**Exemplo**

```
StringListSave("SListe", "c:\Temp\Test.lst")
```

**StringListSetCommaText**

Grava um valor separado por vírgulas em uma Lista de strings.

**Sintaxe**

```
StringListSetCommaText(stringList: String ,commaText: String)
```

**Parâmetro**

Parâmetro	Significado
stringList	Nome da lista de strings
commaText	Valores a serem salvo na lista de strings (separados por vírgulas), por exemplo, "A, B, C"

**Valor de retorno**

-

**Exemplo**

```
StringListSetCommatext ("List1", "A,B,C")
```

**StringListSetValue**

Define o valor de um nome dentro de uma lista de strings no formato: "Name=Value".

**Sintaxe**

```
StringListSetValue(stringList: String ,aName: String ,aValue: String)
```

**Parâmetro**

Parâmetro	Significado
stringList	Nome da lista de strings
aName	Identificador do valor a ser definido.
aValue	Valor a ser atribuído ao identificador.

**Valor de retorno**

-

**Exemplo**

```
StringListSetValue ("Liste1", "Nachname", "Mustermann")
```

**StringListSort**

Classifica a lista de strings.

**Sintaxe**

```
StringListSort(stringList: String)
```

**Parâmetro**

Parâmetro	Significado
stringList	Nome da lista de strings

**Valor de retorno**

-

**Exemplo**

```
StringListSort("SListe")
```

## StrLeft

Formatação alinhada à esquerda de uma string.

### Sintaxe

StrLeft(inString: String): String

### Parâmetro

Parâmetro	Significado
inString	String ou variável

### Valor de retorno

String alinhada à esquerda

### Exemplo

```
Var = " 1234"
VarL = StrLeft(Var)
;--> VarL = "1234 "
```

## StrLen

Calcula o comprimento de uma string.

### Sintaxe

StrLen(inString: String): Integer

### Parâmetro

Parâmetro	Significado
inString	String ou variável

### Valor de retorno

Comprimento da string

### Exemplo

```
Var = "1234"
Len = StrLen(Var)
;-->Len = 4
```

## StrLower

Converte uma string para minúsculas.

### Sintaxe

StrLower(inString: String): String

### Parâmetro

Parâmetro	Significado
inString	String ou variável

### Valor de retorno

String em minúsculas

### Exemplo

```
Var = "PERCEPTIVE SOFTWARE"
VarLow = StrLower(Var)
; VarLow = "perceptive software"
```

### StrPos

Encontra uma string em uma string.

### Sintaxe

StrPos(searchStr: String ,sourceStr: String ,ignoreCase: Boolean): Integer

### Parâmetro

Parâmetro	Significado
searchStr	String a ser encontrada
sourceStr	String a ser pesquisada
ignoreCase	Diferenciação de maiúsculas/minúsculas True/False

### Valor de retorno

Posição da palavra na string (0 se a string pesquisada não tiver sido encontrada).

### Exemplo

```
SourceStr = "Monday Tuesday Wednesday"
SuchStr = "Tuesday"
res = StrPos(SuchStr, SourceStr, TRUE)
;-->res=8
```

### StrReplace

Substitui os caracteres em uma string com outros caracteres.

### Sintaxe

StrReplace(inString: String ,oldValue: String ,newValue: String): String

### Parâmetro

Parâmetro	Significado
inString	String ou variável
oldValue	Caractere a ser substituído
newValue	Caractere a ser inserido

### Valor de retorno

String modificada

### Exemplo

#### Exemplo - Substituir caracteres

```
var1 = "10/11/2011"
Var2 = StrReplace(Var1, "/", ".")
Var3 = StrReplace(Var1, "/", " - ")
;--> Var2 = "10,11.2011"
;--> Var3 = "10 - 11 - 2011"
```

#### Exemplo - Substituir aspas por análise

```
Var1 = ""Manfred","Peter""
;--> Var1 = "Manfred","Peter"
;
quote = Parse("^H22;")
Var2 = StrReplace(Var1, quote, "")
;
;--> Var2=Manfred,Peter
```

### Consulte também

[Analisar](#)

### StrReverse

Reverte a string.

### Sintaxe

StrReverse(inString: String): String

### Parâmetro

Parâmetro	Significado
-----------	-------------



inString	String ou variável
----------	--------------------

**Valor de retorno**

String revertida

**Exemplo**

```
Var = "PERCEPTIVE SOFTWARE"
VarR = StrReverse(Var)
;--> VarR = "ERAWTFOS EVITPECREP"
```

**StrRight**

Formatação alinhada à direita de uma string.

**Sintaxe**

StrRight(inString: String): String

**Parâmetro**

Parâmetro	Significado
inString	String ou variável

**Valor de retorno**

String alinhada à direita

**Exemplo**

```
Var = "1234 "
VarR = StrRight(Var)
; VarR = " 1234"
```

**StrTok**

Divide uma string após um delimitador especificado e retorna a string extraída na frente do delimitador como resultado.

**Sintaxe**

StrTok(varName: String ,delimiter: String): String

**Parâmetro**

Parâmetro	Significado
varName	Nome da variável de string
delimiter	Caractere separador

**Valor de retorno**

String extraída

**Exemplo**

```
Variable = "1;2;3;4;"
V = StrTok("Variable", ";") ;--> V="1" Variable="2;3;4;"
V = StrTok("Variable", ";") ;--> V="2" Variable="3;4;"
V = StrTok("Variable", ";") ;--> V="3" Variable="4;"
V = StrTok("Variable", ";") ;--> V="4" Variable=""
V = StrTok("Variable", ";") ;--> V="" Variable=""
```

**StrToLen**

Reduz uma string a um comprimento específico.

**Sintaxe**

StrToLen(inString: String ,maxLen: Integer ,fillChar: String ,doRight: Boolean): String

**Parâmetro**

Parâmetro	Significado
inString	String ou variável
maxLen	Comprimento do resultado Se o valor desse parâmetro for menor que o comprimento da string, ela será truncada nessa posição. Caso contrário, a string será preenchida com o caractere de preenchimento definido.
fillChar	Caractere de preenchimento Se nenhum caractere de preenchimento for definido, o caractere padrão será um espaço.
doRight	True = insere caracteres de preenchimento no início False = insere caracteres de preenchimento no final

**Valor de retorno**

String alterada

**Exemplo**

```
Var = "1234"
Var10 = StrToLen(Var, 10, "0", TRUE)
;--> Var10="0000001234"
Var10 = StrToLen(Var, 10, "0", FALSE)
;--> Var10="1234000000"
```

## StrUpper

Converte uma string para maiúsculas.

### Sintaxe

StrUpper(inString: String): String

### Parâmetro

Parâmetro	Significado
inString	String ou variável

### Valor de retorno

String em maiúsculas

### Exemplo

```
Var = "perceptive software"
VarUp = StrUpper(Var)
; VarUp = "PERCEPTIVE SOFTWARE"
```

## StrWord

Extrai a parte de uma string que está separada por um delimitador.

### Sintaxe

StrWord(inString: String ,wordNumber: Integer ,delimiter: String): String

### Parâmetro

Parâmetro	Significado
inString	String ou variável
wordNumber	Número da palavra começando por 1
delimiter	Caractere separador

### Valor de retorno

Substring extraída

### Exemplo

```
Var = StrWord("W1.w2.W3", 2, ".")
;--> Var = "w2"
```

## StrWordCount

Calcula o número de palavras de acordo com o delimitador especificado.

### Sintaxe

StrWordCount(inString: String ,delimiter: String): Integer

### Parâmetro

Parâmetro	Significado
inString	String ou variável
delimiter	Caractere separador

### Valor de retorno

Número de palavras

### Exemplo

```
Var = StrWordCount("W1.w2.W3", ".")
;--> Var = 3
```

## Trunc

Trunca as casas decimais após a vírgula.

### Sintaxe

Trunc(value: Double): Integer

### Parâmetro

Parâmetro	Significado
value	Número de ponto flutuante

### Valor de retorno

Número inteiro

### Exemplo

```
d = 1.9
i = trunc(d)
;--> i = 1
```

## UnRegisterLib

Cancelar o registro de uma biblioteca de função (DLL).

**Sintaxe**

```
UnRegisterLib(dllFileName: String)
```

**Parâmetro**

Parâmetro	Significado
dllFileName	Nome e caminho da DLL

**Valor de retorno**

-

**Exemplo**

```
UnRegisterLib("C:\Winnt\System\MSLDB_LIB.DLL")
```

**Update**

Altera um conjunto de dados e preenche os campos com valores.

**Sintaxe**

```
Update(datasetName: String ,rowNum: Integer ,fields: String[])
```

**Parâmetro**

Parâmetro	Significado
datasetName	Nome do conjunto de dados
rowNum	Número da linha após a qual os dados serão inseridos.
fields	Valores de campo na ordem correta

**Valor de retorno**

-

**Exemplo**

```
Update("DSName",1,"Perceptive Software","3","26.10.2011")
```

**VerifyPool**

Verifique se o pool especificado existe. Se não existir, uma exceção é lançada.

**Sintaxe**

```
VerifyPool(poolName: String)
```

**Parâmetro**

Parâmetro	Significado
poolName	Nome do pool a ser verificado.

**Valor de retorno**

-

**Exemplo**

```
VerifyPool("MyPool")
```

**Version**

Determina a versão do interpretador MonaLisa.

**Sintaxe**

```
Version(): String
```

**Parâmetro**

-

**Valor de retorno**

Número da versão

**Exemplo**

```
MLVersion=Version()
```

**Script de exemplo: acesso ao banco de dados**

Esse script executa uma instrução de seleção em um banco de dados e reitera pelos resultados.

```
myDBAlias = "OrderDataDbAlias" ; Name of existing DB Alias to use for
connection to the DB
```

```
myDataSet = "MyOrderDataDbDataSet" ; Name of local dataset to use
```

```
mySQLStatement = "Select * FROM modus_xml_job WHERE state=:myState and
SelParams=:mySelParam" ; The sql-statement using parameters
```

```
mySQLStringList = "SqlList" ; Name of the local list to use for the sql-
statement
```

```
myParamDescList = "ParamDescList" ; Name of the local parameter description
list
```

```
InitDatabase(myDBAlias) ; Initiate DB connection
```

```
InitDataset(myDataSet, myDBAlias) ; Create local dataset
```

```
StringlistAdd(mySQLStringList, mySQLStatement) ; Add sql-statement to the
sql-stringlist
```

```

;Fill parameter description list
StringListClear(myParamDescList)
StringListSetValue(myParamDescList, "myState", "ii")
StringListSetValue(myParamDescList, "mySelParam", "ii")

myState = "6"           ;Local variable used as first parameter
mySelParam ="0815"     ;Local variable used as second parameter

SetSQL(myDataSet, mySQLStringList)      ; Setting sql-statement (with no
parameters in the statement, sql will be executed immediately)
PrepareSQL(myDataSet, myParamDescList) ; Executing sql-statement with
parameters

While (EOF(myDataSet) = false)           ; As long as not EndOfFile
    BundleName = GetField(myDataSet,"BundleName") ; Get the desired data
    Protocol("BundleName: {0}", 8, BundleName) ; Do something with the
data
    Next(myDataSet)                       ; Position dataset cursor
to next set of data
End-while

```

## Funções Xdata

### Controle de transações

#### **Informações gerais**

Os métodos StartTransaction, Commit e Rollback permitem que você agrupe diversas operações de banco de dados (selecionar, atualizar, excluir, inserir) em uma transação.

#### **Exemplo**

No script a seguir para um banco de dados Oracle, a seção **Q1** executa uma instrução de seleção no banco de dados.

Após qualquer processamento de dados no script, a função **ExecuteQuery()** executa uma operação no conjunto de dados selecionado.

#### **Instruções SQL**

```
Q1: select * from TEST where status = 0 and id = (select min(id) from test
where status = 0) for update
```

```
Update_Status: update TEST set status = 1 where id = :ID
```

Para assegurar que os dados estão bloqueados para outra sessão do banco de dados entre **Q1** e **Update\_Status**, a instrução de seleção deve conter um bloqueio de atualização usando o comando **for update**.

Os dois comandos SQL devem ser executados na transação para manter o bloqueio do banco de dados até que a atualização tenha sido executada.

```
db.StartTransaction()  
db.WorkWithQuery("Q1")  
db.ApplyDataSet()  
...  
system.ID = db.GetField("ID")  
db.ExecuteQuery("Update_Status")  
db.Commit()
```

### ***StartTransaction***

Inicia uma transação no banco de dados.

#### **Sintaxe**

```
StartTransaction()
```

#### **Parâmetro**

-

#### **Valor de retorno**

-

### ***Confirmar***

Encera uma transação de banco de dados.

#### **Sintaxe**

```
Commit()
```

#### **Parâmetro**

-

#### **Valor de retorno**

-

### ***Reverter***

Descarta todas as alterações, insere e exclui ações da transação atual e fecha a transação.

#### **Sintaxe**

```
Rollback()
```

#### **Parâmetro**

-



**Valor de retorno**

-

**AddDataRecord**

Este método adiciona um conjunto de dados ao objeto de dados atual.

**Sintaxe**

AddDataRecord(valoresColunas: Variável,...)

**Parâmetro**

Parâmetro	Significado
valoresColunas	Valores que serão inseridos como um conjunto de dados no objeto de dados

**Valor de retorno**

-

**Exemplo**

```
db.AddDataRecord(1234, "Máx. ", "Mustermann")
```

**ApplyDataRecord**

Aplica o conjunto de dados atual do modelo de lógico de dados.

**Sintaxe**

ApplyDataRecord()

**Parâmetro**

-

**Valor de retorno**

-

**Exemplo**

```
db.ApplyDataRecord()
```

**ApplyDataSet**

Aplica os dados da última instrução lida ao modelo lógico de dados.

Todos os conjuntos de dados são aplicados onde relações 1:N são encontradas.

**Sintaxe**

ApplyDataSet()

**Parâmetro**

-

**Valor de retorno**

-

**Exemplo**

```
db.ApplyDataSet()
```

**DefineContainer**

Define um contêiner para extrair valores de uma coluna de banco de dados e armazená-los nas chamadas colunas do contêiner.

**Sintaxe**

```
DefineContainer(usingMode: Integer ,fieldName: String ,delimiter: Integer)
```

```
DefineContainer(usingMode: Integer ,fieldName: String ,delimiterList: String)
```

**Parâmetro**

Parâmetro	Significado
usingMode	Indisponível no momento. O termo é reservado para implementação futura.
fieldName	Nome do campo do contêiner na tabela do banco de dados.
delimiter	Código ASCII do delimitador, por exemplo, 44 para uma vírgula.
delimiterList	Lista de delimitadores separada por vírgulas. Um delimitador é definido por seu código ASCII, por exemplo, "44,59" especifica uma vírgula ou ponto e vírgula como possíveis delimitadores.

**Valor de retorno**

-

**Exemplo**

```
db.WorkWithQuery("Q1")
db.DefineContainer(0, "ContainerCol", 59)
db.ApplyDataSet()
```

**Eof**

Determina se o cursor está no final de um conjunto de dados.

**Sintaxe**

```
Eof(): Boolean
```

**Parâmetro**

-

**Valor de retorno**

True / False

**Exemplo**

```
End_Of_File = db.Eof()
```

**ExecuteQuery**

Executa uma consulta SQL.

**Sintaxe**

```
ExecuteQuery(queryName: String)
```

**Parâmetro**

Parâmetro	Significado
queryName	Nome da consulta a ser executada

**Valor de retorno**

-

**Exemplo**

```
db.ExecuteQuery("Q2")
```

**First**

Posiciona o cursor do conjunto de dados na parte superior do objeto de dados atual.

**Sintaxe**

```
First()
```

**Parâmetro**

-

**Valor de retorno**

-

**Exemplo**

```
db.First()
```

**GetField**

Retorna o valor de campo como uma string.

**Sintaxe**

GetField(fieldName: String): String

**Parâmetro**

Parâmetro	Significado
fieldName	Nome do campo cujos valores serão lidos.

**Valor de retorno**

Valor de campo

**Exemplo**

```
db.WorkWithQuery("Q1")
db.ApplyDataSet()
System.Par_Name = db.GetField("PAR_NAME")
```

**Last**

Posiciona o cursor do conjunto de dados no último conjunto de dados do objeto de dados atual.

**Sintaxe**

Last()

**Parâmetro**

-

**Valor de retorno**

-

**Exemplo**

```
db.Last()
```

**Message**

Se ocorreu um erro, a mensagem resultante pode ser processada por esse método

**Sintaxe**

Message(): String

**Parâmetro**

-

**Valor de retorno**

Mensagem de erro

**Exemplo**

```
FM=db.Message ( )
```

**Mensagem**

Se ocorreu um erro, a mensagem resultante pode ser processada por esse método

**Sintaxe**

```
Mensagem(): String
```

**Parâmetro**

-

**Valor de retorno**

Mensagem de erro

**Exemplo**

```
FM=db.Message ( )
```

**Próximo**

Posiciona o cursor do conjunto de dados no próximo conjunto do objeto de dados atual.

**Sintaxe**

```
Next()
```

**Parâmetro**

-

**Valor de retorno**

-

**Exemplo**

```
db.Next ( )
```

**PrepareParameter**

Substitui parâmetros sem denominação (%s) em uma instrução SQL.

Parâmetros sem denominação (%s) são substituídos em uma instrução SQL antes de serem passados para o banco de dados.

Para obter a valor a ser usado, agora você precisa definir em qual variável de pool de sistema o valor é encontrado.

A função PrepareParameter é usada para fazer isso. A função passa uma string separada por vírgula contendo os nomes das variáveis MonaLisa correspondentes na ordem em que são usadas na instrução SQL.

Os parâmetros não denominados devem ser enviados da mesma maneira que o banco de dados espera que sejam, por exemplo, incluídos entre aspas simples.

### Sintaxe

PrepareParameter(paramValues: String)

### Parâmetro

Parâmetro	Significado
paramValues	Nome da variável separado por vírgula

### Valor de retorno

-

### Exemplo

#### Leitura do script

```
system.ParNum = "0815"
system.LfdNr = "'2'"
system.Typ = "'N'"
db.PrepareParameter("LfdNr, Typ")
db.WorkWithQuery("Q1")
db.ApplyDataSet()
```

### Consulta

```
Selecionar * de "KONTEN.DB"
em que (KTO_Lfd_Num = %s)
E (KTO_Par_Num = :ParNum)
E (KTO_Typ = %s)
```

### Anterior

Posiciona o cursor do conjunto de dados no conjunto de dados anterior do objeto de dados atual.

### Sintaxe

Prev()

### Parâmetro

-

### Valor de retorno

-

### Exemplo

```
db.Prev()
```

### RowCount

Retorna o número de conjuntos de dados lidos.

#### Sintaxe

```
RowCount(): Integer
```

#### Parâmetro

-

#### Valor de retorno

Número de conjunto de dados

#### Exemplo

```
i=db.RowCount()
```

### StartProcedure

Executa um procedimento armazenado.

#### Sintaxe

```
StartProcedure(procedureName: String)
```

```
StartProcedure(procedureName: String ,usingMode: Integer)
```

#### Parâmetro

Parâmetro	Significado	
procedureName	Nome do procedimento armazenado	
usingMode	0	Vínculo com base em uma matriz de parâmetro. Os parâmetros são especificados de acordo com a ordem das definições de procedimentos armazenados.
	1	Vínculo com base em um nome de parâmetro. Os parâmetros são passados por meio de comparação de nomes.

#### Valor de retorno

-

#### Exemplo

```
db.StartProcedure("DM_PROCEDURE", 1)
```

## WorkWithQuery

Abre um conjunto de dados com uma consulta de SQL.

### Sintaxe

WorkWithQuery(queryName: String)

### Parâmetro

Parâmetro	Significado
queryName	Nome da consulta

### Valor de retorno

-

### Exemplo

```
db.WorkWithQuery("Q1")
```

## Funções de data

### Geral

O objeto `MLDate` do `MonaLisa` cria um objeto com função de data.

Os especificadores de formato necessários como um parâmetro por alguns métodos devem estar em conformidade com as regras estabelecidas pelo `.NET framework`.

### Exemplos

Especificadores de formato	Resultado
dd	Dia, dois dígitos numéricos
ddd	Nome do dia da semana (abreviado)
dddd	Nome do dia da semana (completo)
MM	Mês, dois dígitos numéricos
MMM	Nome do mês (abreviado)
MMMM	Nome do mês (completo)
yy	Ano, dois dígitos numéricos
yyyy	Ano, quatro dígitos numéricos



hh	Horas
mm	Minutos
ss	Segundos

Em geral, todas as strings de formatação de data e hora do .NET Framework são suportadas.

Para uma tabela de todos os especificadores de formato possíveis, consulte este link:

<http://msdn.microsoft.com/en-us/library/97x6twsz.aspx>

### Métodos da classe MLDate

#### ***CurrentDate***

Retorna a data atual como um número de série.

#### **Sintaxe**

CurrentDate()

#### **Parâmetro**

-

#### **Valor de retorno**

Data como um número de série.

#### **Exemplo**

```
GetObject("date", "MLDate")
CurrDateAsInt = date.CurrentDate()
```

#### ***CurrentDateToString***

Retorna a data atual como uma string de data.

#### **Sintaxe**

CurrentDateToString(string format)

#### **Parâmetro**

Parâmetro	Significado
format	Regras de formato que definem como a string de data é mostrada.

#### **Valor de retorno**

A data atual formatada como uma string de data.

**Exemplo**

```
GetObject("date", "MLDate")
CurrDateAsStr = date.CurrentDateToString("dd.MM.yyyy")
```

***DateStringToInt64***

Converte uma string de data em uma data de número de série.

**Sintaxe**

```
DateStringToInt64(string format, string date)
```

**Parâmetro**

Parâmetro	Significado
format	Regras de formato que definem como a string de data é mostrada.
date	String de data a converter

**Valor de retorno**

Número de série da data

**Exemplo**

```
GetObject("date", "MLDate")
DateStrToInt = date.DateStringToInt64("MM.dd.yyyy", "10.31.2011")
```

***DayOfWeek***

Retorna o dia da semana para uma data especificada como um número de série.

**Sintaxe**

```
DayOfWeek(long date)
```

**Parâmetro**

Parâmetro	Significado
date	Data inteira

**Valor de retorno**

Número do dia da semana, 0 = Domingo bis 6 = Sábado

**Exemplo**

```
GetObject("date", "MLDate")
CurrDateAsInt = date.CurrentDate()
```

```
DayOfWeek = date.DayOfWeek(CurrDateAsInt)
```

### ***DaysInMonth***

Calcula o número de dias no mês especificado para um ano específico.

#### **Sintaxe**

```
DaysInMonth(int month, int year)
```

#### **Parâmetro**

Parâmetro	Significado
month	Mês como número
year	Ano como número

#### **Valor de retorno**

Número de dias

#### **Exemplo**

```
GetObject("date", "MLDate")
DaysInMonth = date.DaysInMonth(2, 2011)
```

### ***FormatDateTimeString***

Formata a string de data especificada de acordo com a regra para a string de data.

#### **Sintaxe**

```
FormatDateTimeString(string format, string date)
```

#### **Parâmetro**

Parâmetro	Significado
format	Regras de formato que definem como a string de data é mostrada.
date	String de data

#### **Valor de retorno**

String de data formatada

#### **Exemplo**

```
GetObject("date", "MLDate")
FormattedDateTimeStr = date.FormatDateTimeString("M.d.yyy", "10.31.2011")
```

***IncDate***

Adiciona um número específico de dias, meses e anos a uma data.

Se um número negativo for definido, o número será subtraído da data.

**Sintaxe**

`IncDate(long date, int days, int months, int years)`

**Parâmetro**

Parâmetro	Significado
date	Número de série da data
days	Número de dias
months	Número de meses
years	Número de anos

**Valor de retorno**

Número de série da data

**Exemplo**

```
GetObject("date", "MLDate")
CurrDateAsInt = date.CurrentDate()
IncDateInt = date.IncDate(CurrDateAsInt, 10, 0, 0)
```

***Int64ToDateString***

Converte o número de série da data especificada em uma string de data formatada.

**Sintaxe**

`Int64ToDateString(string format, long date)`

**Parâmetro**

Parâmetro	Significado
format	Regras de formato que definem como a string de data é mostrada.
date	Número de série da data

**Valor de retorno**

Data como uma string de data formatada.

**Exemplo**

```
GetObject("date", "MLDate")
CurrDateAsInt = date.CurrentDate()
DateIntToStr = date.Int64ToDateString("MM.dd.yyyy", CurrDateAsInt)
```

***IsLeapYear***

Verifica se o ano em questão é bissexto.

**Sintaxe**

IsLeapYear(int year)

**Parâmetro**

Parâmetro	Significado
year	Ano como número

**Valor de retorno**

Valor	Significado
True	Ano bissexto
False	Não é ano bissexto

**Exemplo**

```
GetObject("date", "MLDate")
IsLeapYear = date.IsLeapYear(2011)
```

***WeekOfYear***

Retorna o número da semana a partir do número de série da data especificada, de acordo com a ISO 8601.

**Sintaxe**

WeekOfYear(long date)

**Parâmetro**

Parâmetro	Significado
date	Número de série da data

**Valor de retorno**

Número da semana

**Exemplo**

```
GetObject("date", "MLDate")
CurrDateAsInt = date.CurrentDate()
WeekOfYear = date.WeekOfYear(CurrDateAsInt)
```

**FileProcessing****Métodos da classe MLDos*****ChangeExtension***

Muda a extensão da string do caminho.

**Sintaxe**

```
ChangeExtension(string path, string extension)
```

**Parâmetro**

Parâmetro	Significado
path	Informação do caminho a ser alterado
extension	A nova extensão (com um ponto à esquerda). Insira uma string vazia para remover a extensão existente de 'path'.

**Valor de retorno**

Uma string contendo a informação do caminho alterado.

***Combine***

Combina duas strings de caminho.

**Sintaxe**

```
Combine(string path1, string path2)
```

**Parâmetro**

Parâmetro	Significado
path1	Primeiro caminho
path2	Segundo caminho

**Valor de retorno**

Uma string contendo as entradas de caminho combinadas.

Uma string contendo os caminhos combinados. Se um dos caminhos especificados for uma string de comprimento zero, esse método retorna o outro caminho. Se path2 contiver um caminho absoluto, esse método retorna path2.

### ***CopyFile***

Copia um arquivo existente para um novo arquivo.

#### **Sintaxe**

CopyFile(string sourceFileName, string destFileName)

CopyFile(string sourceFileName, string destFileName, bool overwrite)

#### **Parâmetro**

Parâmetro	Significado
sourceFileName	O arquivo a ser copiado.
destFileName	Nome do arquivo de destino. O nome não pode ser um diretório e, de o parâmetro 'overwrite' não estiver definido, não pode ser um arquivo existente.
overwrite	True, quando o arquivo de destino pode ser sobrescrito, em caso contrário, False

#### **Valor de retorno**

-

### ***CreateTextFile***

Cria um arquivo de texto que pode ser usado para ler, gravar ou anexar ao arquivo.

#### **Sintaxe**

CreateTextFile()

#### **Parâmetro**

Parâmetro	Significado				
newLine	Define a string para uma quebra de linha. Se o parâmetro não for especificado, CRLF ("\r\n" - \$0D \$0A) é usado como quebra de linha padrão. <b>Valores possíveis:</b>				
	<table border="1"> <thead> <tr> <th>Valor</th> <th>Resultado</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> </tr> </tbody> </table>	Valor	Resultado		
Valor	Resultado				

	DOS.NewLineLF	LineFeed ("\n" - \$0A)	
	DOS.NewLineCRLF	CarriageReturn/LineFeed ("\r\n" - \$0D \$0A)	

**Valor de retorno**

Objeto MonalisaTextFile que pode ser usado para ler, gravar ou anexar ao arquivo.

**CreateDirectory**

Cria todos os diretórios e subdiretórios de acordo com um caminho especificado.

**Sintaxe**

CreateDirectory(string path)

**Parâmetro**

Parâmetro	Significado
path	O caminho do diretório a ser criado.

**Valor de retorno**

-

**DeleteDirectory**

Exclui o diretório especificado e, se assim configurado, todos os subdiretórios do mesmo.

**Sintaxe**

DeleteDirectory(string path)

DeleteDirectory(string path, bool recursive)

**Parâmetro**

Parâmetro	Significado
path	Nome do diretório vazio a ser excluído. O diretório deve ser gravável ou vazio.
recursive	True para excluir diretórios, subdiretórios e arquivos no caminho. Em caso contrário, False.

**Valor de retorno**

-

**DeleteFile**



Exclui o arquivo especificado.

Nenhuma exceção é lançada se o arquivo não for encontrado.

### Sintaxe

DeleteFile(string path)

### Parâmetro

Parâmetro	Significado
path	Nome do arquivo a ser excluído.

### Valor de retorno

-

### *DeleteFiles*

Exclui todos os arquivos em um diretório e subdiretório que correspondam a um padrão de pesquisa.

### Sintaxe

DeleteFiles(string path, string searchPattern, bool recursive)

### Parâmetro

Parâmetro	Significado
path	Nome do diretório
searchPattern	A string de pesquisa para corresponder aos nomes dos arquivos no caminho.
recursive	True: pesquisar subdiretórios False: não pesquisar subdiretórios

### Valor de retorno

-

### *DirectoryExists*

Determina se o caminho especificado aponta para um diretório em uma unidade.

### Sintaxe

DirectoryExists(string path)

### Parâmetro

Parâmetro	Significado
path	Caminho a ser testado

**Valor de retorno**

**True:** o caminho existe.

**False:** o caminho não existe.

***ExecuteSynchron***

Inicia o aplicativo especificado e retorna um código de saída quando o aplicativo é fechado.

**Observação para os sistemas operacionais de 64 bits:**

O ModusOne é um aplicativo de 32 bits

Se o comando **ExecuteSynchron** for usado para iniciar um programa do diretório

`%windir%\System32` ou

`C:\windows\System32`

o diretório

`%windir%\Sysnative` ou

`c:\windows\Sysnative`

deve ser definido.

**Para obter mais informações, use este link:**

[http://msdn.microsoft.com/en-us/library/aa384187\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/aa384187(v=vs.85).aspx)

**Sintaxe**

`ExecuteSynchron(string file, string parameter, bool hide)`

**Parâmetro**

Parâmetro	Significado
file	Aplicativo com caminho
parameter	Parâmetros de aplicativo
hide	True: oculta a janela do aplicativo iniciado False: mostra a janela do aplicativo iniciado

**Valor de retorno**

Código de saída do aplicativo

***FileAppendText***

Abre um arquivo, anexa a string especificada ao arquivo e então fecha o arquivo.

Se o arquivo ainda não existe, o método irá criá-lo, gravar a string no arquivo e fechá-lo.

### Sintaxe

```
FileAppendText(string fileName, string content)
```

```
FileAppendText(string fileName, string content, string encoding)
```

### Parâmetro

Parâmetro	Significado																					
fileName	O arquivo ao qual a string está anexada.																					
content	A string a ser anexada no arquivo.																					
encoding	<p>Os seguintes podem ser usados como parâmetro de codificação:</p> <ul style="list-style-type: none"> <li>String vazia. Nesse caso, a codificação padrão (página de código ANSI do sistema operacional) é usada.</li> <li>página de código</li> <li>nome</li> <li>Nome amigável</li> </ul> <p>Exemplo:</p> <ul style="list-style-type: none"> <li>"1252"</li> <li>"windows-1252"</li> <li>"Europeu Ocidental" (Windows)</li> </ul> <p>Parâmetros de codificação possíveis</p> <table border="1"> <thead> <tr> <th>Página de código</th> <th>Nome</th> <th>Nome de exibição</th> </tr> </thead> <tbody> <tr> <td>37</td> <td>IBM037</td> <td>IBM EBCDIC (EUA-Canadá)</td> </tr> <tr> <td>437</td> <td>IBM437</td> <td>OEM EUA</td> </tr> <tr> <td>500</td> <td>IBM500</td> <td>IBM EBCDIC (Internacional)</td> </tr> <tr> <td>708</td> <td>ASMO-708</td> <td>Árabe (ASMO 708)</td> </tr> <tr> <td>720</td> <td>DOS-720</td> <td>Árabe (DOS)</td> </tr> <tr> <td>737</td> <td>ibm737</td> <td>Grego (DOS)</td> </tr> </tbody> </table>	Página de código	Nome	Nome de exibição	37	IBM037	IBM EBCDIC (EUA-Canadá)	437	IBM437	OEM EUA	500	IBM500	IBM EBCDIC (Internacional)	708	ASMO-708	Árabe (ASMO 708)	720	DOS-720	Árabe (DOS)	737	ibm737	Grego (DOS)
Página de código	Nome	Nome de exibição																				
37	IBM037	IBM EBCDIC (EUA-Canadá)																				
437	IBM437	OEM EUA																				
500	IBM500	IBM EBCDIC (Internacional)																				
708	ASMO-708	Árabe (ASMO 708)																				
720	DOS-720	Árabe (DOS)																				
737	ibm737	Grego (DOS)																				

775	ibm775	Báltico (DOS)
850	ibm850	Europeu Ocidental (DOS)
852	ibm852	Europeu Oriental (DOS)
855	IBM855	OEM Cirílico
857	ibm857	Turco (DOS)
858	IBM00858	OEM Multilíngue Latino I
860	IBM860	Português (DOS)
861	ibm861	Islandês (DOS)
862	DOS-862	Hebraico (DOS)
863	IBM863	Francês do Canadá (DOS)
864	IBM864	Árabe (864)
865	IBM865	Nórdico (DOS)
866	cp866	Cirílico (DOS)
869	ibm869	Grego moderno (DOS)
870	IBM870	IBM EBCDIC (Multilíngue Latino 2)
874	windows-874	Tailandês (Windows)
875	cp875	IBM EBCDIC (Grego moderno)
932	shift_jis	Japonês (Shift-JIS)
936	gb2312	Chinês simplificado (GB2312)
949	ks_c_5601-1987	Coreano

950	big5	Chinês (Tradicional) (Big5)
1026	IBM1026	IBM EBCDIC (Turco, Latino 5)
1047	IBM01047	IBM Latino 1
1140	IBM01140	IBM EBCDIC (EUA-Canadá-Europeu)
1141	IBM01141	IBM EBCDIC (Alemanha-Europeu)
1142	IBM01142	IBM EBCDIC (Dinamarca-Noruega-Europeu)
1143	IBM01143	IBM EBCDIC (Finlândia-Suécia-Europeu)
1144	IBM01144	IBM EBCDIC (Italiano-Europeu)
1145	IBM01145	IBM EBCDIC (Espanha-Europeu)
1146	IBM01146	IBM EBCDIC (Grã-Bretanha-Europeu)
1147	IBM01147	IBM EBCDIC (França-Europeu)
1148	IBM01148	IBM EBCDIC (Internacional-Europeu)
1149	IBM01149	IBM EBCDIC (Islandês-Europeu)
1200	utf-16	Unicode
1201	unicodeFFFE	Unicode (Big Endian)
1250	windows-1250	Centro-europeu (Windows)
1251	windows-1251	Cirílico (Windows)

1252	Windows-1252	Europeu Ocidental (Windows)
1253	windows-1253	Grego (Windows)
1254	windows-1254	Turco (Windows)
1255	windows-1255	Hebraico (Windows)
1256	windows-1256	Árabe (Windows)
1257	windows-1257	Báltico (Windows)
1258	windows-1258	Vietnamita (Windows)
1361	Johab	Coreano (Johab)
10000	macintosh	Europeu Ocidental (Mac)
10001	x-mac-japanese	Japonês (Mac)
10002	x-mac-chinesetrad	Chinês tradicional (Mac)
10003	x-mac-korean	Coreano (Mac)
10004	x mac-arabic	Árabe (Mac)
10005	x-mac-hebrew	Hebraico (Mac)
10006	x-mac-greek	Grego (Mac)
10007	x-mac-cyrillic	Cirílico (Mac)
10008	x-mac-chinesesimp	Chinês simplificado (Mac)
10010	x-mac-romanian	Romeno (Mac)
10017	x-mac-ukrainian	Ucraniano (Mac)
10021	x-mac-thai	Tailandês (Mac)

10029	x-mac-ce	Centro-europeu (Mac)
10079	x-mac-icelandic	Islandês (Mac)
10081	x-mac-turkish	Turco (Mac)
10082	x-mac-croatian	Croata (Mac)
12000	utf-32	Unicode (UTF-32)
12001	utf-32BE	Unicode (UTF-32-Big Endian)
20000	x-Chinese-CNS	Chinês tradicional (CNS)
20001	x-cp20001	TCA Taiwan
20002	x-Chinese-Eten	Chinês tradicional (Eten)
20003	x-cp20003	IBM5550 Taiwan
20004	x-cp20004	TeleText Taiwan
20005	x-cp20005	Wang Taiwan
20105	x-IA5	Europeu Ocidental (IA5)
20106	x-IA5-German	Alemão (IA5)
20107	x-IA5-Swedish	Sueco (IA5)
20108	x-IA5-Norwegian	Norueguês (IA5)
20127	us-ascii	US-ASCII
20261	x-cp20261	T.61
20269	x-cp20269	ISO-6937
20273	IBM273	IBM EBCDIC (Alemanha)
20277	IBM277	IBM EBCDIC (Dinamarca-Noruega)

20278	IBM278	IBM EBCDIC (Finlândia-Suécia)
20280	IBM280	IBM EBCDIC (Itália)
20284	IBM284	IBM EBCDIC (Espanha)
20285	IBM285	IBM EBCDIC (Grã-Bretanha)
20290	IBM290	IBM EBCDIC (japonês Katakana)
20297	IBM297	IBM EBCDIC (França)
20420	IBM420	IBM EBCDIC (Árabe)
20423	IBM423	IBM EBCDIC (Grego)
20424	IBM424	IBM EBCDIC (Hebraico)
20833	x-EBCDIC-KoreanExtended	IBM EBCDIC (Coreano, estendido)
20838	IBM-Thai	IBM EBCDIC (Tailandês)
20866	koi8-r	Cirílico (KOI8-R)
20871	IBM871	IBM EBCDIC (Islandês)
20880	IBM880	IBM EBCDIC (Cirílico, Russo)
20905	IBM905	IBM EBCDIC (Turco)
20924	IBM00924	IBM Latino 1
20932	EUC-JP	Japonês (JIS 0208-1990 e 0212-1990)
20936	x-cp20936	GB2312-80 Chinês (simplificado)
20949	x-cp20949	Coreano Wansung
21025	cp1025	IBM EBCDIC (Cirílico, Servo-



			Búlgaro)
21866	koi8-u		Cirílico (KOI8-U)
28591	iso-8859-1		Europeu Ocidental (ISO)
28592	iso-8859-2		Centro-europeu (ISO)
28593	iso-8859-3		Latino 3 (ISO)
28594	iso-8859-4		Báltico (ISO)
28595	iso-8859-5		Cirílico (ISO)
28596	iso-8859-6		Árabe (ISO)
28597	iso-8859-7		Grego (ISO)
28598	iso-8859-8		Hebraico (ISO-Visual)
28599	iso-8859-9		Turco (ISO)
28603	iso-8859-13		Estoniano (ISO)
28605	iso-8859-15		Latino 9 (ISO)
29001	x-Europa		Europa
38598	iso-8859-8-i		Hebraico (ISO-Logical)
50220	iso-2022-jp		Japonês (JIS)
50221	csISO2022JP		Japonês (JIS, 1 byte kana permitido)
50222	iso-2022-jp		Japonês (JIS, 1 byte kana permitido-SO/SI)
50225	iso-2022-kr		Coreano (ISO)
50227	x-cp50227		ISO-2022 Chinês (simplificado)
51932	euc-jp		Japonês (EUC)

51936	EUC-CN	Chinês simplificado (EUC)
51949	euc-kr	Coreano (EUC)
52936	hz-gb-2312	Chinês simplificado (HZ)
54936	GB18030	GB18030 Chinês simplificado
57002	x-iscii-de	ISCII Devanágari
57003	x-iscii-be	ISCII Bengalês
57004	x-iscii-ta	ISCII Tâmil
57005	x-iscii-te	ISCII Télego
57006	x-iscii-as	ISCII Assamês
57007	x-iscii-or	ISCII Oriá
57008	x-iscii-ka	ISCII Kannada
57009	x-iscii-ma	ISCII Malaiala
57010	x-iscii-gu	ISCII Guzerate
57011	x-iscii-pa	ISCII Punjabi
65000	utf-7	Unicode (UTF-7)
65001	utf-8	Unicode (UTF-8)

**Valor de retorno**

-

***FileExists***

Verifica que um arquivo especificado existe.

**Sintaxe**

FileExists(string path)

**Parâmetro**

Parâmetro	Significado
path	O arquivo a ser verificado.

**Valor de retorno**

True: o chamador tem direitos suficientes e o **caminho** contém o nome de um arquivo existente.

False: Caminho inválido ou comprimento de string 0 definido para o **caminho**.

Uma exceção é lançada se o chamador não tiver direitos suficientes para ler o arquivo especificado. O método retorna False independentemente da existência do **caminho**.

***FileReadAllText***

Abre um arquivo de texto, lê todas as linhas do arquivo e então fecha o arquivo.

**Sintaxe**

FileReadAllText(string fileName)

FileReadAllText(string fileName, string encoding)

**Parâmetro**

Parâmetro	Significado						
fileName	O arquivo a ser aberto.						
encoding	<p>As opções a seguir podem ser usadas como parâmetro de codificação:</p> <ul style="list-style-type: none"> <li>String vazia. Nesse caso, a codificação padrão (página de código ANSI do sistema operacional) é usada.</li> <li>página de código</li> <li>nome</li> <li>nome amigável</li> </ul> <p>Exemplo:</p> <ul style="list-style-type: none"> <li>"1252"</li> <li>"windows-1252"</li> <li>"Europeu Ocidental" (Windows)</li> </ul> <p>Possíveis parâmetros de codificação</p> <table border="1"> <thead> <tr> <th>Página de código</th> <th>Nome</th> <th>Nome para exibição</th> </tr> </thead> <tbody> <tr> <td>37</td> <td>IBM037</td> <td>IBM EBCDIC (EUA-Canadá)</td> </tr> </tbody> </table>	Página de código	Nome	Nome para exibição	37	IBM037	IBM EBCDIC (EUA-Canadá)
Página de código	Nome	Nome para exibição					
37	IBM037	IBM EBCDIC (EUA-Canadá)					

437	IBM437	OEM EUA
500	IBM500	IBM EBCDIC (Internacional)
708	ASMO-708	Árabe (ASMO 708)
720	DOS-720	Árabe (DOS)
737	ibm737	Grego (DOS)
775	ibm775	Báltico (DOS)
850	ibm850	Europeu Ocidental (DOS)
852	ibm852	Europeu Oriental (DOS)
855	IBM855	OEM Cirílico
857	ibm857	Turco (DOS)
858	IBM00858	OEM Multilíngue Latino I
860	IBM860	Português (DOS)
861	ibm861	Islandês (DOS)
862	DOS-862	Hebraico (DOS)
863	IBM863	Francês do Canadá (DOS)
864	IBM864	Árabe (864)
865	IBM865	Nórdico (DOS)
866	cp866	Cirílico (DOS)
869	ibm869	Grego moderno (DOS)
870	IBM870	IBM EBCDIC (Multilíngue Latino 2)
874	windows-874	Tailandês (Windows)

875	cp875	IBM EBCDIC (Grego moderno)
932	shift_jis	Japonês (Shift-JIS)
936	gb2312	Chinês simplificado (GB2312)
949	ks_c_5601-1987	Coreano
950	big5	Chinês (tradicional) (Big5)
1026	IBM1026	IBM EBCDIC (Turco, Latino 5)
1047	IBM01047	IBM Latino 1
1140	IBM01140	IBM EBCDIC (EUA-Canadá-Europeu)
1141	IBM01141	IBM EBCDIC (Alemanha-Europeu)
1142	IBM01142	IBM EBCDIC (Dinamarca-Noruega-Europeu)
1143	IBM01143	IBM EBCDIC (Finlândia-Suécia-Europeu)
1144	IBM01144	IBM EBCDIC (Italiano-Europeu)
1145	IBM01145	IBM EBCDIC (Espanha-Europeu)
1146	IBM01146	IBM EBCDIC (Grã-Bretanha-Europeu)
1147	IBM01147	IBM EBCDIC (França-Europeu)
1148	IBM01148	IBM EBCDIC (Internacional-Europeu)
1149	IBM01149	IBM EBCDIC (Islandês-

			Europeu)
1200	utf-16		Unicode
1201	unicodeFFFE		Unicode (Big Endian)
1250	windows-1250		Europeu Central (Windows)
1251	windows-1251		Cirílico (Windows)
1252	Windows-1252		Europeu Ocidental (Windows)
1253	windows-1253		Grego (Windows)
1254	windows-1254		Turco (Windows)
1255	windows-1255		Hebraico (Windows)
1256	windows-1256		Árabe (Windows)
1257	windows-1257		Báltico (Windows)
1258	windows-1258		Vietnamita (Windows)
1361	Johab		Coreano (Johab)
10000	macintosh		Europeu Ocidental (Mac)
10001	x-mac-japanese		Japonês (Mac)
10002	x-mac-chinesetrad		Chinês tradicional (Mac)
10003	x-mac-korean		Coreano (Mac)
10004	x mac-arabic		Árabe (Mac)
10005	x-mac-hebrew		Hebraico (Mac)
10006	x-mac-greek		Grego (Mac)
10007	x-mac-cyrillic		Cirílico (Mac)

10008	x-mac-chinesesimp	Chinês simplificado (Mac)
10010	x-mac-romanian	Romeno (Mac)
10017	x-mac-ukrainian	Ucraniano (Mac)
10021	x-mac-thai	Tailandês (Mac)
10029	x-mac-ce	Europeu Central (Mac)
10079	x-mac-icelandic	Islandês (Mac)
10081	x-mac-turkish	Turco (Mac)
10082	x-mac-croatian	Croata (Mac)
12000	utf-32	Unicode (UTF-32)
12001	utf-32BE	Unicode (UTF-32-Big Endian)
20000	x-Chinese-CNS	Chinês tradicional (CNS)
20001	x-cp20001	TCA Taiwan
20002	x-Chinese-Eten	Chinês tradicional (Eten)
20003	x-cp20003	IBM5550 Taiwan
20004	x-cp20004	TeleText Taiwan
20005	x-cp20005	Wang Taiwan
20105	x-IA5	Europeu Ocidental (IA5)
20106	x-IA5-German	Alemão (IA5)
20107	x-IA5-Swedish	Sueco (IA5)
20108	x-IA5-Norwegian	Norueguês (IA5)
20127	us-ascii	US-ASCII

20261	x-cp20261	T.61
20269	x-cp20269	ISO-6937
20273	IBM273	IBM EBCDIC (Alemanha)
20277	IBM277	IBM EBCDIC (Dinamarca-Noruega)
20278	IBM278	IBM EBCDIC (Finlândia-Suécia)
20280	IBM280	IBM EBCDIC (Itália)
20284	IBM284	IBM EBCDIC (Espanha)
20285	IBM285	IBM EBCDIC (Grã Bretanha)
20290	IBM290	IBM EBCDIC (Japonês Katakana)
20297	IBM297	IBM EBCDIC (França)
20420	IBM420	IBM EBCDIC (Árabe)
20423	IBM423	IBM EBCDIC (Grego)
20424	IBM424	IBM EBCDIC (Hebraico)
20833	x-EBCDIC-KoreanExtended	IBM EBCDIC (Coreano, estendido)
20838	IBM-Thai	IBM EBCDIC (Tailandês)
20866	koi8-r	Cirílico (KOI8-R)
20871	IBM871	IBM EBCDIC (Islandês)
20880	IBM880	IBM EBCDIC (Cirílico, Russo)
20905	IBM905	IBM EBCDIC (Turco)
20924	IBM00924	IBM Latino 1



20932	EUC-JP	Japonês (JIS 0208-1990 e 0212-1990)
20936	x-cp20936	GB2312-80 Chinês (simples)
20949	x-cp20949	Coreano Wansung
21025	cp1025	IBM EBCDIC (Cirílico, Servo-Búlgaro)
21866	koi8-u	Cirílico (KOI8-U)
28591	iso-8859-1	Europeu Ocidental (ISO)
28592	iso-8859-2	Europeu Central (ISO)
28593	iso-8859-3	Latino 3 (ISO)
28594	iso-8859-4	Báltico (ISO)
28595	iso-8859-5	Cirílico (ISO)
28596	iso-8859-6	Árabe (ISO)
28597	iso-8859-7	Grego (ISO)
28598	iso-8859-8	Hebraico (ISO-Visual)
28599	iso-8859-9	Turco (ISO)
28603	iso-8859-13	Estoniano (ISO)
28605	iso-8859-15	Latino 9 (ISO)
29001	x-Europa	Europa
38598	iso-8859-8-i	Hebraico (ISO-Lógico)
50220	iso-2022-jp	Japonês (JIS)
50221	csISO2022JP	Japonês (JIS, 1 Byte Kana permitido)

50222	iso-2022-jp	Japonês (JIS, 1 Byte Kana permitido -SO/SI)
50225	iso-2022-kr	Coreano (ISO)
50227	x-cp50227	ISO-2022 Chinês (simplificado)
51932	euc-jp	Japonês (EUC)
51936	EUC-CN	Chinês simplificado (EUC)
51949	euc-kr	Coreano (EUC)
52936	hz-gb-2312	Chinês simplificado (HZ)
54936	GB18030	GB18030 Chinês simplificado
57002	x-iscii-de	ISCII Devanágari
57003	x-iscii-be	ISCII Bengalês
57004	x-iscii-ta	ISCII Tâmil
57005	x-iscii-te	ISCII Télego
57006	x-iscii-as	ISCII Assamês
57007	x-iscii-or	ISCII Oriá
57008	x-iscii-ka	ISCII Kannada
57009	x-iscii-ma	ISCII Malaiala
57010	x-iscii-gu	ISCII Guzerate
57011	x-iscii-pa	ISCII Punjabi
65000	utf-7	Unicode (UTF-7)
65001	utf-8	Unicode (UTF-8)

**Valor de retorno**

Uma string contendo todas as linhas do arquivo.

***FileSize***

Retorna o tamanho de um arquivo.

**Sintaxe**

FileSize(string fileName)

**Parâmetro**

Parâmetro	Significado
fileName	Nome do arquivo.

**Valor de retorno**

Tamanho do arquivo.

***GetCommandLine***

Retorna a linha de comando do processo.

**Sintaxe**

GetCommandLine()

**Parâmetro**

-

**Valor de retorno**

String contendo todos os argumentos da linha de comando.

***GetCommandLineArgByName***

Retorna o argumento de linha de comando especificado.

**Sintaxe**

GetCommandLineArgByName(string name)

**Parâmetro**

Parâmetro	Significado
name	Nome do argumento de linha de comando.

**Valor de retorno**

Valor do argumento da linha de comando, ou uma string vazia se o parâmetro não existir.

***GetCommandLineArgs***

Retorna uma string separada por vírgulas dos argumentos da linha de comando do processo atual.

**Sintaxe**

GetCommandLineArgs()

**Parâmetro**

-

**Valor de retorno**

String separada por vírgulas dos argumentos da linha de comando, ou uma string vazia.

***GetCurrentDirectory***

Obtém o diretório de trabalho atual do aplicativo.

**Sintaxe**

GetCurrentDirectory()

**Parâmetro**

-

**Valor de retorno**

String contendo o caminho do diretório de trabalho atual.

***GetDirectories***

Retorna os nomes dos subdiretórios em um diretório especificado como uma string separada por vírgulas.

**Sintaxe**

GetDirectories(string path)

GetDirectories(string path, string searchPattern)

GetDirectories(string path, string searchPattern, bool recursive)

**Parâmetro**

Parâmetro	Significado
path	O caminho onde os nomes dos subdiretórios serão obtidos.

searchPattern	A string de pesquisa para corresponder aos nomes dos arquivos no caminho.
recursive	True: pesquisar subdiretórios False: não pesquisar subdiretórios

**Valor de retorno**

String separada por vírgulas contendo os nomes de todos os subdiretórios.

***GetDirectoryName***

Retorna as informações de diretório para uma string de caminho especificada.

**Sintaxe**

GetDirectoryName(string path)

**Parâmetro**

Parâmetro	Significado
path	Caminho de um arquivo ou diretório.

**Valor de retorno**

String contendo as informações de diretório para um caminho ou uma string vazia, se o caminho identificar um diretório raiz.

String vazia se o caminho não tiver informações de diretório.

***GetDirectoryRoot***

Retorna informações relacionadas ao volume, raiz ou ambos para o caminho especificado.

**Sintaxe**

GetDirectoryRoot(string path)

**Parâmetro**

Parâmetro	Significado
path	Caminho de um arquivo ou diretório.

**Valor de retorno**

String contendo informações relacionadas ao volume, raiz ou ambos para o caminho especificado.

***GetEnvironmentVariable***

Retorna o valor de uma variável de ambiente.

### Sintaxe

GetEnvironmentVariable(string variable)

### Parâmetro

Parâmetro	Significado
Variable	Nome da variável de ambiente

### Valor de retorno

O valor da variável de ambiente especificado como variável, ou uma string vazia se a variável de ambiente não for encontrada.

### *GetExtension*

Retorna a extensão do arquivo da string de caminho especificado.

### Sintaxe

GetExtension(string path)

### Parâmetro

Parâmetro	Significado
path	String do caminho da qual a extensão será lida.

### Valor de retorno

String contendo a extensão do caminho especificado (incluindo ".").

### *GetFileName*

Retorna o nome do arquivo e extensão da string de caminho especificado.

### Sintaxe

GetFileName(string path)

### Parâmetro

Parâmetro	Significado
path	A string de caminho onde o nome e extensão do arquivo serão obtidos.

### Valor de retorno

String contendo os caracteres após o último caractere de diretório no caminho.

Se o último caractere do caminho for um caractere separador de diretório ou volume, esse método retorna uma string vazia.

### ***GetFileNameWithoutExtension***

Retorna o nome do arquivo da string do caminho especificado sem a extensão.

#### **Sintaxe**

GetFileNameWithoutExtension(string path)

#### **Parâmetro**

Parâmetro	Significado
path	caminho do arquivo.

#### **Valor de retorno**

String contendo o resultado de GetFileName sem o ponto final (.) e todos os caracteres seguintes.

### ***GetFiles***

Retorna os nomes dos arquivos em um diretório especificado como uma string separada por vírgulas.

Usar o parâmetro **searchPattern** retorna os nomes dos arquivos que correspondem ao padrão de pesquisa especificado no diretório especificado.

Definir o parâmetro 'recursive' determina se os subdiretórios serão incluídos na pesquisa.

#### **Observação:**

Se o diretório atual e todos os subdiretórios forem pesquisados (recursive = true) e a estrutura do diretório contiver um link que cria um loop de volta para dentro da estrutura do diretório, a pesquisa resultará em um loop infinito.

#### **Sintaxe**

GetFiles(string path)

GetFiles(string path, string searchPattern)

GetFiles(string path, string searchPattern, bool recursive)

#### **Parâmetro**

Parâmetro	Significado
path	O diretório do qual os arquivos serão obtidos.
searchPattern	A string de pesquisa para corresponder aos nomes dos arquivos no caminho. O parâmetro searchPattern não pode terminar em dois pontos ("..") ou conter dois

	pontos ("..")
recursive	Define se apenas o diretório atual será pesquisado (=false) ou o diretório atual e todos os subdiretórios serão pesquisados (=true).

**Valor de retorno**

Uma string de nomes de arquivos separados por vírgulas no diretório especificado.

Definir o parâmetro 'searchPattern' retorna os nomes dos arquivos que correspondem ao padrão de pesquisa especificado no diretório especificado.

Nomes de arquivos contêm o caminho completo.

***GetFileSystemEntries***

Retorna o nome de todos os arquivos e subdiretórios em um diretório especificado em uma string separada por vírgulas.

**Sintaxe**

```
GetFileSystemEntries(string path)
```

```
GetFileSystemEntries(string path, string searchPattern)
```

**Parâmetro**

Parâmetro	Significado
path	O diretório do qual os nomes dos arquivos e diretórios serão retornados.
searchPattern	A string de pesquisa para corresponder aos nomes dos arquivos no caminho. O parâmetro searchPattern não pode terminar em dois pontos ("..") ou conter dois pontos ("..")

**Valor de retorno**

Uma string separada por vírgulas contendo os nomes das entradas do sistema do arquivo encontrados no diretório especificado e correspondente ao critério de pesquisa especificado em **searchPattern**.

***GetFullPath***

Retorna o caminho absoluto para uma string de caminho especificada.

**Sintaxe**

```
GetFullPath(string path)
```

**Parâmetro**

Parâmetro	Significado
-----------	-------------



path	Arquivo ou diretório para o qual a informação de caminho será recuperada.
------	---

**Valor de retorno**

String contendo o caminho absoluto ou variável, por exemplo "C:\MyFile.txt".

***GetLogicalDrives***

Retorna o nome das unidades lógicas desse computador no formato "<DriveLetter>:\".

**Sintaxe**

GetLogicalDrives()

**Parâmetro**

-

**Valor de retorno**

As unidades lógicas desse computador.

***GetParentDirectory***

Recupera o diretório pai, incluindo os caminhos absoluto e relativo.

**Sintaxe**

GetParentDirectory(string path)

**Parâmetro**

Parâmetro	Significado
path	O caminho do qual o diretório pai será recuperado.

**Valor de retorno**

O diretório pai ou uma string vazia quando o caminho não está no diretório raiz, incluindo a raiz de um servidor UNC ou nome de compartilhamento.

***GetPathRoot***

Retorna as informações do diretório raiz do caminho especificado.

**Sintaxe**

GetPathRoot(string path)

**Parâmetro**

Parâmetro	Significado
path	Caminho do qual as informações do diretório raiz serão recuperadas.

**Valor de retorno**

Uma string contendo o diretório raiz do caminho, por exemplo, "C:\\" ou uma string vazia quando o caminho não contém informações sobre o diretório raiz.

***GetRandomFileName***

Retorna um arquivo aleatório ou um nome de diretório.

**Observação:**

O método retorna uma string com criptografia forte e aleatória que pode ser usada como nome de pasta ou de arquivo.

Diferente de `GetTempFileName`, o método `GetRandomFileName` não cria um arquivo.

Quando a segurança de seu sistema de arquivos é primordial, esse método deve ser usado ao invés de `GetTempFileName`.

**Sintaxe**

```
GetRandomFileName()
```

**Parâmetro**

-

**Valor de retorno**

Arquivo aleatório ou um nome de diretório.

***GetTempFileName***

Cria um arquivo temporário com nome exclusivo de 0 (zero) byte de tamanho no dispositivo de armazenamento de dados e retorna o caminho completo para esse arquivo.

**Observação:**

Esse método cria um arquivo temporário com a extensão `.tmp`.

**Sintaxe**

```
GetTempFileName()
```

**Parâmetro**

-

**Valor de retorno**

String contendo o caminho completo para o arquivo temporário.

***GetTempPath***

Retorna o caminho para o diretório temporário do sistema atual.

**Sintaxe**

GetTempPath()

**Parâmetro**

-

**Valor de retorno**

String contendo o caminho para um diretório temporário.

***HasExtension***

Determina se um caminho contém uma extensão de nome de arquivo.

**Sintaxe**

HasExtension(string path)

**Parâmetro**

Parâmetro	Significado
path	O caminho no qual a extensão será procurada.

**Valor de retorno**

True, quando o caractere que segue o separador do último diretório (\ ou /) ou caractere separador do volume (:) contém um ponto (.) ou um ou mais caracteres à direita. Em caso contrário, False

***IsPathRooted***

Retorna um valor que denota se a string de caminho contém um caminho relativo ou absoluto.

**Sintaxe**

IsPathRooted(string path)

**Parâmetro**

Parâmetro	Significado
path	Caminho a ser testado

**Valor de retorno**

True quando a string contém um caminho absoluto. Caso contrário, False.

## ***MoveDirectory***

Movimenta um arquivo ou diretório e seu conteúdo para um novo local de armazenamento.

### **Sintaxe**

`MoveDirectory(string sourceDirName, string destDirName)`

### **Parâmetro**

Parâmetro	Significado
sourceDirName	Caminho do arquivo ou diretório a ser movido.
destDirName	Caminho para o novo local de armazenamento de <b>sourceDirName</b> . Se <b>sourceDirName</b> for um arquivo, então <b>destDirName</b> também deve ser um nome de arquivo.

### **Valor de retorno**

-

## ***MoveFile***

Movimenta o arquivo especificado para um novo local de armazenamento. Também permite que você mude o nome do arquivo.

### **Sintaxe**

`MoveFile(string sourceFileName, string destFileName)`

### **Parâmetro**

Parâmetro	Significado
sourceFileName	Nome do arquivo a ser movido.
destFileName	Novo caminho para o arquivo

### **Valor de retorno**

-

## ***ReplaceFile***

Substitui o conteúdo de um arquivo especificado pelo conteúdo de outro arquivo, exclui o arquivo original e cria uma cópia de back-up do arquivo que foi substituído.

### **Sintaxe**

`ReplaceFile(string sourceFileName, string destinationFileName, string destinationBackupFileName)`

**Parâmetro**

Parâmetro	Significado
sourceFileName	O nome do arquivo que o arquivo definido como destinationFileName substituirá.
destinationFileName	O nome do arquivo que será substituído.
destinationBackupFileName	O nome do arquivo de back-up do arquivo que será substituído. String vazia se não será criado arquivo de back-up.

**Valor de retorno**

-

***SetCurrentDirectory***

Define o diretório de trabalho atual do aplicativo do diretório especificado.

**Sintaxe**

SetCurrentDirectory(string path)

**Parâmetro**

Parâmetro	Significado
path	Caminho do diretório de trabalho atual.

**Valor de retorno**

-

Propriedades da classe `MLDos`

***NewLine***

Obtém a string para uma quebra de linha definida para um ambiente específico.

***NewLineLF***

Retorna o LineFeed como uma string ("\n" - \$0A).

***NewLineCRLF***

Retorna a CarriageReturn/LineFeed (CRLF) como uma string ("\r\n" - \$0D \$0A).

Métodos da classe `MonaLisaTextFile`

***Acrescentar***

Abre um arquivo para o acréscimo de texto.

### Sintaxe

```
Append(string fileName)
```

```
Append(string fileName, string encoding)
```

### Parâmetro

Parâmetro	Significado																											
fileName	O caminho para o arquivo a ser acrescentado.																											
encoding	<p>O texto a seguir pode ser usado como parâmetro de codificação:</p> <ul style="list-style-type: none"> <li>String vazia. Nesse caso, a codificação padrão (página de código ANSI do sistema operacional) é usada.</li> <li>página de código</li> <li>name</li> <li>nome amigável</li> </ul> <p>Exemplo:</p> <ul style="list-style-type: none"> <li>"1252"</li> <li>"windows-1252"</li> <li>"Europeu Ocidental" (Windows)</li> </ul> <p>Possíveis parâmetros de codificação</p> <table border="1"> <thead> <tr> <th>Página de código</th> <th>Nome</th> <th>Nome para exibição</th> </tr> </thead> <tbody> <tr> <td>37</td> <td>IBM037</td> <td>IBM EBCDIC (EUA-Canadá)</td> </tr> <tr> <td>437</td> <td>IBM437</td> <td>OEM EUA</td> </tr> <tr> <td>500</td> <td>IBM500</td> <td>IBM EBCDIC (Internacional)</td> </tr> <tr> <td>708</td> <td>ASMO-708</td> <td>Árabe (ASMO 708)</td> </tr> <tr> <td>720</td> <td>DOS-720</td> <td>Árabe (DOS)</td> </tr> <tr> <td>737</td> <td>ibm737</td> <td>Grego (DOS)</td> </tr> <tr> <td>775</td> <td>ibm775</td> <td>Báltico (DOS)</td> </tr> <tr> <td>850</td> <td>ibm850</td> <td>Europeu Ocidental (DOS)</td> </tr> </tbody> </table>	Página de código	Nome	Nome para exibição	37	IBM037	IBM EBCDIC (EUA-Canadá)	437	IBM437	OEM EUA	500	IBM500	IBM EBCDIC (Internacional)	708	ASMO-708	Árabe (ASMO 708)	720	DOS-720	Árabe (DOS)	737	ibm737	Grego (DOS)	775	ibm775	Báltico (DOS)	850	ibm850	Europeu Ocidental (DOS)
Página de código	Nome	Nome para exibição																										
37	IBM037	IBM EBCDIC (EUA-Canadá)																										
437	IBM437	OEM EUA																										
500	IBM500	IBM EBCDIC (Internacional)																										
708	ASMO-708	Árabe (ASMO 708)																										
720	DOS-720	Árabe (DOS)																										
737	ibm737	Grego (DOS)																										
775	ibm775	Báltico (DOS)																										
850	ibm850	Europeu Ocidental (DOS)																										

852	ibm852	Europeu Oriental (DOS)
855	IBM855	OEM Cirílico
857	ibm857	Turco (DOS)
858	IBM00858	OEM Multilíngue Latino I
860	IBM860	Português (DOS)
861	ibm861	Islandês (DOS)
862	DOS-862	Hebraico (DOS)
863	IBM863	Francês do Canadá (DOS)
864	IBM864	Árabe (864)
865	IBM865	Nórdico (DOS)
866	cp866	Cirílico (DOS)
869	ibm869	Grego moderno (DOS)
870	IBM870	IBM EBCDIC (Multilíngue Latino 2)
874	windows-874	Tailandês (Windows)
875	cp875	IBM EBCDIC (Grego moderno)
932	shift_jis	Japonês (Shift-JIS)
936	gb2312	Chinês simplificado (GB2312)
949	ks_c_5601-1987	Coreano
950	big5	Chinês (tradicional) (Big5)
1026	IBM1026	IBM EBCDIC (Turco, Latino 5)

1047	IBM01047	IBM Latino 1
1140	IBM01140	IBM EBCDIC (EUA-Canadá-Europeu)
1141	IBM01141	IBM EBCDIC (Alemanha-Europeu)
1142	IBM01142	IBM EBCDIC (Dinamarca-Noruega-Europeu)
1143	IBM01143	IBM EBCDIC (Finlândia-Suécia-Europeu)
1144	IBM01144	IBM EBCDIC (Italiano-Europeu)
1145	IBM01145	IBM EBCDIC (Espanha-Europeu)
1146	IBM01146	IBM EBCDIC (Grã-Bretanha-Europeu)
1147	IBM01147	IBM EBCDIC (França-Europeu)
1148	IBM01148	IBM EBCDIC (Internacional-Europeu)
1149	IBM01149	IBM EBCDIC (Islandês-Europeu)
1200	utf-16	Unicode
1201	unicodeFFFE	Unicode (Big Endian)
1250	windows-1250	Centro-europeu (Windows)
1251	windows-1251	Cirílico (Windows)
1252	Windows-1252	Europeu Ocidental (Windows)
1253	windows-1253	Grego (Windows)



1254	windows-1254	Turco (Windows)
1255	windows-1255	Hebraico (Windows)
1256	windows-1256	Árabe (Windows)
1257	windows-1257	Báltico (Windows)
1258	windows-1258	Vietnamita (Windows)
1361	Johab	Coreano (Johab)
10000	macintosh	Europeu Ocidental (Mac)
10001	x-mac-japanese	Japonês (Mac)
10002	x-mac-chinesetrad	Chinês tradicional (Mac)
10003	x-mac-korean	Coreano (Mac)
10004	x mac-arabic	Árabe (Mac)
10005	x-mac-hebrew	Hebraico (Mac)
10006	x-mac-greek	Grego (Mac)
10007	x-mac-cyrillic	Cirílico (Mac)
10008	x-mac-chinesesimp	Chinês simplificado (Mac)
10010	x-mac-romanian	Romeno (Mac)
10017	x-mac-ukrainian	Ucraniano (Mac)
10021	x-mac-thai	Tailandês (Mac)
10029	x-mac-ce	Centro-europeu (Mac)
10079	x-mac-icelandic	Islandês (Mac)
10081	x-mac-turkish	Turco (Mac)

10082	x-mac-croatian	Croata (Mac)
12000	utf-32	Unicode (UTF-32)
12001	utf-32BE	Unicode (UTF-32-Big Endian)
20000	x-Chinese-CNS	Chinês tradicional (CNS)
20001	x-cp20001	TCA Taiwan
20002	x-Chinese-Eten	Chinês tradicional (Eten)
20003	x-cp20003	IBM5550 Taiwan
20004	x-cp20004	TeleText Taiwan
20005	x-cp20005	Wang Taiwan
20105	x-IA5	Europeu Ocidental (IA5)
20106	x-IA5-German	Alemão (IA5)
20107	x-IA5-Swedish	Sueco (IA5)
20108	x-IA5-Norwegian	Norueguês (IA5)
20127	us-ascii	US-ASCII
20261	x-cp20261	T.61
20269	x-cp20269	ISO-6937
20273	IBM273	IBM EBCDIC (Alemanha)
20277	IBM277	IBM EBCDIC (Dinamarca-Noruega)
20278	IBM278	IBM EBCDIC (Finlândia-Suécia)
20280	IBM280	IBM EBCDIC (Itália)
20284	IBM284	IBM EBCDIC (Espanha)

20285	IBM285	IBM EBCDIC (Grã-Bretanha)
20290	IBM290	IBM EBCDIC (Japonês katakana)
20297	IBM297	IBM EBCDIC (França)
20420	IBM420	IBM EBCDIC (Árabe)
20423	IBM423	IBM EBCDIC (Grego)
20424	IBM424	IBM EBCDIC (Hebraico)
20833	x-EBCDIC-KoreanExtended	IBM EBCDIC (Coreano estendido)
20838	IBM-Thai	IBM EBCDIC (Tailandês)
20866	koi8-r	Cirílico (KOI8-R)
20871	IBM871	IBM EBCDIC (Islandês)
20880	IBM880	IBM EBCDIC (Cirílico russo)
20905	IBM905	IBM EBCDIC (Turco)
20924	IBM00924	IBM Latino 1
20932	EUC-JP	Japonês (JIS 0208-1990 e 0212-1990)
20936	x-cp20936	GB2312-80 Chinês (simplificado)
20949	x-cp20949	Coreano Wansung
21025	cp1025	IBM EBCDIC (Cirílico servo-búlgaro)
21866	koi8-u	Cirílico (KOI8-U)
28591	iso-8859-1	Europeu Ocidental (ISO)
28592	iso-8859-2	Centro-europeu (ISO)

28593	iso-8859-3	Latino 3 (ISO)
28594	iso-8859-4	Báltico (ISO)
28595	iso-8859-5	Cirílico (ISO)
28596	iso-8859-6	Árabe (ISO)
28597	iso-8859-7	Grego (ISO)
28598	iso-8859-8	Hebraico (ISO-Visual)
28599	iso-8859-9	Turco (ISO)
28603	iso-8859-13	Estoniano (ISO)
28605	iso-8859-15	Latino 9 (ISO)
29001	x-Europa	Europa
38598	iso-8859-8-i	Hebraico (ISO-lógico)
50220	iso-2022-jp	Japonês (JIS)
50221	csISO2022JP	Japonês (JIS, 1 byte kana permitido)
50222	iso-2022-jp	Japonês (JIS, 1 byte kana permitido-SO/SI)
50225	iso-2022-kr	Coreano (ISO)
50227	x-cp50227	ISO-2022 Chinês simplificado
51932	euc-jp	Japonês (EUC)
51936	EUC-CN	Chinês simplificado (EUC)
51949	euc-kr	Coreano (EUC)
52936	hz-gb-2312	Chinês simplificado (HZ)

54936	GB18030	GB18030 Chinês simplificado
57002	x-iscii-de	ISCII Devanágari
57003	x-iscii-be	ISCII Bengalês
57004	x-iscii-ta	ISCII Tâmil
57005	x-iscii-te	ISCII Télego
57006	x-iscii-as	ISCII Assamês
57007	x-iscii-or	ISCII Oriá
57008	x-iscii-ka	ISCII Kannada
57009	x-iscii-ma	ISCII Malaiala
57010	x-iscii-gu	ISCII Guzerate
57011	x-iscii-pa	ISCII Punjabi
65000	utf-7	Unicode (UTF-7)
65001	utf-8	Unicode (UTF-8)

**Valor de retorno**

-

***Fechar***

Fecha um arquivo aberto.

**Sintaxe**

Close()

**Parâmetro**

-

**Valor de retorno**

-

**Criar**

Cria ou abre um arquivo para a escrita de texto.

**Sintaxe**

```
Create(string fileName)
```

```
Create(string fileName, string encoding)
```

**Parâmetro**

Parâmetro	Significado																					
fileName	O arquivo a ser aberto.																					
encoding	<p>Os seguintes podem ser usados como parâmetro de codificação:</p> <ul style="list-style-type: none"> <li>• Cadeia de caracteres vazia. Nesse caso, a codificação padrão (página de código ANSI do sistema operacional) é usada.</li> <li>• página de código</li> <li>• nome</li> <li>• Nome amigável</li> </ul> <p>Exemplo:</p> <ul style="list-style-type: none"> <li>• "1252"</li> <li>• "windows-1252"</li> <li>• "Europeu Ocidental" (Windows)</li> </ul> <p>Parâmetros de codificação possíveis</p> <table border="1"> <thead> <tr> <th>Página de código</th> <th>Nome</th> <th>Nome de exibição</th> </tr> </thead> <tbody> <tr> <td>37</td> <td>IBM037</td> <td>IBM EBCDIC (EUA-Canadá)</td> </tr> <tr> <td>437</td> <td>IBM437</td> <td>OEM EUA</td> </tr> <tr> <td>500</td> <td>IBM500</td> <td>IBM EBCDIC (Internacional)</td> </tr> <tr> <td>708</td> <td>ASMO-708</td> <td>Árabe (ASMO 708)</td> </tr> <tr> <td>720</td> <td>DOS-720</td> <td>Árabe (DOS)</td> </tr> <tr> <td>737</td> <td>ibm737</td> <td>Grego (DOS)</td> </tr> </tbody> </table>	Página de código	Nome	Nome de exibição	37	IBM037	IBM EBCDIC (EUA-Canadá)	437	IBM437	OEM EUA	500	IBM500	IBM EBCDIC (Internacional)	708	ASMO-708	Árabe (ASMO 708)	720	DOS-720	Árabe (DOS)	737	ibm737	Grego (DOS)
Página de código	Nome	Nome de exibição																				
37	IBM037	IBM EBCDIC (EUA-Canadá)																				
437	IBM437	OEM EUA																				
500	IBM500	IBM EBCDIC (Internacional)																				
708	ASMO-708	Árabe (ASMO 708)																				
720	DOS-720	Árabe (DOS)																				
737	ibm737	Grego (DOS)																				

775	ibm775	Báltico (DOS)
850	ibm850	Europeu Ocidental (DOS)
852	ibm852	Europeu Oriental (DOS)
855	IBM855	OEM Cirílico
857	ibm857	Turco (DOS)
858	IBM00858	OEM Latino Multilingue I
860	IBM860	Português (DOS)
861	ibm861	Islandês (DOS)
862	DOS-862	Hebraico (DOS)
863	IBM863	Francês, Canadá (DOS)
864	IBM864	Árabe (864)
865	IBM865	Nórdico (DOS)
866	cp866	Cirílico (DOS)
869	ibm869	Grego, moderno (DOS)
870	IBM870	IBM EBCDIC (Latino Multilingue-2)
874	windows-874	Tailandês (Windows)
875	cp875	IBM EBCDIC (Grego, moderno)
932	shift_jis	Japonês (Shift-JIS)
936	gb2312	Chinês simplificado (GB2312)
949	ks_c_5601-1987	Coreano

950	big5	Chinês (Tradicional) (Big5)
1026	IBM1026	IBM EBCDIC (Turco, Latino-5)
1047	IBM01047	IBM Latino-1
1140	IBM01140	IBM EBCDIC (EUA-Canadá-Europeu)
1141	IBM01141	IBM EBCDIC (Alemanha-Europeu)
1142	IBM01142	IBM EBCDIC (Dinamarca-Noruega-Europeu)
1143	IBM01143	IBM EBCDIC (Finlândia-Suécia-Europeu)
1144	IBM01144	IBM EBCDIC (Italiano-Europeu)
1145	IBM01145	IBM EBCDIC (Espanha-Europeu)
1146	IBM01146	IBM EBCDIC (Grã Bretanha-Europeu)
1147	IBM01147	IBM EBCDIC (França-Europeu)
1148	IBM01148	IBM EBCDIC (Internacional-Europeu)
1149	IBM01149	IBM EBCDIC (Islandês-Europeu)
1200	utf-16	Unicode
1201	unicodeFFFE	Unicode (Big Endian)
1250	windows-1250	Europeu Central (Windows)
1251	windows-1251	Cirílico (Windows)



1252	Windows-1252	Europeu Ocidental (Windows)
1253	windows-1253	Grego (Windows)
1254	windows-1254	Turco (Windows)
1255	windows-1255	Hebraico (Windows)
1256	windows-1256	Árabe (Windows)
1257	windows-1257	Báltico (Windows)
1258	windows-1258	Vietnamita (Windows)
1361	Johab	Coreano (Johab)
10000	macintosh	Europeu Ocidental (Mac)
10001	x-mac-japanese	Japonês (Mac)
10002	x-mac-chinesetrad	Chinês tradicional (Mac)
10003	x-mac-korean	Coreano (Mac)
10004	x mac-arabic	Árabe (Mac)
10005	x-mac-hebrew	Hebraico (Mac)
10006	x-mac-greek	Grego (Mac)
10007	x-mac-cyrillic	Cirílico (Mac)
10008	x-mac-chinesesimp	Chinês simplificado (Mac)
10010	x-mac-romanian	Romeno (Mac)
10017	x-mac-ukrainian	Ucraniano (Mac)
10021	x-mac-thai	Tailandês (Mac)

10029	x-mac-ce	Europeu Central (Mac)
10079	x-mac-icelandic	Islandês (Mac)
10081	x-mac-turkish	Turco (Mac)
10082	x-mac-croatian	Croata (Mac)
12000	utf-32	Unicode (UTF-32)
12001	utf-32BE	Unicode (UTF-32-Big Endian)
20000	x-Chinese-CNS	Chinês tradicional (CNS)
20001	x-cp20001	TCA Taiwan
20002	x-Chinese-Eten	Chinês tradicional (Eten)
20003	x-cp20003	IBM5550 Taiwan
20004	x-cp20004	TeleText Taiwan
20005	x-cp20005	Wang Taiwan
20105	x-IA5	Europeu Ocidental (IA5)
20106	x-IA5-German	Alemão (IA5)
20107	x-IA5-Swedish	Sueco (IA5)
20108	x-IA5-Norwegian	Norueguês (IA5)
20127	us-ascii	US-ASCII
20261	x-cp20261	T.61
20269	x-cp20269	ISO-6937
20273	IBM273	IBM EBCDIC (Alemanha)
20277	IBM277	IBM EBCDIC (Dinamarca-Noruega)

20278	IBM278	IBM EBCDIC (Finlândia-Suécia)
20280	IBM280	IBM EBCDIC (Itália)
20284	IBM284	IBM EBCDIC (Espanha)
20285	IBM285	IBM EBCDIC (Grã Bretanha)
20290	IBM290	IBM EBCDIC (japonês Katakana)
20297	IBM297	IBM EBCDIC (França)
20420	IBM420	IBM EBCDIC (Árabe)
20423	IBM423	IBM EBCDIC (Grego)
20424	IBM424	IBM EBCDIC (Hebraico)
20833	x-EBCDIC-KoreanExtended	IBM EBCDIC (Coreano, estendido)
20838	IBM-Thai	IBM EBCDIC (Tailandês)
20866	koi8-r	Cirílico (KOI8-R)
20871	IBM871	IBM EBCDIC (Islandês)
20880	IBM880	IBM EBCDIC (Cirílico, Russo)
20905	IBM905	IBM EBCDIC (Turco)
20924	IBM00924	IBM Latino-1
20932	EUC-JP	Japonês (JIS 0208-1990 e 0212-1990)
20936	x-cp20936	GB2312-80 Chinês (simples)
20949	x-cp20949	Coreano Wansung
21025	cp1025	IBM EBCDIC (Cirílico, Servo-Búlgaro)

21866	koi8-u	Cirílico (KOI8-U)
28591	iso-8859-1	Europeu Ocidental (ISO)
28592	iso-8859-2	Europeu Central (ISO)
28593	iso-8859-3	Latino 3 (ISO)
28594	iso-8859-4	Báltico (ISO)
28595	iso-8859-5	Cirílico (ISO)
28596	iso-8859-6	Árabe (ISO)
28597	iso-8859-7	Grego (ISO)
28598	iso-8859-8	Hebraico (ISO-Visual)
28599	iso-8859-9	Turco (ISO)
28603	iso-8859-13	Estoniano (ISO)
28605	iso-8859-15	Latino 9 (ISO)
29001	x-Europa	Europa
38598	iso-8859-8-i	Hebraico (ISO-Lógico)
50220	iso-2022-jp	Japonês (JIS)
50221	csISO2022JP	Japonês (JIS, 1 Byte Kana permitido)
50222	iso-2022-jp	Japonês (JIS, 1 Byte Kana permitido -SO/SI)
50225	iso-2022-kr	Coreano (ISO)
50227	x-cp50227	ISO-2022 Chinês (simplificado)
51932	euc-jp	Japonês (EUC)

51936	EUC-CN	Chinês simplificado (EUC)
51949	euc-kr	Coreano (EUC)
52936	hz-gb-2312	Chinês simplificado (HZ)
54936	GB18030	GB18030 Chinês simplificado
57002	x-iscii-de	ISCII Devanágari
57003	x-iscii-be	ISCII Bengalês
57004	x-iscii-ta	ISCII Tâmil
57005	x-iscii-te	ISCII Télego
57006	x-iscii-as	ISCII Assamês
57007	x-iscii-or	ISCII Oriá
57008	x-iscii-ka	ISCII Kannada
57009	x-iscii-ma	ISCII Malaiala
57010	x-iscii-gu	ISCII Guzerate
57011	x-iscii-pa	ISCII Punjabi
65000	utf-7	Unicode (UTF-7)
65001	utf-8	Unicode (UTF-8)

**Valor de retorno**

-

***Eof***

Retorna um valor denotando se a posição atual é o fim do arquivo.

**Sintaxe**

Eof():

**Parâmetro**

-

**Valor de retorno**

True quando a posição é o fim do arquivo. Em caso contrário, false.

***Flush***

Limpa todos os buffers e os grava no arquivo.

**Sintaxe**

Flush()

**Parâmetro**

-

**Valor de retorno**

-

***Abrir***

Abre um arquivo para ler um texto.

**Sintaxe**

Open(string fileName)

Open(string fileName, string encoding)

**Parâmetro**

Parâmetro	Significado
fileName	O arquivo para abrir para fazer a leitura
encoding	Os seguintes podem ser usados como parâmetro de codificação: <ul style="list-style-type: none"> <li>• String vazia. Nesse caso, a codificação padrão (página de código ANSI do sistema operacional) é usada.</li> <li>• página de código</li> <li>• nome</li> <li>• nome amigável</li> </ul> Exemplo: <ul style="list-style-type: none"> <li>• "1252"</li> <li>• "windows-1252"</li> <li>• "Western European" (Windows)</li> </ul>

## Parâmetros de codificação possíveis

Página de código	Nome	Nome de exibição
37	IBM037	IBM EBCDIC (EUA-Canadá)
437	IBM437	OEM EUA
500	IBM500	IBM EBCDIC (Internacional)
708	ASMO-708	Árabe (ASMO 708)
720	DOS-720	Árabe (DOS)
737	ibm737	Grego (DOS)
775	ibm775	Báltico (DOS)
850	ibm850	Europeu Ocidental (DOS)
852	ibm852	Europeu Oriental (DOS)
855	IBM855	OEM Cirílico
857	ibm857	Turco (DOS)
858	IBM00858	OEM Multilíngue Latino I
860	IBM860	Português (DOS)
861	ibm861	Islandês (DOS)
862	DOS-862	Hebraico (DOS)
863	IBM863	Francês do Canadá (DOS)
864	IBM864	Árabe (864)
865	IBM865	Nórdico (DOS)
866	cp866	Cirílico (DOS)
869	ibm869	Grego moderno (DOS)

870	IBM870	IBM EBCDIC (Latino Multilíngue-2)
874	windows-874	Tailandês (Windows)
875	cp875	IBM EBCDIC (Grego moderno)
932	shift_jis	Japonês (Shift-JIS)
936	gb2312	Chinês simplificado (GB2312)
949	ks_c_5601-1987	Coreano
950	big5	Chinês (tradicional) (Big5)
1026	IBM1026	IBM EBCDIC (Turco, Latino-5)
1047	IBM01047	IBM Latino-1
1140	IBM01140	IBM EBCDIC (EUA-Canadá-Europeu)
1141	IBM01141	IBM EBCDIC (Alemanha-Europeu)
1142	IBM01142	IBM EBCDIC (Dinamarca-Noruega-Europeu)
1143	IBM01143	IBM EBCDIC (Finlândia-Suécia-Europeu)
1144	IBM01144	IBM EBCDIC (Italiano-Europeu)
1145	IBM01145	IBM EBCDIC (Espanha-Europeu)
1146	IBM01146	IBM EBCDIC (Grã Bretanha-Europeu)
1147	IBM01147	IBM EBCDIC (França-Europeu)



1148	IBM01148	IBM EBCDIC (Internacional-Europeu)
1149	IBM01149	IBM EBCDIC (Islandês-Europeu)
1200	utf-16	Unicode
1201	unicodeFFFE	Unicode (Big Endian)
1250	windows-1250	Europeu Central (Windows)
1251	windows-1251	Cirílico (Windows)
1252	Windows-1252	Europeu Ocidental (Windows)
1253	windows-1253	Grego (Windows)
1254	windows-1254	Turco (Windows)
1255	windows-1255	Hebraico (Windows)
1256	windows-1256	Árabe (Windows)
1257	windows-1257	Báltico (Windows)
1258	windows-1258	Vietnamita (Windows)
1361	Johab	Coreano (Johab)
10000	macintosh	Europeu Ocidental (Mac)
10001	x-mac-japanese	Japonês (Mac)
10002	x-mac-chinesetrad	Chinês tradicional (Mac)
10003	x-mac-korean	Coreano (Mac)
10004	x mac-arabic	Árabe (Mac)
10005	x-mac-hebrew	Hebraico (Mac)

10006	x-mac-greek	Grego (Mac)
10007	x-mac-cyrillic	Cirílico (Mac)
10008	x-mac-chinesesimp	Chinês simplificado (Mac)
10010	x-mac-romanian	Romeno (Mac)
10017	x-mac-ukrainian	Ucraniano (Mac)
10021	x-mac-thai	Tailandês (Mac)
10029	x-mac-ce	Europeu Central (Mac)
10079	x-mac-icelandic	Islandês (Mac)
10081	x-mac-turkish	Turco (Mac)
10082	x-mac-croatian	Croata (Mac)
12000	utf-32	Unicode (UTF-32)
12001	utf-32BE	Unicode (UTF-32-Big Endian)
20000	x-Chinese-CNS	Chinês tradicional (CNS)
20001	x-cp20001	TCA Taiwan
20002	x-Chinese-Eten	Chinês tradicional (Eten)
20003	x-cp20003	IBM5550 Taiwan
20004	x-cp20004	TeleText Taiwan
20005	x-cp20005	Wang Taiwan
20105	x-IA5	Europeu Ocidental (IA5)
20106	x-IA5-German	Alemão (IA5)
20107	x-IA5-Swedish	Sueco (IA5)

20108	x-IA5-Norwegian	Norueguês (IA5)
20127	us-ascii	US-ASCII
20261	x-cp20261	T.61
20269	x-cp20269	ISO-6937
20273	IBM273	IBM EBCDIC (Alemanha)
20277	IBM277	IBM EBCDIC (Dinamarca-Noruega)
20278	IBM278	IBM EBCDIC (Finlândia-Suécia)
20280	IBM280	IBM EBCDIC (Itália)
20284	IBM284	IBM EBCDIC (Espanha)
20285	IBM285	IBM EBCDIC (Grã Bretanha)
20290	IBM290	IBM EBCDIC (Japonês Katakana)
20297	IBM297	IBM EBCDIC (França)
20420	IBM420	IBM EBCDIC (Árabe)
20423	IBM423	IBM EBCDIC (Grego)
20424	IBM424	IBM EBCDIC (Hebraico)
20833	x-EBCDIC-KoreanExtended	IBM EBCDIC (Coreano, estendido)
20838	IBM-Thai	IBM EBCDIC (Tailandês)
20866	koi8-r	Cirílico (KOI8-R)
20871	IBM871	IBM EBCDIC (Islandês)
20880	IBM880	IBM EBCDIC (Cirílico, Russo)

20905	IBM905	IBM EBCDIC (Turco)
20924	IBM00924	IBM Latino-1
20932	EUC-JP	Japonês (JIS 0208-1990 e 0212-1990)
20936	x-cp20936	GB2312-80 Chinês (simples)
20949	x-cp20949	Coreano Wansung
21025	cp1025	IBM EBCDIC (Cirílico, Servo-Búlgaro)
21866	koi8-u	Cirílico (KOI8-U)
28591	iso-8859-1	Europeu Ocidental (ISO)
28592	iso-8859-2	Europeu Central (ISO)
28593	iso-8859-3	Latino 3 (ISO)
28594	iso-8859-4	Báltico (ISO)
28595	iso-8859-5	Cirílico (ISO)
28596	iso-8859-6	Árabe (ISO)
28597	iso-8859-7	Grego (ISO)
28598	iso-8859-8	Hebraico (ISO-Visual)
28599	iso-8859-9	Turco (ISO)
28603	iso-8859-13	Estoniano (ISO)
28605	iso-8859-15	Latino 9 (ISO)
29001	x-Europa	Europa
38598	iso-8859-8-i	Hebraico (ISO-Lógico)
50220	iso-2022-jp	Japonês (JIS)

50221	csISO2022JP	Japonês (JIS, 1 Byte Kana permitido)
50222	iso-2022-jp	Japonês (JIS, 1 Byte Kana permitido -SO/SI)
50225	iso-2022-kr	Coreano (ISO)
50227	x-cp50227	ISO-2022 Chinês (simplificado)
51932	euc-jp	Japonês (EUC)
51936	EUC-CN	Chinês simplificado (EUC)
51949	euc-kr	Coreano (EUC)
52936	hz-gb-2312	Chinês simplificado (HZ)
54936	GB18030	GB18030 Chinês simplificado
57002	x-iscii-de	ISCII Devanágari
57003	x-iscii-be	ISCII Bengalês
57004	x-iscii-ta	ISCII Tâmil
57005	x-iscii-te	ISCII Télego
57006	x-iscii-as	ISCII Assamês
57007	x-iscii-or	ISCII Oriá
57008	x-iscii-ka	ISCII Kannada
57009	x-iscii-ma	ISCII Malaiala
57010	x-iscii-gu	ISCII Guzerate
57011	x-iscii-pa	ISCII Punjabi
65000	utf-7	Unicode (UTF-7)

	65001	utf-8	Unicode (UTF-8)
--	-------	-------	-----------------

**Valor de retorno**

-

***ReadLine***

Lê uma linha de caracteres do arquivo atual e retorna os dados como uma string.

**Sintaxe**

```
ReadLine()
```

**Parâmetro**

-

**Valor de retorno**

A linha seguinte do arquivo ou uma string vazia se o final do arquivo for atingido.

***ReadToEnd***

Lê o arquivo da posição atual até o final.

**Sintaxe**

```
ReadToEnd()
```

**Parâmetro**

-

**Valor de retorno**

O restante do arquivo da posição atual até o final como uma string.

Se a posição atual for o final do arquivo, uma string vazia ("") é retornada.

***Write***

Grava uma string em um arquivo.

**Sintaxe**

```
Write(string text)
```

**Parâmetro**

Parâmetro	Significado
-----------	-------------

text	A string a ser gravada.
------	-------------------------

**Valor de retorno**

-

***WriteFormat***

Grava a string formatada usando a mesma semântica de FormatStr.

**Sintaxe**

WriteFormat(string formatRule, params object[] formatParams)

**Parâmetro**

Parâmetro	Significado
formatRule	Regra de formatação.
formatParams	Objetos a serem gravados na string formatada.

**Valor de retorno**

-

***WriteLine***

Grava uma string seguida por um caractere de término de linha em um arquivo.

**Sintaxe**

WriteLine(string text)

**Parâmetro**

Parâmetro	Significado
text	A string a ser gravada.

**Valor de retorno**

-

***WriteLineFormat***

Grava uma string formatada e uma nova linha usando a mesma semântica de FormatStr.

**Sintaxe**

WriteLineFormat(string formatRule, params object[] formatParams)

**Parâmetro**

Parâmetro	Significado
formatRule	Regra de formatação.
formatParams	Objetos a serem gravados na string formatada.

**Valor de retorno**

-

Propriedades da classe `MonaLisaTextFile`***FileName***

Retorna um nome de arquivo de um arquivo aberto, ou uma string vazia se nenhum arquivo estiver aberto.

***NewLine***

A string para uma quebra de linha usada, por exemplo, por `WriteLine`.

**Exemplos**

```

; Create DOS object
GetObject("DOS", "MLDos")
;
;-----
; Create file and open to write
;-----
try
File = DOS.CreateTextFile()
File.Create("c:\temp\test.txt")
File.WriteLine("Line 1")
File.Write("Line ")
File.Write("2")
File.WriteLine("")
File.WriteLine("Line 3")
lineNo = 4
File.WriteLineFormat("Line {0}", lineNo)
File.WriteFormat("Line {0}", lineNo + 1)
File.WriteLine("")

```



```

Finally
File.Close()
end-try
;
;-----
; Create file and open to write, line break is LineFeed ($0A)
;-----
try
File = DOS.CreateTextFile(DOS.NewLineLF)
File.Create("c:\temp\test.txt")
File.WriteLine("Line 1 ")
File.Write("Line ")
File.Write("2")
File.WriteLine("")
File.WriteLine("Line 3")
; change to a new line break CRLF ($0D $0A)
File.NewLine = DOS.NewLineCRLF
lineNo = 4
File.WriteLineFormat("Line {0}", lineNo)
File.WriteFormat("Line {0}", lineNo + 1)
File.WriteLine("")
Finally
File.Close()
end-try

;
;-----
; open file to append
;-----
try
File = DOS.CreateTextFile()
File.Append("c:\temp\test.txt")
File.WriteLine("Line 6")
Finally

```

```

File.Close()
end-try
;
;-----
; Open file to append, line break by default CRLF ($0D $0A)
;-----
try
File = DOS.CreateTextFile()
; Set line break to LineFeed ($0A)
File.NewLine = DOS.NewLineLF
File.Append("c:\temp\test.txt")
File.WriteLine("Line 6")
Finally
File.Close()
end-try

;-----
; Open file to read
;-----
try
File = DOS.CreateTextFile()
File.Open("c:\temp\test.txt")
;
while (not File.Eof())
StringListAdd("file", File.ReadLine())
end-while
Finally
File.Close()
end-try
text = StringListGetCommaText("file")
;
;-----
; Read complete text of a file
;-----

```

```

text2 = DOS.FileReadAllText("c:\temp\test.txt")
;
;-----
; Append text to a file
;-----
DOS.FileAppendText("c:\temp\test.txt", "last line")
;

```

## Processamento de XML

### Geral

O objeto MLXmlDocument do MonaLisa torna um objeto da classe XmlDocument do .net Framework disponível.

Essa classe permite a leitura e gravação de arquivos XML.

**Mais informações sobre essa classe podem ser encontradas na Ajuda da Microsoft:**

<http://msdn.microsoft.com/de-de/library/system.xml.xmldocument.aspx>

Métodos adicionais da classe MLXmlDocument

### ***GetAttribute***

Lê o valor de um atributo.

### Sintaxe

GetAttribute(string xpath, string attributeName)

### Parâmetro

Parâmetro	Significado
XPath	Expressão XPath para a seleção do nó.
attributeName	Nome do atributo pertencente à seleção do nó com o parâmetro XPath.

### Valor de retorno

Valor do atributo ou uma string vazia, se o atributo não for encontrado.

### ***GetAttributeOfNode***

Lê o valor de atributo de um elemento XML específico.

### Sintaxe

GetAttributeOfNode(object xmlNode, string attributeName)

**Parâmetro**

Parâmetro	Significado
xmlNode	Instância de um elemento XML do qual o atributo será lido.
attributeName	Nome do atributo

**Valor de retorno**

Valor do atributo ou uma string vazia, se o atributo não for encontrado.

***GetXpathValue***

Lê o valor de um atributo ou elemento.

**Sintaxe**

GetXPathValue(object xmlNode, string xpath)

**Parâmetro**

Parâmetro	Significado
xmlNode	Nó XML
XPath	XPath

**Valor de retorno**

Valor de um atributo ou elemento, ou uma string vazia se nada for encontrado.

***SetAttribute***

Define o valor do atributo de um elemento XML especificado em uma expressão de XPath.

**Sintaxe**

SetAttribute(string xpath, string attributeName, string value)

SetAttribute(string xpath, string attributeName, string value, bool create)

**Parâmetro**

Parâmetro	Significado
XPath	Expressão de XPath para selecionar o nó.
attributeName	Nome de um atributo que pertence ao nó selecionado com o parâmetro XPath.

value	Novo valor do atributo.
create	True: O atributo será criado, se não existir. False: O atributo não será criado.

**Valor de retorno**

True: O valor do atributo foi definido com êxito.

False: O valor não foi definido.

***SetAttributeOfNode***

Define o valor de atributo de um elemento XML específico.

**Sintaxe**

SetAttributeOfNode(object xmlNode, string attributeName, string value)

SetAttributeOfNode(object xmlNode, string attributeName, string value, bool create)

**Parâmetro**

Parâmetro	Significado
xmlNode	Instância de um elemento XML cujo atributo será definido.
attributeName	Nome do atributo.
value	Novo valor do atributo.
create	True: O atributo será criado, se não existir. False: O atributo não será criado.

**Valor de retorno**

True: O valor do atributo foi definido com êxito.

False: O valor não foi definido.

**Scripts de exemplo*****Script de exemplo Ler e modificar atributos***

```

; create XmlDocument Object
GetObject("xml", "MLXmlDocument")
;
; load XML from file
xml.Load("C:\Temp\bookstore.xml")
;

```

```

; or read XML from a string
;xml.LoadXml("<bookstore><book><title lang=\"en\">Italienisch
kochen</title></book></bookstore>")
;
; Get root element of the XML document
root = xml.DocumentElement
;
; Using XPath select all title nodes in an XmlNodeList
nodeList = root.SelectNodes("/bookstore/book/title")
;
; get an node enumerator.
nodeListEnumerator = nodeList.GetEnumerator()
;
; Iterate over all items...
readNext = nodeListEnumerator.MoveNext()
while (readNext)
    ;
    ; get current item node
    node = nodeListEnumerator.Current
    ;
    ;Language attribute
    BookLang = xml.GetAttributeOfNode(node, "lang")
    ; and read value of the title node
    BookTitle = node.InnerText
    ;
    Protocol("Title: {0}, language: {1}", 5,BookTitle, BookLang)
    ;
    ; set Language attribute to German
    xml.SetAttributeOfNode(node, "lang", "de")
    ;
    ; next item node
    readNext = nodeListEnumerator.MoveNext()
    ;
end-while
;
; set language attribute of the 2. Book node to English with XPath
xml.SetAttribute("/bookstore/book[2]/title", "lang", "en")
;
; save as a test
xml.Save("C:\Temp\bookstore_test.xml")
;
;Example for GetXPathValue()
FirstTitle = xml.GetXPathValue(root, "/bookstore/book/title[1]")
CoverTextOfFirstTitle = xml.GetXPathValue(root,
"/bookstore/book/covertext[1]")
CommentOfFirstTitle = xml.GetXPathValue(root, "/bookstore/book[1]/comment()")

```

```
Protocol("GetXPathValue() 1. Title: {0}, cover-text: {1}, comment: {2}", 5,
FirstTitle, CoverTextOfFirstTitle, CommentOfFirstTitle)
;
```

The XML used in the example:

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book>
    <title lang="en">Italian cooking</title>
    <covertext>

    </covertext>
    <author>Giada De Florenz</author>
    <year>2001</year>
    <price>30.00</price>
    <!-- comment for 1st book -->
  </book>
  <book>
    <title lang="de">Harry Schlotter</title>
    <covertext>

    </covertext>
    <author>J R. Bowling</author>
    <year>2005</year>
    <price>29.99</price>
    <!-- comment for 2nd book -->
  </book>
</bookstore>
```

### ***Script de exemplo Criar um arquivo XML***

```
; create XmlDocument Object
GetObject("xml", "MLXmlDocument")
;
; create XML Declaration
decl = xml.CreateXmlDeclaration("1.0", "UTF-8", "")
xml.AppendChild(decl)
;
; create and insert new Fields-Element
fields = xml.CreateElement("FIELDS")
xml.AppendChild(fields)
;
; create new Field-Element
field = xml.CreateElement("FIELD")
;
; Attribute name=test (only if the attribute already exists!)
;xml.SetAttributeOfNode(field, "name", "test")
;
```

```

;Example: ; create and insert CDATA
cdata = xml.CreateCDATASection("test value")
field.AppendChild(cdata)
;
; get Attributes-Collection of the Field-Node
attributes = field.Attributes
;
; create Name-Attribute
name = xml.CreateAttribute("name")
; define valor de atributo
name.Value = "test"
; and insert the Attribute of the Attributes-Collection
attributes.Append(name)
;
; the new Field-Element can now be inserted into the Fields-Element
fields.AppendChild(field)
;
; and save
xml.Save("c:\temp\fields.xml")
;
Resultado
<?xml version="1.0" encoding="UTF-8"?>
<FIELDS>
  <FIELD name="test"></FIELD>
</FIELDS>

```

## Processamento de arquivo Ini

### Métodos da classe MIniFile

#### ***Fechar***

Fecha um arquivo **.ini** aberto.

#### **Sintaxe**

Close()

#### **Parâmetro**

-

#### **Valor de retorno**

-

#### **Exemplo**

```
GetObject("IniFile", "MIniFile")
```

...



```
IniFile.Close()
```

### ***CopySection***

Copia a seção especificada e os identificadores no arquivo '.ini' especificado.

Se o identificador ou a seção não existirem, eles serão criados.

#### **Sintaxe**

```
CopySection(string section, string destFile)
```

#### **Parâmetro**

Parâmetro	Significado
section	Nome da seção
destFile	Nome do arquivo '.ini' para o qual a seção será copiada, na forma de: [Drive\Path\]File name

#### **Valor de retorno**

-

#### **Exemplo**

```
GetObject("IniFile", "MLIniFile")
....
IniFile.CopySection("Section_Test", "d:\test\MLTestIniFile_COPY.ini")
```

### ***DeleteKey***

Exclui o identificador especificado da seleção respectiva.

#### **Sintaxe**

```
DeleteKey(string section, string key)
```

#### **Parâmetro**

Parâmetro	Significado
section	Nome da seção
key	Identificador a ser excluído

#### **Valor de retorno**

Valor	Significado
-------	-------------

True	com êxito
False	com falha

**Exemplo**

```
GetObject("IniFile", "MLIniFile")
...
ok = IniFile.DeleteKey("Delete-Section", "Delete")
```

**EraseSection**

Exclui uma seção especificada de um arquivo .ini

**Sintaxe**

```
EraseSection(string section)
```

**Parâmetro**

Parâmetro	Significado
section	Nome da seção a ser excluída

**Valor de retorno**

Valor	Significado
True	com êxito
False	com falha

**Exemplo**

```
GetObject("IniFile", "MLIniFile")
...
ok = IniFile.EraseSection("Delete-Section")
```

**GetSectionNames**

Retorna uma string separada por vírgulas contendo todos os identificadores de uma seção definida.

**Sintaxe**

```
GetSectionNames(string section)
```

**Parâmetro**

Parâmetro	Significado
-----------	-------------

section	Nome da seção
---------	---------------

**Valor de retorno**

String separada por vírgulas de identificadores.

**Exemplo**

```
GetObject("IniFile", "MLIniFile")
...
sectionNames = IniFile.GetSectionNames("Section_Test")
```

***Abrir***

Abre o arquivo .ini especificado para operações de leitura/gravação.

**Sintaxe**

Open(string fileName)

**Parâmetro**

Parâmetro	Significado
fileName	Nome do arquivo '.ini' na forma de [Drive\Path\]Nome do arquivo

**Valor de retorno**

Valor	Significado
True	com êxito
False	com falha

**Exemplo**

```
GetObject("IniFile", "MLIniFile")
...
ok = IniFile.Open("d:\test\MLTestIniFile.ini")
```

***ReadBool***

Lê o operador booleano do identificador especificado da respectiva seção.

**Sintaxe**

ReadBool(string section, string key)

**Parâmetro**

Parâmetro	Significado
section	Nome da seção
key	Identificador

**Valor de retorno**

Valor lido

**Exemplo**

```
GetObject("IniFile", "MLIniFile")
...
Flag = IniFile.ReadBool("Section_Test", "IsNewCustomer")
```

***ReadInteger***

Lê o valor inteiro do identificador especificado da respectiva seção.

**Sintaxe**

ReadInteger(string section, string key)

**Parâmetro**

Parâmetro	Significado
section	Nome da seção
key	Identificador

**Valor de retorno**

-

**Exemplo**

```
GetObject("IniFile", "MLIniFile")
...
Year = IniFile.ReadInteger("Section_Test", "Year")
```

***ReadSectionToCommatext***

Lê todos pares identificador=valor da seção especificada e retorna-os em uma string separada por vírgulas.

**Sintaxe**

ReadSectionToCommatext(string section)

### Parâmetro

Parâmetro	Significado
section	Nome da seção

### Valor de retorno

String separada por vírgulas contendo todos os pares identificador=valor no formato "identificador=valor1,identificador2=valor2"

### Exemplo

```
GetObject("IniFile", "MLIniFile")
...
section = IniFile.ReadSectionToCommatext("Section_Test")
```

### *ReadString*

Lê a string do identificador especificado da respectiva seção.

### Sintaxe

ReadString(string section, string key)

### Parâmetro

Parâmetro	Significado
section	Nome da seção
key	Identificador

### Valor de retorno

Valor lido

### Exemplo

```
GetObject("IniFile", "MLIniFile")
...
Name = IniFile.ReadString("Section_Test", "Name")s
```

### *WriteBool*

Grava o valor booleano como o valor do identificador especificado na seção respectiva.

Se a seção não existir, ela será criada.

Se o identificador não existir, ele será criado.

**Sintaxe**

WriteBool(string section, string key, bool value)

**Parâmetro**

Parâmetro	Significado
section	Nome da seção
key	Identificador
value	O valor booleano a ser gravado

**Valor de retorno**

-

**Exemplo**

```
GetObject("IniFile", "MLIniFile")
...
IniFile.WriteBool("Section_Test ", "IsNewCustomer", TRUE)
```

***WriteInteger***

Grava o valor inteiro como o valor do identificador especificado na seção respectiva.

Se a seção não existir, ela será criada.

Se o identificador não existir, ele será criado.

**Sintaxe**

WriteInteger(string section, string key, int value)

**Parâmetro**

Parâmetro	Significado
section	Nome da seção
key	Identificador
value	O valor inteiro a ser gravado

**Valor de retorno**

-

**Exemplo**

```
GetObject("IniFile", "MLIniFile")
...
IniFile.WriteInteger("Section_Test ", "Year", 1998)
```

### ***WriteSectionFromCommatext***

Grava todos os pares 'identifier=value' na seção especificada.  
Se o identificador ou a seção não existirem, eles serão criados.

#### **Sintaxe**

```
WriteSectionFromCommatext(string section, string commaText)
```

#### **Parâmetro**

Parâmetro	Significado
section	Nome da seção
commaText	String separada por vírgulas no formato "identifier=value1,identifier2=value2"

#### **Valor de retorno**

-

#### **Exemplo**

```
GetObject("IniFile", "MLIniFile")
...
IniFile.WriteSectionFromCommatext("Section_Test", "NAME=MyName,Company=PA")
```

### ***WriteString***

Grava a string como o valor do identificador especificado na seção respectiva.  
Se a seção não existir, ela será criada.  
Se o identificador não existir, ele será criado.

#### **Sintaxe**

```
WriteString(string section, string key, string value)
```

#### **Parâmetro**

Parâmetro	Significado
section	Nome da seção
key	Identificador a ser excluído

value	A string a ser gravada
-------	------------------------

**Valor de retorno**

-

**Exemplo**

```
GetObject("IniFile", "MLIniFile")
...
IniFile.WriteString("Section_Test ", "Company", "MyCompanyName")
```

**MWS Functions****Geral**

Usar o objeto **MLMwsClient** fornece ao MonaLisa um objeto com funcionalidade MWS.

Ele pode ser carregado pelo método MonaLisa **GetObject()**:

```
GetObject("MwsClient", "MLMwsClient")
```

O objeto fornece todos os métodos necessários para a comunicação com o MWSPProcessService, métodos de ajuda gerais e diversas propriedades necessárias.

**Métodos das classes****Geral**

Uma descrição da estrutura e o significado de diversas opções XMLs e resultados XML ou códigos de resultado podem ser encontrados na documentação do módulo de serviço da web WS Process do ModusOne (Modus\_one Web Services.chm).

A propriedade LastXmlResult pode ser usada para ser o resultado XML gravado por métodos diretamente após o método ter sido chamado.

Ela não é parte da assinatura de método, ainda que, para métodos individuais, seja descrita como uma nos comentários.

O mesmo se aplica à propriedade LastBase64DocumentString do método Doc\_GetFile\_Mime em relação à Parameters doc.

**ClearCache**

Limpa o cache do RepositoryRuntimeService.

**Sintaxe**

```
ClearCache()
```

**Parâmetro**

-



**Valor de retorno**

Valor	Significado
0	com êxito
<>0	com falha

***Doc\_GetFile\_Mime***

Retorna um arquivo que gerencia o objeto de negócio.

**Sintaxe**

Doc\_GetFile\_Mime(string processId, string fileName)

**Parâmetro**

Parâmetro	Significado
processId	ID do processo cujo objeto de negócios deve ser gerenciado.
fileName	Nome do arquivo a ser retornado.
doc	String com código Base64, representado o conteúdo do arquivo.
xmlResult	Vazio quando ok

**Valor de retorno**

Valor	Significado
0	com êxito
<>0	com falha

***Doc\_SetFile\_Mime***

Grava um arquivo que gerencia o objeto de negócio.

**Sintaxe**

Doc\_SetFile\_Mime(string processId, string fileName, string doc)

**Parâmetro**

Parâmetro	Significado
processId	ID do processo cujo objeto de negócios deve ser gerenciado.
fileName	Nome do arquivo a ser retornado.
doc	String com código Base64, representado o conteúdo do arquivo.
xmlResult	Vazio quando ok

### Valor de retorno

-

### ***GetList***

Retorna uma lista de todos os processos existentes no momento.

### Sintaxe

GetList(int startAt, int max, string options)

### Parâmetro

Parâmetro	Significado
startAt	Iniciar índice
max	Número máximo de processos a serem retornados.
options	Este parâmetro pode ser usado para especificar quais processos devem ser inclusos na lista de processos recuperados.
xmlResult	Um XML MwsProcessList quando ok. Caso contrário, MwsError.

### Valor de retorno

Valor	Significado
0	com êxito
<>0	com falha

### ***Obj\_GetStructure***

Retorna a estrutura do objeto referenciado.

**Sintaxe**

Obj\_GetStructure(string processId, string rootRef, int maxLevel)

**Parâmetro**

Parâmetro	Significado
processId	ID do processo do objeto que está sendo referenciado.
rootRef	Referência do objeto dentro do processo
maxLevel	-1: tudo 0: apenas o objeto >0: a estrutura, incluindo todos os elementos filhos no nível n-th
xmlResult	XML do objeto dependendo do componente

**Valor de retorno**

Valor	Significado
0	com êxito
<>0	com falha

***Obj\_SetStructure***

Define a estrutura do objeto de negócio.

**Sintaxe**

Obj\_SetStructure(string processId, string content, string rootRef, int maxLevel)

**Parâmetro**

Parâmetro	Significado
processId	ID do processo do objeto que está sendo referenciado.
content	As informações do objeto de negócio a ser definido.
rootRef	Referência do objeto cuja estrutura deve ser retornada pela chamada.
maxLevel	Determina quanto níveis serão retornados -1 = todos

xmlResult	Estrutura do objeto do objeto de negócio, se OK.
-----------	--

**Valor de retorno**

Valor	Significado
0	com êxito
<>0	com falha

***Obj\_Toggle***

Alterna o status de um objeto

**Sintaxe**

Obj\_Toggle(string processId, string objRef, string rootRef, int maxLevel)

**Parâmetro**

Parâmetro	Significado
processId	ID do processo do objeto que está sendo referenciado.
objRef	Referência do objeto a ser alternado.
rootRef	Referência do objeto cuja estrutura deve ser retornada pela chamada.
maxLevel	Determina quanto níveis serão retornados -1 = todos
xmlResult	Estrutura do objeto do objeto de negócio, se OK.

**Valor de retorno**

Valor	Significado
0	com êxito
<>0	com falha

***Ping***

Retorna a string **Serviço do processo MWS**.

**Sintaxe**

Ping()

**Parâmetro**

-

**Valor de retorno**

String: Serviço do processo MWS

***Process\_Close***

Encerra um processo com o status determinado.

**Sintaxe**

Process\_Close(string processId, int status, string options)

**Parâmetro**

Parâmetro	Significado
processId	ID do processo
status	Definição de status em texto livre
options	Fecha opções XML com título e descrição
xmlResult	Vazio quando ok

**Valor de retorno**

Valor	Significado
0	com êxito
<>0	com falha

***Process\_Create***

Cria um novo processo MWS.

**Sintaxe**

Process\_Create(string processId, string mSystem, string mType, string mTitle, string objectIndex, string objectName, string objectType, string objectData, string options)

**Parâmetro**

Parâmetro	Significado
processId	Se o ID do processo não for passada, uma nova será criada.
mSystem	OID de sistema do sistema em que o objeto reside.
mType	Texto livre para definir o tipo de processo.
mTitle	Texto livre para definir um título para o processo.
objectIndex	Índice do objeto a ser carregado. Se esse parâmetro não estiver definido, o parâmetro objectName será ignorado.
objectName	Nome do objeto a ser carregado. Esse parâmetro é ignorado se o parâmetro objectIndex não for definido.
objectType	Tipo de objeto a ser carregado (no momento, somente P = pacote)
objectData	Dados opcionais de XML para um pacote. Os dados serão atribuídos à primeira seleção passiva no script OnNewStart.
options	Opções XML para o processo MWS.
xmlResult	Se o processo for criado com êxito, a estrutura do pacote ou as informações do processo serão retornadas. A estrutura do pacote é retornada se um comando de início é passado às opções.

### Valor de retorno

Valor	Significado
0	com êxito
<>0	com falha

### ***Process\_Delete***

Exclui um processo aberto e libera seus recursos do servidor.

#### Sintaxe

Process\_Delete(string processId)

#### Parâmetro

Parâmetro	Significado
-----------	-------------

processId	ID do processo a ser excluído.
xmlResult	Vazio quando está ok, caso contrário, MwsError

**Valor de retorno**

Valor	Significado
0	com êxito
<>0	sem êxito

***Process\_Forward***

Encaminha um processo.

**Sintaxe**

Process\_Forward(string processId, string destUser, string destUsergroup, string options)

**Parâmetro**

Parâmetro	Significado
processId	ID do processo
destUser	Nome do usuário de destino para o qual o processo será encaminhado.
destUsergroup	Nome do grupo de destino para o qual o processo será encaminhado.
options	Fecha opções XML com título e descrição
xmlResult	Vazio quando está ok

**Valor de retorno**

Valor	Significado
0	com êxito
<>0	sem êxito

***Process\_GetInfo***

Retorna informações sobre o processo.

### Sintaxe

Process\_GetInfo(string processId, string infoType, string options)

### Parâmetro

Parâmetro	Significado
processId	ID do processo para o qual as informações serão recuperados.
infoType	String separada por vírgulas contendo as informações a serem recuperadas. Essas informações são, em primeiro lugar, os tipos predefinidos: <ul style="list-style-type: none"> <li>PROCESSINFO</li> <li>POOLVARS</li> </ul> e, em segundo lugar, dependentes do processo MWS e, por isso, são carregadas e avaliadas pelo script OnGetInfo.
options	Não suportado no momento.
xmlResult	ProcessInfo XML, se OK

### Valor de retorno

Valor	Significado
0	com êxito
<>0	com falha

### ***Process\_GetLastError***

Retorna o último erro de um processo.

### Sintaxe

Process\_GetLastError(string processId)

### Parâmetro

Parâmetro	Significado
processId	ID do processo, no qual o último erro é recuperado.



xmlResult	Quando há êxito, o último erro XML retorna.
-----------	---

**Valor de retorno**

Valor	Significado
0	com êxito
<>0	sem êxito

***Process\_Open***

Reinicia o processamento de um processo e define o nome do usuário como o do usuário atual.

**Sintaxe**

Process\_Open(string processId, string options)

**Parâmetro**

Parâmetro	Significado
processId	ID do processo a ser aberto.
options	Não suportado no momento.

**Valor de retorno**

Valor	Significado
0	com êxito
<>0	sem êxito

***Process\_SetData***

Define dados XML para o objeto de negócios.

**Sintaxe**

Process\_SetData(string processId, string dataType, string data)

**Parâmetro**

Parâmetro	Significado
processId	ID do processo
dataType	Nome a ser atribuído aos dados. No momento, refere-se ao nome da referência de dados.
data	Os dados XML a serem definidos.
xmlResult	Vazio quando está ok

#### Valor de retorno

Valor	Significado
0	com êxito
<>0	sem êxito

### ***Process\_Start***

Inicia uma ação do processo.

#### Sintaxe

Process\_Start(string processId, string command, string options)

#### Parâmetro

Parâmetro	Significado
processId	ID do processo onde a ação será iniciada.
command	Este parâmetro contém o comando a ser passado Os nomes dos comandos são derivados do script das Atividades da chamada do processo MWS.
options	Este parâmetro pode ser usado para definir várias opções do processo. Ele pode ser usado para definir atividades do processo, passar variáveis etc.
xmlResult	A estrutura do pacote, se tiver êxito

#### Valor de retorno

Valor	Significado
0	com êxito
<>0	com falha

## Métodos da classe Helper

### ***LoadBase64StringFromBinFile***

Carrega o arquivo especificado e converte o conteúdo em uma string Base64.

#### **Sintaxe**

LoadBase64StringFromBinFile(string binFileName)

#### **Parâmetro**

Parâmetro	Significado
binFileName	Nome do arquivo

#### **Valor de retorno**

String codificada Base64 do arquivo.

Se nenhum arquivo for encontrado, uma string vazia é retornada.

#### **Exemplo**

```
newDocString =
MwsClient.LoadBase64StringFromBinFile("c:\temp\MyDoc_changed.docx")
```

### ***SaveBase64StringToBinFile***

Grava uma string Base64 convertida em uma matriz de bytes em um arquivo binário.

#### **Sintaxe**

SaveBase64StringToBinFile(string base64Content, string binFileName)

#### **Parâmetro**

Parâmetro	Significado
base64Content	Conteúdo Base64
binFileName	Nome do arquivo

**Valor de retorno**

True: O arquivo foi gravado com êxito

False: O arquivo não foi gravado com êxito

**Exemplo**

```
if (myDocString <> "")
    ok = MwsClient.SaveBase64StringToBinFile(myDocString,
    "c:\temp\MyDoc_org.docx")
end-if
```

**Propriedades da classe*****LastBase64DocumentString*****ReadOnly**

O método do MWSService Doc\_GetFile\_Mime retorna a string codificada Base64 do documento especificado, além de um código de resultado em números inteiros e a string de resultados XML.

Esta propriedade permite a leitura da última string Base64 recuperada.

**Exemplo**

```
myXmlResult = ""
resCode = MwsClient.Doc_GetFile_Mime(myProcessId, filename)
myDocString = MwsClient.LastBase64DocumentString
myXmlResult = MwsClient.LastXmlResult
```

***LastProcessId*****ReadOnly**

O método MWSService [Process Create](#) permite a definição de uma ID do processo, ou, se o parâmetro correspondente for passado como uma string vazia, permite automaticamente a geração de uma ID de processo.

Esta propriedade permite a leitura da última ID do processo recuperada.

Isto é, independente se a ID do processo foi pré-definida ou criada automaticamente.

A propriedade é definida com a ID do processo correspondente quando a chamada Process\_Createmethod teve êxito.

**Exemplo**

```
myXmlResult = ""
myProcessId = ""
options = ""
resCode = MwsClient.Process_Create(myProcessId, "dml", "MyType", "MyTitle",
    "", "MyBundleName", "P", "", options)
myProcessId = MwsClient.LastProcessId
myXmlResult = MwsClient.LastXmlResult
```

## ***LastXmlResult***

### **Readonly**

A maioria dos métodos MWSService retornam um código de resultado em números inteiros e um XmlResult como string.

Esta propriedade permite a leitura da última string XmlResult recuperada.

### **Exemplo**

```
myXmlResult = ""
resCode = MwsClient.Process_Open(myProcessId, "")
myXmlResult = MwsClient.LastXmlResult
```

## ***MwsUrl***

Esta propriedade é usada para definir a URL do servidor MWS.

A propriedade deve ser definida com o endereço do serviço MWS antes que a primeira chamada do serviço MWS seja feita.

Internamente, quando a propriedade está sendo definida, um canal de serviço previamente aberto é fechado para que a chamada seguinte para o serviço MWS possa criar um novo canal de serviço.

### **Exemplo**

```
MwsClient.MwsUrl = "http://localhost:8011/mws/mwsprocess"
```

Se essa propriedade não é definida, a primeira chamada para o serviço MWS tenta ler a URL exigida para o serviço MWS da chave AppSettings (key="mws").

Uma exceção é lançada se nenhuma configuração correspondente puder ser encontrada aqui também.

### **Exemplo**

```
;***** MLMwsClient *****
;
GetObject("MwsClient", "MLMwsClient")
;
MwsClient.MwsUrl = "http://localhost:8011/mws/mwsprocess"
;
dummy = MwsClient.Ping()
Protocol(dummy, 8)
;
; On setting the URL, a previously created ServiceChannel is closed
;MwsClient.MwsUrl = "http://localhost:8011/mws/mwsprocess"
;dummy = MwsClient.Ping()
;Protocol(dummy, 8)
;MwsClient.MwsUrl = "http://localhost:8011/mws/mwsprocess"
;dummy = MwsClient.Ping()
;Protocol(dummy, 8)
;
myXmlResult = ""
```

```

myProcessId = ""
options = "<mws><mwsaction name=""DATASELECTION"" configured = ""1""
><selparams><selparam name=""Partnernumber"" >0815</selparam><selparam
name=""test""></selparam></selparams></mwsaction></mws>"
res = MwsClient.Process_Create(myProcessId, "dml", "MyType", "MyTitle", "",
"Application2", "P", "", options)
myProcessId = MwsClient.LastProcessId
myXmlResult = MwsClient.LastXmlResult
Protocol(result, 8)
Protocol(res, 8)
;
if (res = 0)
    myXmlResult = ""
    res = MwsClient.Process_Start(myProcessId, "CREATE_DOCUMENTS", "")
    myXmlResult = MwsClient.LastXmlResult
    Protocol(result, 8)
    Protocol(res, 8)
end-if
;
if (myXmlResult <> "")
    filename = ""
    GetObject("xml", "MLXmlDocument")
    xml.LoadXml(myXmlResult)
    filename = xml.GetAttribute("/mws/process/object/object[1]", "filename")
    FreeObject("xml")
end-if
;
if (filename <> "")
    res = MwsClient.Doc_GetFile_Mime(myProcessId, filename)
    myDocString = MwsClient.LastBase64DocumentString
    Protocol(myDocString, 8)
    Protocol(res, 8)
    if (myDocString <> "")
        ok = MwsClient.SaveBase64StringToBinFile(myDocString,
"c:\temp\MyDoc_org.docx")
        Protocol(ok, 8)
    end-if
end-if
;
; now alter document, upload it and then download it again for manual check
if (filename <> "")
    myNewDocString =
MwsClient.LoadBase64StringFromBinFile("c:\temp\MyDoc_changed.docx")
    Protocol(myNewDocString, 8)
    if (myNewDocString <> "")
        res = MwsClient.Doc_SetFile_Mime(myProcessId, filename, myNewDocString)

```

```

Protocol(res, 8)
if (res = 0)
    res = MwsClient.Doc_GetFile_Mime(myProcessId, filename)
    myDocString = MwsClient.LastBase64DocumentString
    Protocol(myDocString, 8)
    Protocol(res, 8)
    if (myDocString <> "")
        ok = MwsClient.SaveBase64StringToBinFile(myDocString,
"c:\temp\MyDoc_downChanged.docx")
        Protocol(ok, 8)
    end-if
end-if
end-if
end-if
;
; upload any file and download it again
filename= "c:\temp\MMDMO_33.B.migrate.xml"
myFileString = MwsClient.LoadBase64StringFromBinFile(filename)
Protocol(myFileString, 8)
res = MwsClient.Doc_SetFile_Mime(myProcessId, filename, myFileString)
Protocol(res, 8)
if (res = 0)
    res = MwsClient.Doc_GetFile_Mime(myProcessId, filename)
    myFileString = MwsClient.LastBase64DocumentString
    Protocol(myFileString, 8)
    Protocol(res, 8)
    if (myFileString <> "")
        ok = MwsClient.SaveBase64StringToBinFile(myFileString,
"c:\temp\MMDMO_DOWNLOAD_33.B.migrate.xml")
        Protocol(ok, 8)
    end-if
end-if
;
;all remaining methods possible
if (myProcessId <> " " )
    myXmlResult = ""
    res = MwsClient.Obj_GetStructure(myProcessId, "", 1)
    myXmlResult = MwsClient.LastXmlResult
    Protocol(result, 8)
    Protocol(res, 8)
;
    myXmlResult = ""
    res = MwsClient.Obj_Toggle(myProcessId, "3C331B86-8552-4AD6-A242-
29106BFBCA21", "", 1)
    myXmlResult = MwsClient.LastXmlResult
    Protocol(result, 8)

```

```

    Protocol(res, 8)
;
    myXmlResult = ""
    res = MwsClient.Process_SetData(myProcessId,
"My_nonExisting_BundleDataId", "<data>MyDataString</data>")
    myXmlResult = MwsClient.LastXmlResult
    Protocol(result, 8)
    Protocol(res, 8)
;
    myXmlResult = ""
    res = MwsClient.Process_GetLastError(myProcessId)
    myXmlResult = MwsClient.LastXmlResult
    Protocol(result, 8)
    Protocol(res, 8)
;
    myXmlResult = ""
    res = MwsClient.ClearCache()
    myXmlResult = MwsClient.LastXmlResult
    Protocol(result, 8)
    Protocol(res, 8)
;
    myXmlResult = ""
    res = MwsClient.Process_GetInfo(myProcessId, "PROCESSINFO", "")
    myXmlResult = MwsClient.LastXmlResult
    Protocol(result, 8)
    Protocol(res, 8)
;
    content = "<mws type=""OBJECT"" version=""1""><process><object
action=""includedoc"" name=""TestDocument"" target_reference=""
position=""after"" /></process></mws>"
    myXmlResult = ""
    res = MwsClient.Obj_SetStructure(myProcessId, content, "", 1)
    myXmlResult = MwsClient.LastXmlResult
    Protocol(result, 8)
    Protocol(res, 8)
;
    myXmlResult = ""
    res = MwsClient.Process_GetInfo(myProcessId, "PROCESSINFO", "")
    myXmlResult = MwsClient.LastXmlResult
    Protocol(result, 8)
    Protocol(res, 8)
;
    options = "<mws version=""2""><title>Forwarded to
Mike</title><description>Description of the task</description></mws>"
    myXmlResult = ""
    res = MwsClient.Process_Forward(myProcessId, "Mike", "", options)

```



```

myXmlResult = MwsClient.LastXmlResult
Protocol(result, 8)
Protocol(res, 8)
;
options = "<mws><processlist><item name=\"listuser\" include=\"Mike\"
/><item name=\"shortprocdesc\" include=\"30\" /></processlist></mws>"
myXmlResult = ""
res = MwsClient.Process_GetList(0, 100, options)
myXmlResult = MwsClient.LastXmlResult
Protocol(result, 8)
Protocol(res, 8)
;
;IntegerDate = date.DateStringToInt64("ddMMyyyyHHmmss", "21012010174853")
;DateStr = date.Int64ToDateString("dd.MM.yyyy HH:mm:ss", IntegerDate)
;
options = "<mws><processlist><item name=\"listuser\" include=\"Mike\"
/><item name=\"shortprocdesc\" include=\"30\" /><item name=\"forwarded\"
include=\"1\" /><item name=\"nonforwarded\" include=\"0\"
/></processlist></mws>"
myXmlResult = ""
res = MwsClient.Process_GetList(0, 100, options)
myXmlResult = MwsClient.LastXmlResult
Protocol(result, 8)
Protocol(res, 8)
;
myXmlResult = ""
res = MwsClient.Process_Open(myProcessId, "")
myXmlResult = MwsClient.LastXmlResult
Protocol(result, 8)
Protocol(res, 8)
;
myXmlResult = ""
res = MwsClient.Process_GetTransDocsList(myProcessId, "")
myXmlResult = MwsClient.LastXmlResult
Protocol(result, 8)
Protocol(res, 8)
;
myXmlResult = ""
res = MwsClient.Process_Close(myProcessId, 0, "")
myXmlResult = MwsClient.LastXmlResult
Protocol(result, 8)
Protocol(res, 8)
;
myXmlResult = ""
res = MwsClient.Process_Delete(myProcessId)
myXmlResult = MwsClient.LastXmlResult

```

```

    Protocol(result, 8)
    Protocol(res, 8)
;
end-if
;
myXmlResult = ""
res = MwsClient.Process_GetList(0, 100, "")
myXmlResult = MwsClient.LastXmlResult
Protocol(result, 8)
Protocol(res, 8)
;
FreeObject("MwsClient")]></scriptcode>
;

```

### Script de exemplo Criar um processo

```

GetObject("MwsClient","MLMwsClient")
MwsClient.MwsUrl = "http://localhost:8011/mws/mwsprocess"
result = MwsClient.Ping()
Protocol(result, 8)
;
myXmlResult = ""
myProcessId = ""
options = ""
res = MwsClient.Process_Create(myProcessId, "MySystemOId", "MyType",
"MyTitle", "", "MyBundleName", "P", "", options)
myProcessId = MwsClient.LastProcessId
myXmlResult = MwsClient.LastXmlResult
Protocol(result, 8)
Protocol(res, 8)
;
FreeObject("MwsClient")

```

## Atributos do usuário

### Geral

Com a configuração correspondente, o objeto `MLUserRepository` permite que as propriedades de perfil e direitos de usuário sejam lidos de LDAP.

### Métodos da classe `MLUserRepository`

#### ***GetCurrentUser***

Retorna o objeto do usuário `MonaLisa` par ao usuário conetado no momento ou um usuário específico.

Se o usuário não for encontrado no `UserRepository`, uma exceção do tipo **Usuário 'xyz' não foi encontrado no UserRepository** é lançada.

### Sintaxe

GetCurrentUser()

GetUser(string userName)

#### Parâmetro

Parâmetro	Significado
UserName	Nome de usuário

#### Valor de retorno

Objeto de usuário MonaLisa

Métodos do objeto do usuário MonaLisa

#### ***GetAttribute***

Retorna o valor do atributo.

Os atributos **UserId** (nome de usuário) e **UserRoleNames** (funções, separadas por vírgulas) estão sempre presentes. Todos os outros atributos dependem de um perfil e seu mapeamento para LDAP.

#### Sintaxe

GetAttribute(string attribute)

#### Parâmetro

Parâmetro	Significado
Attributes	Atributos do usuário

#### Valor de retorno

Valor do atributo

Uma string vazia se o atributo não existir.

#### ***HasPermission***

[▶ Mostrar tudo](#)

Verifica se um usuário tem os direitos correspondentes.

#### Sintaxe

HasPermission(ModusPermission permission)

HasPermission(int permission)

#### Parâmetro

Parâmetro	Significado																																		
Permissão	Direito do usuário																																		
	O parâmetro pode ser retornado como um valor inteiro ou um enumerador.																																		
	Enumerador ModusPermission																																		
		<table border="1"> <thead> <tr> <th>Enumerador</th> <th>Valor inteiro</th> <th>Significado</th> </tr> </thead> <tbody> <tr> <td>ModusPermission.None</td> <td>0</td> <td>Nenhum direito</td> </tr> <tr> <td>ModusPermission.ExecuteModusStudio</td> <td>1</td> <td>Tem permissão para iniciar o ModusOne Studio</td> </tr> <tr> <td>ModusPermission.ShowNavigator</td> <td>2</td> <td>Mostra o Navigator</td> </tr> <tr> <td>ModusPermission.MwsShowNavigator</td> <td>3</td> <td>Mostra o Navigator no cliente fino</td> </tr> <tr> <td>ModusPermission.CommonSettings</td> <td>4</td> <td>Configurações gerais do usuário (ribbon system menu -&gt; settings)</td> </tr> <tr> <td>ModusPermission.ChangeSystem</td> <td>5</td> <td>Permite que um usuário altere o sistema</td> </tr> <tr> <td>ModusPermission.MwsChangeSystem</td> <td>6</td> <td>Permite que um usuário altere o sistema no cliente fino</td> </tr> <tr> <td>ModusPermission.ChangeFolder 7</td> <td>7</td> <td>Permite a alteração para uma pasta inferior, mas não uma pasta pai</td> </tr> <tr> <td>ModusPermission.Search</td> <td>8</td> <td>Pesquisa é permitida no sistema atual</td> </tr> <tr> <td>ModusPermission.ShowTypeFilter</td> <td>9</td> <td>Mostra o filtro de objeto para os diversos tipos de objetos no</td> </tr> </tbody> </table>	Enumerador	Valor inteiro	Significado	ModusPermission.None	0	Nenhum direito	ModusPermission.ExecuteModusStudio	1	Tem permissão para iniciar o ModusOne Studio	ModusPermission.ShowNavigator	2	Mostra o Navigator	ModusPermission.MwsShowNavigator	3	Mostra o Navigator no cliente fino	ModusPermission.CommonSettings	4	Configurações gerais do usuário (ribbon system menu -> settings)	ModusPermission.ChangeSystem	5	Permite que um usuário altere o sistema	ModusPermission.MwsChangeSystem	6	Permite que um usuário altere o sistema no cliente fino	ModusPermission.ChangeFolder 7	7	Permite a alteração para uma pasta inferior, mas não uma pasta pai	ModusPermission.Search	8	Pesquisa é permitida no sistema atual	ModusPermission.ShowTypeFilter	9	Mostra o filtro de objeto para os diversos tipos de objetos no
	Enumerador	Valor inteiro	Significado																																
	ModusPermission.None	0	Nenhum direito																																
	ModusPermission.ExecuteModusStudio	1	Tem permissão para iniciar o ModusOne Studio																																
	ModusPermission.ShowNavigator	2	Mostra o Navigator																																
	ModusPermission.MwsShowNavigator	3	Mostra o Navigator no cliente fino																																
	ModusPermission.CommonSettings	4	Configurações gerais do usuário (ribbon system menu -> settings)																																
	ModusPermission.ChangeSystem	5	Permite que um usuário altere o sistema																																
ModusPermission.MwsChangeSystem	6	Permite que um usuário altere o sistema no cliente fino																																	
ModusPermission.ChangeFolder 7	7	Permite a alteração para uma pasta inferior, mas não uma pasta pai																																	
ModusPermission.Search	8	Pesquisa é permitida no sistema atual																																	
ModusPermission.ShowTypeFilter	9	Mostra o filtro de objeto para os diversos tipos de objetos no																																	

			navigator
ModusPermission.TransferSystem	10		O sistema pode ser transferido
ModusPermission.SynchronizeSystem	11		O sistema pode ser sincronizado
ModusPermission.Administrate	12		O usuário pode executar tarefas de administração
ModusPermission.MwsDownloadDocument	13		O download de documentos criados pelo servidor ModusOne para o ModusOne Client é impossível se a permissão não for concedida.
ModusPermission.MwsProcessList 14	14		
ModusPermission.MwsForwardedProcessList	15		
ModusPermission.MwsInsertDocument	16		
ModusPermission.MwsInsertBlock	17		
ModusPermission.MwsSetBundlePrinter	18		
ModusPermission.MwsSetDocumentPrinter	19		
ModusPermission.MwsSetDocumentPrinterWizard	20		
ModusPermission.MwsManualForwardProcess	21		
ModusPermission.MwsCreateDocuments	22		
ModusPermission.MwsDataSelection	23		
ModusPermission.MwsPrintAndArchive	24		
ModusPermission.MwsAdministrate	25		
ModusPermission.MwsProcessListFromDefinedUser	26		

ModusPermission.MwsChangeFolder	27	
ModusPermission.MwsViewProcess	28	
ModusPermission.MwsViewStack	29	
ModusPermission.MwsViewOpenEnvelope	30	
ModusPermission.MwsViewOpenJob	31	
ModusPermission.MwsClearCache	32	
ModusPermission.ModifyFontFace	2000	
ModusPermission.ModifyFontSize	2001	
ModusPermission.ModifyFontColor	2002	
ModusPermission.ModifyFontStyleBold	2003	
ModusPermission.ModifyFontStyleItalic	2004	
ModusPermission.ModifyFontStyleUnderline	2005	
ModusPermission.ModifyFontAlignment	2006	
ModusPermission.CanPrintInPreview	2011	
ModusPermission.OdinProcessUnlock	3000	O usuário pode desbloquear um processo bloqueado
ModusPermission.OdinProcessSetBack	3001	(Retomar) O usuário pode redefinir um processo de <b>Bloqueado</b> para <b>Apresentação criada</b> ou <b>Importado</b>
ModusPermission.OdinProcessCreateStack	3002	O usuário pode criar uma pilha para impressão/exclusão de trabalhos de processo

ModusPermission.OdinProcessDelete	3003	O usuário pode definir processos para o status "excluir"
ModusPermission.OdinProcessSuspend	3004	O usuário pode bloquear um processo
ModusPermission.OdinOpenJobCreateStack	3005	O usuário pode criar uma pilha para impressão/exclusão de trabalhos abertos
ModusPermission.OdinStackEdit	3500	O usuário pode editar pilhas
ModusPermission.OdinStackFree	3501	O usuário pode liberar pilhas
ModusPermission.OdinStackDelete	3502	O usuário pode excluir pilhas
ModusPermission.OdinStackSetBack	3503	O usuário pode redefinir uma pilha para o status 'Aguardar'
ModusPermission.OdinStackSetBackToStreaming	3504	O usuário pode redefinir uma pilha para fluxo
ModusPermission.OdinStackLock	3505	O usuário pode bloquear uma pilha
ModusPermission.OdinStackUnlock	3506	O usuário pode desbloquear uma pilha
ModusPermission.OdinStackSetBackTo	3507	O usuário pode redefinir uma pilha para um tipo específico de pilha
ModusPermission.OdinJobSetState	3508	O usuário pode redefinir trabalhos para um status

			específico
	ModusPermission.Test	3509	O usuário pode testar documentos e pacotes no Cliente MWS integrado no Studio.
	ModusPermission.OdinJobSetPrinter	3510	O usuário pode alterar a impressora de um trabalho
	ModusPermission.OdinStackEnvelopeRemove	3511	O usuário pode remover um envelope de uma pilha
	ModusPermission.OdinJobRemove	3512	O usuário pode remover um trabalho de um envelope ou pilha

**Valor de retorno**

True: o usuário tem os direitos correspondentes.

False: o usuário não tem os direitos correspondentes

***HasObjectPermission***

[▶ Mostrar tudo](#)

Verifica se um usuário tem os direitos do objeto especificado (criação/leitura/atualização/exclusão) para um objeto ModusOne.

**Sintaxe**

HasObjectPermission(ModusSuiteObjectType objectType, ModusObjectActionPermission permission)

HasObjectPermission(int objectType, int permission)

**Parâmetro**

Parâmetro	Significado						
ObjectType	O parâmetro pode ser retornado como um valor inteiro ou um enumerador.						
	<table border="1"> <thead> <tr> <th>Enumerador</th> <th>Número</th> <th>Tipo de objeto</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Enumerador	Número	Tipo de objeto			
Enumerador	Número	Tipo de objeto					



	inteiro	
ModusSuiteObjectType.TextComponent	4	4 Bloco de texto
ModusSuiteObjectType.TextDocument	6	6 Modelo de documento
ModusSuiteObjectType.TextDocumentCollection	8	8 Pacote
ModusSuiteObjectType.Script	9	9 Script
ModusSuiteObjectType.Condition	10	10 Condição
ModusSuiteObjectType.LookupTable	11	11 Tabela de pesquisa
ModusSuiteObjectType.Process	12	12 Processo/workflow
ModusSuiteObjectType.Folder	13	13 Pasta
ModusSuiteObjectType.DataObject	22	22 Objeto de dados (Xdata)
ModusSuiteObjectType.Selection	24	24 Seleção
ModusSuiteObjectType.DataObjectScript	27	27 Script de objeto de dados
ModusSuiteObjectType.DataObjectQuery	28	28 Consulte de objeto de dados
ModusSuiteObjectType.TransferJob	30	30 Trabalho de transferência
ModusSuiteObjectType.Annex	34	34 Documento anexo
ModusSuiteObjectType.Envelope	36	36 Envelope
ModusSuiteObjectType.PostageDefinition	38	38 Definição de postagem
ModusSuiteObjectType.XsdSelection	39	39 Seleção XSD
ModusSuiteObjectType.Printer	40	40 Impressora online Odin

ModusSuiteObjectType.AliasTable	42	42 Tabela de aliases no Modus
ModusSuiteObjectType.Form	49	49 Formulário Odin
ModusSuiteObjectType.ValueHelp	52	52 Objeto de ajuda de valor para variáveis
ModusSuiteObjectType.DataProviderDefinition	53	53 Definição de consulta de banco de dados
ModusSuiteObjectType.MwsProcess	59	59 Processo MWS
ModusSuiteObjectType.StandardProcess	61	61 Processo padrão
ModusSuiteObjectType.EMailPrinter	62	62 Impressora de email Odin
ModusSuiteObjectType.OutsourcingPrinter	63	63 Impressora terceirizada Odin
ModusSuiteObjectType.CustomUI	64	64 Definição CustomUI (ribbon)
ModusSuiteObjectType.Profile	69	69 Definição de perfil (coleta de atributo de perfil)
ModusSuiteObjectType.Role	71	71 Função
ModusSuiteObjectType.SimpleTextContainer	72	72 Contêiner para blocos de texto simples, sem formatação, a serem inseridos em documentos do Word
ModusSuiteObjectType.RoleMapping	73	73 Armazena o mapeamento das funções externas das funções do Modus

Permission	O parâmetro pode ser retornado como um valor inteiro ou um enumerador.	
	<b>Enumerador</b>	<b>Número inteiro</b>
	ModusObjectActionPermission.None	0
	ModusObjectActionPermission.Create	1
	ModusObjectActionPermission.Read	2
	ModusObjectActionPermission.Update	4
	ModusObjectActionPermission.Delete	8

**Valor de retorno**

True: o usuário tem os direitos de objeto correspondentes.

False: o usuário não tem os direitos de objeto correspondentes

***IsInRole***

Verifica se o usuário é um membro da função correspondente.

**Sintaxe**

IsInRole(string roleName)

**Parâmetro**

Parâmetro	Significado
RoleName	Nome da função

**Valor de retorno**

True: o usuário é um membro da função.

False: o usuário não é um membro da função.

**Script de exemplo**

```
; Initialize MonaLisa UserRepository Object
GetObject("userRepository", "MLUserRepository")
```

```
try
```

```

; Get current logged in User
currentUser = userRepository.CurrentUser()

; Get user from UserRepository
user = userRepository.GetUser("JKlinger")

; UserName ("UserId" always exists per default)
userName = user.GetAttribute("UserId")

; Roles of the user ("UserRoleNames" always exists per default)
roles = user.GetAttribute("UserRoleNames")

; Check if user is in a specific role
isAdmin = user.IsInRole("AdminRole")

; Check if user has permission to change a system
canChangeSystem = user.HasPermission(ModusPermission.ChangeSystem)
; same with integer instead of enum
canChangeSystem_int = user.HasPermission(5)

; Check if user has object-permission to create/delete a script
canCreateScript = user.HasObjectPermission(ModusSuiteObjectType.Script,
ModusObjectActionPermission.Create)

canDeleteScript = user.HasObjectPermission(ModusSuiteObjectType.Script,
ModusObjectActionPermission.Delete)

; same with integer instead of enum
canCreateScript_int = user.HasObjectPermission(9, 1)
canDeleteScript_int = user.HasObjectPermission(9, 8)
canCreateDeleteScript_int = user.HasObjectPermission(9, 1+8) ; combined
create+delete -> 1+8

OnError

Protocol("Error reading User from UserRepository.", 0)

end-try

```

## Configuração

Arquivos de configuração (e.g. ModusStudio.exe.config etc.) necessitam da seguinte seleção:

```

<configuration>
  <configSections>
  ...

```

```

    <section name="userRepository"
type="ModusSuite.UserRepository.DataAccess.UserRepositoryConfiguration,
ModusSuite.UserRepository.DataAccess"/>
    <section name="userRepository_Ldap"
type="ModusSuite.UserRepository.DataAccess.Ldap.LdapConfiguration,
ModusSuite.UserRepository.DataAccess"/>
    ...
</configSections>
<userRepository configSource = "UserRepository.config"/>
<userRepository_Ldap configSource = "UserRepository_ldap.config"/>
</configuration>

```

## Propriedades e métodos de contextos

### Geral

Contexto para processamento em lote.

### Contexto ModusBatch

Propriedades do contexto ModusBatch

Propriedade	Significado
ErrorMessage	Mensagem de erro. Se um valor é atribuído, a geração é abortada com a mensagem de erro correspondente.
InsertObject	Continuar processamento True/False
InsertObjectS (ObjektEinfuegen)	Continuar processamento Y/N
DocumentCollection	Pacote a ser gerado.
SystemOld (Nome do sistema)	ID de objeto do sistema em que o pacote reside.
Directory	Diretório no qual os pacotes gerado serão salvos. (Somente leitura)

Métodos do contexto ModusBatch

### ***AssignData***

Atribui dados XML com uma DataID.

**Sintaxe**

```
AssignData(string dataId, string xmlData)
```

**Parâmetro**

Parâmetro	Significado
dataId	DataID
xmlData	XML

**Valor de retorno**

-

**Exemplo**

```
AssignData(System.DataId, System.LastXml)
```

***DeleteDirectory***

Exclui o diretório definido na propriedade 'Directory'.

**Sintaxe**

```
DeleteDirectory()
```

**Parâmetro**

-

**Valor de retorno**

-

***GetData***

Recupera os dados XML de determinada DataID.

**Sintaxe**

```
GetData(string dataId)
```

**Parâmetro**

Parâmetro	Significado
dataId	ID de dados

**Valor de retorno**

Dados XML

**Exemplo**

```
GetData ( System.DataId )
```

***SetDataContext***

Especifica os dados XML a serem navegados.

**Sintaxe**

```
SetDataContext(string xmlData)
```

**Parâmetro**

Parâmetro	Significado
xmlData	XML

**Valor de retorno**

-

**Exemplo**

```
SetDataContext ( System.LastXml )
```

**Context MwsProcess****Geral**

Contexto para processos MWS.

**Propriedades do contexto MwsProcess**

[▶ Mostrar tudo](#)

Propriedade	Significado
ChangeAction	Executa o script atribuído ao evento <b>OnStatusChange</b> e fornece um enumerador com as seguintes propriedades: None: no action Toggle: Change status SetStructure: Refresh structure SetData: Assign data
IsObjectSet	Estados se o objeto tiver sido definido ou configurado. (ReadOnly)
ObjectData	Dados XML para recuperação de dados (somente leitura)
ProcessBlocked	O pacote deve ser encaminhado ou não: True / False (somente leitura)

ProcessDescription	Descrição do processo								
ProcessDirectory	Diretório no qual os processos estão para serem salvos.								
ProcessId	Id do processo								
ProcessTitle	Título do processo								
ProcessType	Tipo do processo								
ModusProcess	<p>Fornece as seguintes propriedades e métodos:</p> <p><b>Propriedades:</b></p> <table border="1"> <thead> <tr> <th>Propriedade</th> <th>Significado</th> </tr> </thead> <tbody> <tr> <td>ObjectName</td> <td>Nomes dos processos</td> </tr> <tr> <td>SystemOID</td> <td>A ID do sistema em que o processo é executado</td> </tr> </tbody> </table> <p><b>Métodos:</b></p> <p>Cancelar  Cancela o processo.</p> <p><b>Sintaxe</b></p> <p>Cancel()</p> <p><b>Parâmetro</b></p> <p>-</p> <p><b>Valor de retorno</b></p> <p>-</p> <p><b>Exemplo</b></p> <p>ModusProcess.Cancel()  GetComponent</p> <p>Permite o acesso às propriedades e métodos dos componentes.</p> <p><b>Sintaxe</b></p> <p>GetComponent(string componentName)</p> <p><b>Parâmetro</b></p> <table border="1"> <thead> <tr> <th>Parâmetro</th> <th>Significado</th> </tr> </thead> <tbody> </tbody> </table>	Propriedade	Significado	ObjectName	Nomes dos processos	SystemOID	A ID do sistema em que o processo é executado	Parâmetro	Significado
Propriedade	Significado								
ObjectName	Nomes dos processos								
SystemOID	A ID do sistema em que o processo é executado								
Parâmetro	Significado								



componentName	Nome do componente
---------------	--------------------

**Valor de retorno**

-

**Exemplo**

```
DocumentCollection =
ModusProcess.GetComponent("DocumentCollectionManager")
```

```
SendInfoMessage
```

Envia uma mensagem informativa para se conectar aos clientes de controle remoto. Um carimbo de data e hora é inserido antes de cada mensagem.

**Sintaxe**

```
SendInfoMessage(string message)
```

**Parâmetro**

Parâmetro	Significado
message	A mensagem informativa (InfoMessage) a ser enviada

**Valor de retorno**

-

**Exemplo**

```
ModusProcess.SendInfoMessage("This is the InfoText")
```

```
Parar
```

Finaliza as execuções dos processos.

**Sintaxe**

```
Stop()
```

**Parâmetro**

-

**Valor de retorno**

-

**Exemplo**

```
ModusProcess.Stop()
```

## Métodos do contexto MwsProcess

***GetActivityStatus***

Retorna propriedades e métodos para acessar uma MWSActivity.

**Sintaxe**

GetActivityStatus(string statusName)

**Parâmetro**

Parâmetro	Significado
statusName	Nome de MWSActivityStatus

**Valor de retorno**

Valor	Significado
IsReady	A atividade está pronta para execução: True / False
IsBusy	A atividade está ocupada: True / False
IsFinished	A atividade está concluída: True / False
IsConfigured	A atividade está configurada: True / False
InformationXml	Informação para a troca com o cliente
SetLastError(int errorCode, string errorMessage)	Define um erro para a atividade com uma mensagem e um número de erro.
EnforceConfiguration(string informationXml)	Força o processo a parar a execução, pois dados de configuração adicionais são necessários.
EnforceConfiguration(string informationXml, string infoType)	Força o processo a parar a execução, pois dados de configuração adicionais são necessários. O processamento é retornado ao Cliente ou chamador com as informações especificadas. O tipo de informação retornado é definido como tipo mws no XML.

***GetForwardInformation***

Retorna as informações de encaminhamento.

**Sintaxe**

GetForwardInformation()

**Parâmetro**

-

**Valor de retorno**

String XML com DestUser e DestUserGroup.

***GetObjectSelectionParametersRetorna parâmetros de seleção.*****Sintaxe**

GetObjectSelectionParameters(string systemOld, string folder, string filter)

**Parâmetro**

Parâmetro	Significado
systemOld	ID do objeto do sistema.
folder	Subdiretório do sistema.
filter	Filtro do tipo do objeto. Atualmente, é possível apenas "P" para pacotes.

**Valor de retorno**

Parâmetro de seleção como string XML.

***GetOptions***

Retorna as opções para determinada OptionsID.

**Sintaxe**

GetOptions(string optionsId)

**Parâmetro**

Parâmetro	Significado
optionsId	OptionsID

**Valor de retorno**

Opções da OptionsID, se encontradas.

***IsInformationRequested***

Verifica se as informações foram solicitadas pela InformationID transmitida.

**Sintaxe**

IsInformationRequested(string informationId)

**Parâmetro**

Parâmetro	Significado
informationId	ID das informações

**Valor de retorno**

True / False

***SetObjectSelectionParameters***

Define os parâmetros de seleção.

**Sintaxe**

SetObjectSelectionParameters(string xmlOptions)

**Parâmetro**

Parâmetro	Significado
xmlOptions	Opções do XML

**Valor de retorno**

-

**Contexto OdinArchive****Geral**

Contexto para arquivamento.

**Propriedades do contexto OdinArchive**[▶ Mostrar tudo](#)

Parâmetro	Significado		
CurrentEnvelope	Envelope atual, se for encontrado. (ReadOnly)		
	<b>Métodos</b> GetValue Retorna o valor da coluna dada. <b>Sintaxe</b> GetValue(string columnName)		
	<table border="1"> <thead> <tr> <th>Parâmetro</th> <th>Significado</th> </tr> </thead> <tbody> </tbody> </table>	Parâmetro	Significado
Parâmetro	Significado		

columnName	Nome da coluna
------------	----------------

**Valor de retorno**

Valor da coluna

Se o tipo de dados da coluna do banco de dados for DATE, Datetime ou qualquer tipo de dados válido, a data lida geralmente será retornada no formato MM/DD/AAAA hh:mm:ss.

**Exemplo**

```
GetObject("DateVar", "MLDate")
```

...

```
StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT")
```

;Converter a data no formato desejado com a ajuda do objeto MLDate

```
IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy  
hh:mm:ss", StringDateVar )
```

```
v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy hh:mm:ss  
", IntDateVar)
```

GetFieldNameEnumerator

Retorna um enumerador.

**Sintaxe**

```
GetEnvelopeEnumerator()
```

**Parâmetro**

-

**Valor de retorno**

Enumerador com estes métodos:

Método	Função
Current	Retorna o objeto atual.
MoveNext	Move a posição até o próximo objeto a ser processado.
Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.

CurrentJob

Trabalho atual, se for encontrado. (ReadOnly)

**Métodos**

GetValue

Retorna o valor da coluna dada.

**Sintaxe**

```
GetValue(string columnName)
```

Parâmetro	Significado
columnName	Nome da coluna

**Valor de retorno**

Valor da coluna

Se o tipo de dados da coluna do banco de dados for DATE, Datetime ou qualquer tipo de dados válido, a data lida geralmente será retornada no formato MM/DD/AAAA hh:mm:ss.

**Exemplo**

```
GetObject("DateVar", "MLDate")
```

...

```
StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT")
```

;Converter a data no formato desejado com a ajuda do objeto MLDate

```
IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy  
hh:mm:ss", StringDateVar )
```

```
v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy hh:mm:ss  
", IntDateVar)
```

GetFieldNameEnumerator

Retorna um enumerador.

**Sintaxe**

```
GetEnvelopeEnumerator()
```

**Parâmetro**

-

**Valor de retorno**

Enumerador com estes métodos:

Método	Função
Current	Retorna o objeto atual.
MoveNext	Move a posição até o próximo objeto a ser processado.
Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.

CurrentStack

Pilha atual (ReadOnly)

**Métodos**

GetValue

Retorna o valor da coluna dada.

**Sintaxe**

GetValue(string columnName)

Parâmetro	Significado
columnName	Nome da coluna

#### Valor de retorno

Valor da coluna

Se o tipo de dados da coluna do banco de dados for DATE, Datetime ou qualquer tipo de dados válido, a data lida geralmente será retornada no formato MM/DD/AAAA hh:mm:ss.

#### Exemplo

```
GetObject("DateVar", "MLDate")
```

...

```
StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT")
;Converter a data no formato desejado com a ajuda do objeto
MLDate
```

```
IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy
hh:mm:ss", StringDateVar )
```

```
v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy hh:mm:ss
", IntDateVar)
```

GetFieldNameEnumerator

Retorna um enumerador.

#### Sintaxe

GetEnvelopeEnumerator()

#### Parâmetro

-

#### Valor de retorno

Enumerador com estes métodos:

Método	Função
Current	Retorna o objeto atual.
MoveNext	Move a posição até o próximo objeto a ser processado.
Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.

Métodos do contexto OdinArchive

### **HasData**

Verifica se os dados existem para o tipo determinado.

### Sintaxe

HasData(OdinDataRowType dataType)

### Parâmetro

Parâmetro	Significado	Valores possíveis
DataType	Tipo de dados	OdinDataRowType.Process OdinDataRowType.Stack OdinDataRowType.Envelope OdinDataRowType.Job OdinDataRowType.VirtualJob

### Valor de retorno

True: A linha de dados especificada existe.

False: A linha de dados especificada não existe.

## Contexto OdinConvert

### Geral

Contexto para conversão.

### Propriedades do contexto OdinConvert

[▶ Mostrar tudo](#)

Propriedade	Significado
ConvertDocument	<p>True / False</p> <p>Essa propriedade pode ser usada para controlar o evento <b>OnDocumentConvert</b> e determinar se um documento pode ser convertido/adicionado ao arquivo de saída ou não.</p> <p>Essa propriedade só pode ser avaliada depois do evento <b>OnDocumentConvert</b>. Ela não tem influência sobre outros eventos.</p>
CurrentDocument	<p>Documento atual quando encontrado. (somente leitura)</p> <p><b>Métodos</b></p> <p>GetDocumentFileName</p> <p>Retorna o nome do documento original.</p> <p><b>Sintaxe</b></p> <p>GetDocumentFileName()</p>



**Parâmetro**

-

**Valor de retorno**

Nome do arquivo

GetDuplex

Retorna o modo duplex para este documento.

**Sintaxe**

GetDuplex()

**Parâmetro**

-

**Valor de retorno**

Enumerador com estes métodos:

Método	Significado
OdinDuplexSetting.Simplex	Sem duplex
OdinDuplexSetting.Horizontal	Duplex na borda menor
OdinDuplexSetting.Vertical	Duplex na borda maior

GetFileName

Retorna o nome do arquivo de apresentação do documento.

**Sintaxe**

GetFileName()

**Parâmetro**

-

**Valor de retorno**

Nome do arquivo de apresentação

GetFirstInputBin

Retorna a bandeja da primeira página do documento.

**Sintaxe**

GetFirstInputBin()

**Parâmetro**

-

**Valor de retorno**

Nome da bandeja

GetId

Retorna a ID do documento (= Job-ID ou StackCoverPage / EnvelopeCoverPage, conforme o caso)

**Sintaxe**

GetId()

**Parâmetro**

-

**Valor de retorno**

ID do documento.

GetNextInputBin

Retorna a bandeja de acompanhamento para as páginas 2-n de um documento.

**Sintaxe**

GetNextInputBin()

**Parâmetro**

-

**Valor de retorno**

NextInputBin

GetPageCount

Retorna o número de páginas.

**Sintaxe**

GetPageCount()

**Parâmetro**

-

**Valor de retorno**

Número de páginas

IsPortrait

Verifica se o layout da página especificada é retrato.

**Sintaxe**

IsPortrait(int pageNumber)

Parâmetro	Significado
pageNumber	Número da página

**Valor de retorno**

True: O layout da página é retrato.

False: O layout da página é paisagem.

SetDuplex

Especifica o modo duplex para esse documento.

#### Sintaxe

SetDuplex(duplexMode)

Parâmetro	Método	Significado
duplexMode	OdinDuplexSetting.Simplex	Sem duplex
	OdinDuplexSetting.Horizontal	Duplex na borda menor
	OdinDuplexSetting.Vertical	Duplex na borda maior

#### Valor de retorno

-

SetFirstInputBin

Define a bandeja da primeira página do documento.

#### Sintaxe

SetFirstInputBin(string inputBin)

Parâmetro	Significado
inputBin	Bandeja

#### Valor de retorno

-

SetNextInputBin

Define a bandeja de acompanhamento para as páginas 2-n de um documento.

#### Sintaxe

SetNextInputBin(string inputBin)

Parâmetro	Significado
inputBin	Bandeja

#### Valor de retorno

-

HasDocument Documento carregado True/False (somente leitura)

CurrentEnvelope Envelope atual, se encontrado. (Somente leitura)

**Métodos**

GetValue

Retorna o valor da coluna dada.

**Sintaxe**

GetValue(string columnName)

Parâmetro	Significado
columnName	Nome da coluna

**Valor de retorno**

Valor da coluna

Se o tipo de dados da coluna do banco de dados for DATE, Datetime ou qualquer tipo de dados válido, a data lida geralmente será retornada no formato MM/DD/AAAA hh:mm:ss.

**Exemplo**

```
GetObject("DateVar", "MLDate")
...
StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT")
;Converter a data no formato desejado com a ajuda do objeto
MLDate
IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy
hh:mm:ss", StringDateVar )
v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy
hh:mm:ss ", IntDateVar)
```

GetFieldNameEnumerator

Retorna um enumerador.

**Sintaxe**

GetEnvelopeEnumerator()

**Parâmetro**

-

**Valor de retorno**

Enumerador com estes métodos:

Método	Função
Current	Retorna o objeto atual.
MoveNext	Move a posição até o próximo objeto a ser processado.
Redefinir	Redefine o enumerador de volta ao início do primeiro

	elemento.								
CurrentJob	<p>Trabalho atual, se encontrado. (Somente leitura)</p> <p><b>Métodos</b></p> <p>GetValue</p> <p>Retorna o valor da coluna dada.</p> <p><b>Sintaxe</b></p> <p>GetValue(string columnName)</p> <table border="1"> <thead> <tr> <th>Parâmetro</th> <th>Significado</th> </tr> </thead> <tbody> <tr> <td>columnName</td> <td>Nome da coluna</td> </tr> </tbody> </table> <p><b>Valor de retorno</b></p> <p>Valor da coluna</p> <p>Se o tipo de dados da coluna do banco de dados for DATE, Datetime ou qualquer tipo de dados válido, a data lida geralmente será retornada no formato MM/DD/AAAA hh:mm:ss.</p> <p><b>Exemplo</b></p> <pre>GetObject("DateVar", "MLDate") ... StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT") ;Converter a data no formato desejado com a ajuda do objeto MLDate IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy hh:mm:ss", StringDateVar ) v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy hh:mm:ss ", IntDateVar)</pre> <p>GetFieldNameEnumerator</p> <p>Retorna um enumerador.</p> <p><b>Sintaxe</b></p> <p>GetEnvelopeEnumerator()</p> <p><b>Parâmetro</b></p> <p>-</p> <p><b>Valor de retorno</b></p> <p>Enumerador com estes métodos:</p> <table border="1"> <thead> <tr> <th>Método</th> <th>Função</th> </tr> </thead> <tbody> <tr> <td>Current</td> <td>Retorna o objeto atual.</td> </tr> </tbody> </table>	Parâmetro	Significado	columnName	Nome da coluna	Método	Função	Current	Retorna o objeto atual.
Parâmetro	Significado								
columnName	Nome da coluna								
Método	Função								
Current	Retorna o objeto atual.								

	<table border="1"> <tr> <td>MoveNext</td> <td>Move a posição até o próximo objeto a ser processado.</td> </tr> <tr> <td>Redefinir</td> <td>Redefine o enumerador de volta ao início do primeiro elemento.</td> </tr> </table>	MoveNext	Move a posição até o próximo objeto a ser processado.	Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.
MoveNext	Move a posição até o próximo objeto a ser processado.				
Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.				
CurrentStack	<p>Pilha atual (Somente leitura)</p> <p><b>Métodos</b></p> <p>GetValue</p> <p>Retorna o valor da coluna dada.</p> <p><b>Sintaxe</b></p> <p>GetValue(string columnName)</p> <table border="1"> <thead> <tr> <th>Parâmetro</th> <th>Significado</th> </tr> </thead> <tbody> <tr> <td>columnName</td> <td>Nome da coluna</td> </tr> </tbody> </table> <p><b>Valor de retorno</b></p> <p>Valor da coluna</p> <p>Se o tipo de dados da coluna do banco de dados for DATE, Datetime ou qualquer tipo de dados válido, a data lida geralmente será retornada no formato MM/DD/AAAA hh:mm:ss.</p> <p><b>Exemplo</b></p> <pre>GetObject("DateVar", "MLDate") ... StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT") ;Converter a data no formato desejado com a ajuda do objeto MLDate IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy hh:mm:ss", StringDateVar ) v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy hh:mm:ss ", IntDateVar)  GetFieldNameEnumerator Retorna um enumerador.</pre> <p><b>Sintaxe</b></p> <p>GetEnvelopeEnumerator()</p> <p><b>Parâmetro</b></p> <p>-</p> <p><b>Valor de retorno</b></p> <p>Enumerador com estes métodos:</p>	Parâmetro	Significado	columnName	Nome da coluna
Parâmetro	Significado				
columnName	Nome da coluna				

	Método	Função
	Current	Retorna o objeto atual.
	MoveNext	Movê a posição até o próximo objeto a ser processado.
	Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.

## Métodos do contexto OdinConvert

### ***HasData***

Verifica se os dados existem para o tipo determinado.

### Sintaxe

HasData(OdinDataRowType dataType)

### Parâmetro

Parâmetro	Significado	Valores possíveis
DataType	Tipo de dados	OdinDataRowType.Process OdinDataRowType.Stack OdinDataRowType.Envelope OdinDataRowType.Job OdinDataRowType.VirtualJob

### Valor de retorno

True: A linha de dados especificada existe.

False: A linha de dados especificada não existe.

## Contexto OdinCustomProcess

### Geral

Contexto para processamento de pilha.

### Propriedades do contexto OdinCustomProcess

Parâmetro	Significado
CurrentEnvelope	Envelope atual, se for encontrado. (ReadOnly)

**Métodos**

GetValue

Retorna o valor da coluna dada.

**Sintaxe**

GetValue(string columnName)

Parâmetro	Significado
columnName	Nome da coluna

**Valor de retorno**

Valor da coluna

Se o tipo de dados da coluna do banco de dados for DATE, Datetime ou qualquer tipo de dados válido, a data lida geralmente será retornada no formato MM/DD/AAAA hh:mm:ss.

**Exemplo**

```
GetObject("DateVar", "MLDate")
```

...

```
StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT")
```

```
;Converter a data no formato desejado com a ajuda do objeto  
MLDate
```

```
IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy
```

```
hh:mm:ss", StringDateVar )
```

```
v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy hh:mm:ss  
", IntDateVar)
```

GetFieldNameEnumerator

Retorna um enumerador.

**Sintaxe**

GetEnvelopeEnumerator()

**Parâmetro**

-

**Valor de retorno**

Enumerador com estes métodos:

Método	Função
Current	Retorna o objeto atual.
MoveNext	Move a posição até o próximo objeto a ser processado.



	<table border="1"> <tr> <td>Redefinir</td> <td>Redefine o enumerador de volta ao início do primeiro elemento.</td> </tr> </table>	Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.						
Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.								
CurrentJob	<p>Trabalho atual, se for encontrado. (ReadOnly)</p> <p><b>Métodos</b></p> <p>GetValue</p> <p>Retorna o valor da coluna dada.</p> <p><b>Sintaxe</b></p> <p>GetValue(string columnName)</p> <table border="1"> <thead> <tr> <th>Parâmetro</th> <th>Significado</th> </tr> </thead> <tbody> <tr> <td>columnName</td> <td>Nome da coluna</td> </tr> </tbody> </table> <p><b>Valor de retorno</b></p> <p>Valor da coluna</p> <p>Se o tipo de dados da coluna do banco de dados for DATE, Datetime ou qualquer tipo de dados válido, a data lida geralmente será retornada no formato MM/DD/AAAA hh:mm:ss.</p> <p><b>Exemplo</b></p> <pre>GetObject("DateVar", "MLDate") ... StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT") ;Converter a data no formato desejado com a ajuda do objeto MLDate IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy hh:mm:ss", StringDateVar ) v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy hh:mm:ss ", IntDateVar)</pre> <p>GetFieldNameEnumerator</p> <p>Retorna um enumerador.</p> <p><b>Sintaxe</b></p> <p>GetEnvelopeEnumerator()</p> <p><b>Parâmetro</b></p> <p>-</p> <p><b>Valor de retorno</b></p> <p>Enumerador com estes métodos:</p> <table border="1"> <thead> <tr> <th>Método</th> <th>Função</th> </tr> </thead> <tbody> <tr> <td>Current</td> <td>Retorna o objeto atual.</td> </tr> </tbody> </table>	Parâmetro	Significado	columnName	Nome da coluna	Método	Função	Current	Retorna o objeto atual.
Parâmetro	Significado								
columnName	Nome da coluna								
Método	Função								
Current	Retorna o objeto atual.								

	MoveNext	Move a posição até o próximo objeto a ser processado.						
	Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.						
CurrentStack	<p>Pilha atual (ReadOnly)</p> <p><b>Métodos</b></p> <p>GetValue</p> <p>Retorna o valor da coluna dada.</p> <p><b>Sintaxe</b></p> <p>GetValue(string columnName)</p> <table border="1"> <thead> <tr> <th>Parâmetro</th> <th>Significado</th> </tr> </thead> <tbody> <tr> <td>columnName</td> <td>Nome da coluna</td> </tr> </tbody> </table> <p><b>Valor de retorno</b></p> <p>Valor da coluna</p> <p>Se o tipo de dados da coluna do banco de dados for DATE, Datetime ou qualquer tipo de dados válido, a data lida geralmente será retornada no formato MM/DD/AAAA hh:mm:ss.</p> <p><b>Exemplo</b></p> <pre>GetObject("DateVar", "MLDate") ... StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT") ;Converter a data no formato desejado com a ajuda do objeto MLDate IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy hh:mm:ss", StringDateVar ) v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy hh:mm:ss ", IntDateVar)</pre> <p>GetFieldNameEnumerator</p> <p>Retorna um enumerador.</p> <p><b>Sintaxe</b></p> <p>GetEnvelopeEnumerator()</p> <p><b>Parâmetro</b></p> <p>-</p> <p><b>Valor de retorno</b></p> <p>Enumerador com estes métodos:</p> <table border="1"> <thead> <tr> <th>Método</th> <th>Função</th> </tr> </thead> </table>		Parâmetro	Significado	columnName	Nome da coluna	Método	Função
Parâmetro	Significado							
columnName	Nome da coluna							
Método	Função							

	Current	Retorna o objeto atual.
	MoveNext	Move a posição até o próximo objeto a ser processado.
	Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.

## Métodos do contexto TextVariable

### **HasData**

Verifica se os dados existem para o tipo determinado.

### Sintaxe

HasData(OdinDataRowType dataType)

### Parâmetro

Parâmetro	Significado	Valores possíveis
DataType	Tipo de dados	OdinDataRowType.Process OdinDataRowType.Stack OdinDataRowType.Envelope OdinDataRowType.Job OdinDataRowType.VirtualJob

### Valor de retorno

True: A linha de dados especificada existe.

False: A linha de dados especificada não existe.

## Contexto OdinDbTask

### Geral

Esse contexto fornece acesso à CurrentStack/CurrentEnvelope/CurrentJob quando os eventos OnDeleteJob/OnDeleteStack do contexto OdinDbTasks são executados.

### Propriedades do contexto OdinDbTask

Parâmetro	Significado
CurrentEnvelope	Envelope atual, se for encontrado. (ReadOnly)

**Métodos**

GetValue

Retorna o valor da coluna dada.

**Sintaxe**

GetValue(string columnName)

Parâmetro	Significado
columnName	Nome da coluna

**Valor de retorno**

Valor da coluna

Se o tipo de dados da coluna do banco de dados for DATE, Datetime ou qualquer tipo de dados válido, a data lida geralmente será retornada no formato MM/DD/AAAA hh:mm:ss.

**Exemplo**

```
GetObject("DateVar", "MLDate")
```

...

```
StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT")
;Converter a data no formato desejado com a ajuda do objeto
MLDate
```

```
IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy
hh:mm:ss", StringDateVar )
```

```
v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy hh:mm:ss
", IntDateVar)
```

GetFieldNameEnumerator

Retorna um enumerador.

**Sintaxe**

GetEnvelopeEnumerator()

**Parâmetro**

-

**Valor de retorno**

Enumerador com estes métodos:

Método	Função
Current	Retorna o objeto atual.
MoveNext	Move a posição até o próximo objeto a ser processado.
Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.

**CurrentJob**

Trabalho atual, se for encontrado. (ReadOnly)

**Métodos**

GetValue

Retorna o valor da coluna dada.

**Sintaxe**

GetValue(string columnName)

Parâmetro	Significado
columnName	Nome da coluna

**Valor de retorno**

Valor da coluna

Se o tipo de dados da coluna do banco de dados for DATE, Datetime ou qualquer tipo de dados válido, a data lida geralmente será retornada no formato MM/DD/AAAA hh:mm:ss.

**Exemplo**

```
GetObject("DateVar", "MLDate")
```

...

```
StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT")
;Converter a data no formato desejado com a ajuda do objeto
MLDate
```

```
IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy
hh:mm:ss", StringDateVar )
```

```
v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy hh:mm:ss
", IntDateVar)
```

GetFieldNameEnumerator

Retorna um enumerador.

**Sintaxe**

GetEnvelopeEnumerator()

**Parâmetro**

-

**Valor de retorno**

Enumerador com estes métodos:

Método	Função
Current	Retorna o objeto atual.
MoveNext	Move a posição até o próximo objeto a ser processado.

	<table border="1"> <tr> <td>Redefinir</td> <td>Redefine o enumerador de volta ao início do primeiro elemento.</td> </tr> </table>	Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.						
Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.								
CurrentStack	<p>Pilha atual (ReadOnly)</p> <p><b>Métodos</b></p> <p>GetValue</p> <p>Retorna o valor da coluna dada.</p> <p><b>Sintaxe</b></p> <p>GetValue(string columnName)</p> <table border="1"> <thead> <tr> <th>Parâmetro</th> <th>Significado</th> </tr> </thead> <tbody> <tr> <td>columnName</td> <td>Nome da coluna</td> </tr> </tbody> </table> <p><b>Valor de retorno</b></p> <p>Valor da coluna</p> <p>Se o tipo de dados da coluna do banco de dados for DATE, Datetime ou qualquer tipo de dados válido, a data lida geralmente será retornada no formato MM/DD/AAAA hh:mm:ss.</p> <p><b>Exemplo</b></p> <pre>GetObject("DateVar", "MLDate") ... StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT") ;Converter a data no formato desejado com a ajuda do objeto MLDate IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy hh:mm:ss", StringDateVar ) v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy hh:mm:ss ", IntDateVar)</pre> <p>GetFieldNameEnumerator</p> <p>Retorna um enumerador.</p> <p><b>Sintaxe</b></p> <p>GetEnvelopeEnumerator()</p> <p><b>Parâmetro</b></p> <p>-</p> <p><b>Valor de retorno</b></p> <p>Enumerador com estes métodos:</p> <table border="1"> <thead> <tr> <th>Método</th> <th>Função</th> </tr> </thead> <tbody> <tr> <td>Current</td> <td>Retorna o objeto atual.</td> </tr> </tbody> </table>	Parâmetro	Significado	columnName	Nome da coluna	Método	Função	Current	Retorna o objeto atual.
Parâmetro	Significado								
columnName	Nome da coluna								
Método	Função								
Current	Retorna o objeto atual.								

	MoveNext	Move a posição até o próximo objeto a ser processado.
	Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.

## Métodos do contexto OdinDbTask

### ***HasData***

Verifica se os dados existem para o tipo determinado.

### **Sintaxe**

HasData(OdinDataRowType dataType)

### **Parâmetro**

Parâmetro	Significado	Valores possíveis
DataType	Tipo de dados	OdinDataRowType.Process OdinDataRowType.Stack OdinDataRowType.Envelope OdinDataRowType.Job OdinDataRowType.VirtualJob

### **Valor de retorno**

True: A linha de dados especificada existe.

False: A linha de dados especificada não existe.

## Contexto OdinEmail

### **Geral**

Contexto para o envio de emails.

O contexto fornece scripts com um objeto que pode ser acessado por **Email** para definir propriedades de um email.

As propriedades de email do host SMTP, já definidas na impressora de email, são definidas automaticamente para o objeto de email.

O destinatário (**MailTo**) é recuperado do campo **ODJ\_EMAIL\_TO** do trabalho Odin.

O método **Send()** é chamado automaticamente pela impressora de email. Apenas as propriedades obrigatórias precisam ser definidas no script.

**As entradas a seguir devem ser feitas para que um email seja enviado com êxito:**

- MailFrom

- MailTo
- MailSubject
- BodyText ou BodyHtml

**Para OdinSmtpDeliveryMethod.Network:**

- SmtpHost e SmtpPort
- SmtpUserName e SmtpPassword

**Para OdinSmtpDeliveryMethod.PickupDirectory:**

- SmtpPickupDirectory

**Os endereços de email são inseridos da maneira a seguir:**

"DisplayName" <email@domain>

**Endereços de email separados por vírgulas são inseridos da maneira a seguir:**

- "DisplayName" <email@domain>,<email@domain>,email@domain  
por exemplo: "displayName-1" <email-1@domain-1.de>,"displayName-N" <email-N@domain-N.de>,<email-X@domain-X.de>,email-Y@domain-Y.de
- Em formato de lista de string: DisplayName=endereço de email ou apenas endereço de email (sem DisplayName)  
por exemplo: displayName-1=email-1@domain-1.de,displayName-N=email-N@domain-N.de  
Info=email@test.de,email@test,Kontakt=email2@test.de

**Propriedades do contexto OdinEmail**

Propriedade	Significado
SmtpHost	Host no qual o servidor SMTP está em execução.
SmtpPort	Porta do servidor SMTP.
SmtpUserName	Nome de usuário para a autenticação do servidor SMTP.
SmtpPassword	Senha para autenticação do servidor STPM. A senha deve ser codificada. Para codificar a senha, use a ferramenta <b>Encoder.exe</b> no diretório bin.
SmtpDomain	Domínio/computador no qual as informações de login serão verificadas.
UseDefaultCredentials	True: Autenticação do servidor SMTP usando o usuário conectado. False: Autenticação do servidor SMTP usando SmtpUserName / SmtpPassword.



OdinSmtpDeliveryMethod	<p>Especifica quantos emails serão enviados.</p> <p><b>Valores possíveis:</b></p> <table border="1"> <tr> <td data-bbox="592 401 1182 625">OdinSmtpDeliveryMethod.Network</td> <td data-bbox="1182 401 1424 625">Os emails são enviados por meio do servidor SMTP configurado (SmptHost / Smtport).</td> </tr> <tr> <td data-bbox="592 625 1182 940">OdinSmtpDeliveryMethod.SpecifiedPickupDirectory</td> <td data-bbox="1182 625 1424 940">Os emails são copiados para o diretório especificado na propriedade Smtppickupdirectory e devem ser enviados por um aplicativo externo.</td> </tr> <tr> <td data-bbox="592 940 1182 1163">OdinSmtpDeliveryMethod.PickupDirectoryFromIIS</td> <td data-bbox="1182 940 1424 1163">Os emails são copiados para o diretório configurado no Internet Information Server (IIS).</td> </tr> </table>	OdinSmtpDeliveryMethod.Network	Os emails são enviados por meio do servidor SMTP configurado (SmptHost / Smtport).	OdinSmtpDeliveryMethod.SpecifiedPickupDirectory	Os emails são copiados para o diretório especificado na propriedade Smtppickupdirectory e devem ser enviados por um aplicativo externo.	OdinSmtpDeliveryMethod.PickupDirectoryFromIIS	Os emails são copiados para o diretório configurado no Internet Information Server (IIS).						
OdinSmtpDeliveryMethod.Network	Os emails são enviados por meio do servidor SMTP configurado (SmptHost / Smtport).												
OdinSmtpDeliveryMethod.SpecifiedPickupDirectory	Os emails são copiados para o diretório especificado na propriedade Smtppickupdirectory e devem ser enviados por um aplicativo externo.												
OdinSmtpDeliveryMethod.PickupDirectoryFromIIS	Os emails são copiados para o diretório configurado no Internet Information Server (IIS).												
OdinDeliveryNotificationOptions	<p>Define as opções de notificação para emails.</p> <p><b>Valores possíveis:</b></p> <table border="1"> <tr> <td data-bbox="592 1287 1122 1444">OdinDeliveryNotificationOptions.Network</td> <td data-bbox="1122 1287 1424 1444">Os emails são enviados por meio do servidor SMTP configurado (SmptHost / Smtport).</td> </tr> <tr> <td data-bbox="592 1444 1122 1518">OdinDeliveryNotificationOptions.None</td> <td data-bbox="1122 1444 1424 1518">Sem notificação.</td> </tr> <tr> <td data-bbox="592 1518 1122 1623">OdinDeliveryNotificationOptions.OnSuccess</td> <td data-bbox="1122 1518 1424 1623">Notificação das entregas com êxito.</td> </tr> <tr> <td data-bbox="592 1623 1122 1728">OdinDeliveryNotificationOptions.OnFailure</td> <td data-bbox="1122 1623 1424 1728">Notificação das entregas sem êxito.</td> </tr> <tr> <td data-bbox="592 1728 1122 1822">OdinDeliveryNotificationOptions.Delay</td> <td data-bbox="1122 1728 1424 1822">Notificação das entregas atrasadas.</td> </tr> <tr> <td data-bbox="592 1822 1122 1877">OdinDeliveryNotificationOptions.Never</td> <td data-bbox="1122 1822 1424 1877">Sem qualquer</td> </tr> </table>	OdinDeliveryNotificationOptions.Network	Os emails são enviados por meio do servidor SMTP configurado (SmptHost / Smtport).	OdinDeliveryNotificationOptions.None	Sem notificação.	OdinDeliveryNotificationOptions.OnSuccess	Notificação das entregas com êxito.	OdinDeliveryNotificationOptions.OnFailure	Notificação das entregas sem êxito.	OdinDeliveryNotificationOptions.Delay	Notificação das entregas atrasadas.	OdinDeliveryNotificationOptions.Never	Sem qualquer
OdinDeliveryNotificationOptions.Network	Os emails são enviados por meio do servidor SMTP configurado (SmptHost / Smtport).												
OdinDeliveryNotificationOptions.None	Sem notificação.												
OdinDeliveryNotificationOptions.OnSuccess	Notificação das entregas com êxito.												
OdinDeliveryNotificationOptions.OnFailure	Notificação das entregas sem êxito.												
OdinDeliveryNotificationOptions.Delay	Notificação das entregas atrasadas.												
OdinDeliveryNotificationOptions.Never	Sem qualquer												

		notificação.																		
OdinMailPriority	<p>Prioridade de email</p> <p><b>Valores possíveis:</b></p> <table border="1"> <tr> <td>OdinMailPriority.Normal</td> <td>O email tem prioridade normal.</td> </tr> <tr> <td>OdinMailPriority.Low</td> <td>O email tem baixa prioridade.</td> </tr> <tr> <td>OdinMailPriority.High</td> <td>O email tem alta prioridade.</td> </tr> </table>		OdinMailPriority.Normal	O email tem prioridade normal.	OdinMailPriority.Low	O email tem baixa prioridade.	OdinMailPriority.High	O email tem alta prioridade.												
OdinMailPriority.Normal	O email tem prioridade normal.																			
OdinMailPriority.Low	O email tem baixa prioridade.																			
OdinMailPriority.High	O email tem alta prioridade.																			
IsBodyHtml	<p>Define se o texto do corpo do email está no formato HTML.</p> <p>True: BodyHtml é usado como texto HTML e BodyText como alternativa de texto simples.</p> <p>False: Somente o BodyText é usado como texto simples.</p>																			
MailEncoding	<p>O texto a seguir pode ser usado como parâmetro de codificação:</p> <ul style="list-style-type: none"> <li>• String vazia. Nesse caso, a codificação padrão (página de código ANSI do sistema operacional) é usada.</li> <li>• página de código</li> <li>• name</li> <li>• nome amigável</li> </ul> <p>Exemplo:</p> <ul style="list-style-type: none"> <li>• "1252"</li> <li>• "windows-1252"</li> <li>• "Europeu Ocidental" (Windows)</li> </ul> <p>Possíveis parâmetros de codificação</p> <table border="1"> <thead> <tr> <th>Página de código</th> <th>Nome</th> <th>Nome para exibição</th> </tr> </thead> <tbody> <tr> <td>37</td> <td>IBM037</td> <td>IBM EBCDIC (EUA-Canadá)</td> </tr> <tr> <td>437</td> <td>IBM437</td> <td>OEM EUA</td> </tr> <tr> <td>500</td> <td>IBM500</td> <td>IBM EBCDIC (Internacional)</td> </tr> <tr> <td>708</td> <td>ASMO-708</td> <td>Árabe (ASMO 708)</td> </tr> <tr> <td>720</td> <td>DOS-720</td> <td>Árabe (DOS)</td> </tr> </tbody> </table>		Página de código	Nome	Nome para exibição	37	IBM037	IBM EBCDIC (EUA-Canadá)	437	IBM437	OEM EUA	500	IBM500	IBM EBCDIC (Internacional)	708	ASMO-708	Árabe (ASMO 708)	720	DOS-720	Árabe (DOS)
Página de código	Nome	Nome para exibição																		
37	IBM037	IBM EBCDIC (EUA-Canadá)																		
437	IBM437	OEM EUA																		
500	IBM500	IBM EBCDIC (Internacional)																		
708	ASMO-708	Árabe (ASMO 708)																		
720	DOS-720	Árabe (DOS)																		

737	ibm737	Grego (DOS)
775	ibm775	Báltico (DOS)
850	ibm850	Europeu Ocidental (DOS)
852	ibm852	Europeu Oriental (DOS)
855	IBM855	OEM Cirílico
857	ibm857	Turco (DOS)
858	IBM00858	OEM Multilíngue Latino I
860	IBM860	Português (DOS)
861	ibm861	Islandês (DOS)
862	DOS-862	Hebraico (DOS)
863	IBM863	Francês do Canadá (DOS)
864	IBM864	Árabe (864)
865	IBM865	Nórdico (DOS)
866	cp866	Cirílico (DOS)
869	ibm869	Grego moderno (DOS)
870	IBM870	IBM EBCDIC (Multilíngue Latino 2)
874	windows-874	Tailandês (Windows)
875	cp875	IBM EBCDIC (Grego moderno)
932	shift_jis	Japonês (Shift-JIS)
936	gb2312	Chinês simplificado (GB2312)

949	ks_c_5601-1987	Coreano
950	big5	Chinês (tradicional) (Big5)
1026	IBM1026	IBM EBCDIC (Turco, Latino 5)
1047	IBM01047	IBM Latino 1
1140	IBM01140	IBM EBCDIC (EUA-Canadá-Europeu)
1141	IBM01141	IBM EBCDIC (Alemanha-Europeu)
1142	IBM01142	IBM EBCDIC (Dinamarca-Noruega-Europeu)
1143	IBM01143	IBM EBCDIC (Finlândia-Suécia-Europeu)
1144	IBM01144	IBM EBCDIC (Italiano-Europeu)
1145	IBM01145	IBM EBCDIC (Espanha-Europeu)
1146	IBM01146	IBM EBCDIC (Grã-Bretanha-Europeu)
1147	IBM01147	IBM EBCDIC (França-Europeu)
1148	IBM01148	IBM EBCDIC (Internacional-Europeu)
1149	IBM01149	IBM EBCDIC (Islandês-Europeu)
1200	utf-16	Unicode
1201	unicodeFFFE	Unicode (Big Endian)
1250	windows-1250	Centro-europeu (Windows)

1251	windows-1251	Cirílico (Windows)
1252	Windows-1252	Europeu Ocidental (Windows)
1253	windows-1253	Grego (Windows)
1254	windows-1254	Turco (Windows)
1255	windows-1255	Hebraico (Windows)
1256	windows-1256	Árabe (Windows)
1257	windows-1257	Báltico (Windows)
1258	windows-1258	Vietnamita (Windows)
1361	Johab	Coreano (Johab)
10000	macintosh	Europeu Ocidental (Mac)
10001	x-mac-japanese	Japonês (Mac)
10002	x-mac-chinesetrad	Chinês tradicional (Mac)
10003	x-mac-korean	Coreano (Mac)
10004	x mac-arabic	Árabe (Mac)
10005	x-mac-hebrew	Hebraico (Mac)
10006	x-mac-greek	Grego (Mac)
10007	x-mac-cyrillic	Cirílico (Mac)
10008	x-mac-chinesesimp	Chinês simplificado (Mac)
10010	x-mac-romanian	Romeno (Mac)
10017	x-mac-ukrainian	Ucraniano (Mac)

10021	x-mac-thai	Tailandês (Mac)
10029	x-mac-ce	Centro-europeu (Mac)
10079	x-mac-icelandic	Islandês (Mac)
10081	x-mac-turkish	Turco (Mac)
10082	x-mac-croatian	Croata (Mac)
12000	utf-32	Unicode (UTF-32)
12001	utf-32BE	Unicode (UTF-32-Big Endian)
20000	x-Chinese-CNS	Chinês tradicional (CNS)
20001	x-cp20001	TCA Taiwan
20002	x-Chinese-Eten	Chinês tradicional (Eten)
20003	x-cp20003	IBM5550 Taiwan
20004	x-cp20004	TeleText Taiwan
20005	x-cp20005	Wang Taiwan
20105	x-IA5	Europeu Ocidental (IA5)
20106	x-IA5-German	Alemão (IA5)
20107	x-IA5-Swedish	Sueco (IA5)
20108	x-IA5-Norwegian	Norueguês (IA5)
20127	us-ascii	US-ASCII
20261	x-cp20261	T.61
20269	x-cp20269	ISO-6937
20273	IBM273	IBM EBCDIC (Alemanha)
20277	IBM277	IBM EBCDIC (Dinamarca-

		Noruega)
20278	IBM278	IBM EBCDIC (Finlândia-Suécia)
20280	IBM280	IBM EBCDIC (Itália)
20284	IBM284	IBM EBCDIC (Espanha)
20285	IBM285	IBM EBCDIC (Grã-Bretanha)
20290	IBM290	IBM EBCDIC (Japonês katakana)
20297	IBM297	IBM EBCDIC (França)
20420	IBM420	IBM EBCDIC (Árabe)
20423	IBM423	IBM EBCDIC (Grego)
20424	IBM424	IBM EBCDIC (Hebraico)
20833	x-EBCDIC-KoreanExtended	IBM EBCDIC (Coreano estendido)
20838	IBM-Thai	IBM EBCDIC (Tailandês)
20866	koi8-r	Cirílico (KOI8-R)
20871	IBM871	IBM EBCDIC (Islandês)
20880	IBM880	IBM EBCDIC (Cirílico russo)
20905	IBM905	IBM EBCDIC (Turco)
20924	IBM00924	IBM Latino 1
20932	EUC-JP	Japonês (JIS 0208-1990 e 0212-1990)
20936	x-cp20936	GB2312-80 Chinês (simplificado)
20949	x-cp20949	Coreano Wansung

21025	cp1025	IBM EBCDIC (Cirílico servo-búlgaro)
21866	koi8-u	Cirílico (KOI8-U)
28591	iso-8859-1	Europeu Ocidental (ISO)
28592	iso-8859-2	Centro-europeu (ISO)
28593	iso-8859-3	Latino 3 (ISO)
28594	iso-8859-4	Báltico (ISO)
28595	iso-8859-5	Cirílico (ISO)
28596	iso-8859-6	Árabe (ISO)
28597	iso-8859-7	Grego (ISO)
28598	iso-8859-8	Hebraico (ISO-Visual)
28599	iso-8859-9	Turco (ISO)
28603	iso-8859-13	Estoniano (ISO)
28605	iso-8859-15	Latino 9 (ISO)
29001	x-Europa	Europa
38598	iso-8859-8-i	Hebraico (ISO-lógico)
50220	iso-2022-jp	Japonês (JIS)
50221	csISO2022JP	Japonês (JIS, 1 byte kana permitido)
50222	iso-2022-jp	Japonês (JIS, 1 byte kana permitido-SO/SI)
50225	iso-2022-kr	Coreano (ISO)
50227	x-cp50227	ISO-2022 Chinês simplificado



	51932	euc-jp	Japonês (EUC)
	51936	EUC-CN	Chinês simplificado (EUC)
	51949	euc-kr	Coreano (EUC)
	52936	hz-gb-2312	Chinês simplificado (HZ)
	54936	GB18030	GB18030 Chinês simplificado
	57002	x-iscii-de	ISCII Devanágari
	57003	x-iscii-be	ISCII Bengalês
	57004	x-iscii-ta	ISCII Tâmil
	57005	x-iscii-te	ISCII Télego
	57006	x-iscii-as	ISCII Assamês
	57007	x-iscii-or	ISCII Oriá
	57008	x-iscii-ka	ISCII Kannada
	57009	x-iscii-ma	ISCII Malaiala
	57010	x-iscii-gu	ISCII Guzerate
	57011	x-iscii-pa	ISCII Punjabi
	65000	utf-7	Unicode (UTF-7)
	65001	utf-8	Unicode (UTF-8)
MailEncodingCodepage	Codificação a ser usada para o email.		
MailSubject	Assunto do email		
BodyText	Define ou obtém o texto do email em um texto simples.		
BodyHtml	Define ou obtém o texto do email em um formato HTML.		

BodyHtmlImages	Retorna ou define a string de BodyHtmlImages separada por vírgula.
SmtppickupDirectory	Retorna ou define o diretório no qual os emails são salvos, se o SmtppickupMethod.SpecifiedPickupDirectory é definido.
MailFrom	Retorna ou define o remetente do email.
MailTo	Define ou retorna o destinatário do email (separado por vírgula).
MailCC	Retorna ou define o destinatário Cc do email (separado por vírgula).
MailBcc	Retorna ou define o destinatário Cco do email (separado por vírgula).
MailReplyTo	Retorna ou define o endereço ReplyTo do email.
AttachmentCount	Retorna o número de anexos do email.

## Métodos do contexto OdinEmail

### ***AddAttachment***

Adiciona um anexo a um email.

#### **Sintaxe**

AddAttachment(string fileName)

#### **Parâmetro**

Parâmetro	Significado
filename	Nome do arquivo anexado.

#### **Valor de retorno**

-

### ***AddBodyHtmlImage***

Adiciona uma imagem ao corpo HTML.

#### **Sintaxe**

AddBodyHtmlImage(string id, string imageFileName)

#### **Parâmetro**

Parâmetro	Significado
-----------	-------------

id	ID da imagem (cid). Se nenhum ID for especificado, será usado o nome do arquivo sem a extensão.
imageFileName	Nome do arquivo da imagem

**Valor de retorno**

-

**AddToHeader**

Adiciona uma entrada de nome/valor a um cabeçalho de email.

**Sintaxe**

AddToHeader(string name, string value)

**Parâmetro**

Parâmetro	Significado
name	Nome da entrada. Se a entrada já existir, seu valor é sobrescrito. Em caso contrário, uma nova entrada é criada.
value	Valor da entrada.

**Valor de retorno**

-

**Exemplo**

```
Email.AddToHeader("Organization", "Perceptive Software")
```

**ClearAttachments**

Exclui todos os anexos carregados.

**Sintaxe**

ClearAttachments()

**Parâmetro**

-

**Valor de retorno**

-

### ***ClearBodyHtmlImages***

Exclui todas as imagens carregadas no corpo HTML.

#### **Sintaxe**

ClearBodyHtmlImages()

#### **Parâmetro**

-

#### **Valor de retorno**

-

### ***ClearCertificates***

Exclui todos os certificados carregados/definidos.

#### **Sintaxe**

ClearCertificates()

#### **Parâmetro**

#### **Valor de retorno**

-

### ***GetAttachments***

Retorna uma string separada por vírgulas contendo os nomes de arquivo de todos os anexos.

#### **Sintaxe**

GetAttachments()

#### **Parâmetro**

-

#### **Valor de retorno**

-

#### **Exemplo**

### ***LoadBodyFromFile***

Carrega uma caixa de texto a partir de um arquivo.

**Sintaxe**

LoadBodyFromFile(string fileName, bool isHtml)

**Parâmetro**

Parâmetro	Significado
fileName	Nome do arquivo
isHtml	True / False

**Valor de retorno**

-

***LoadCertificateFromCertFile***

Carrega o certificado X.509 do arquivo especificado para enviar o e-mail.

**Sintaxe**

LoadCertificateFromCertFile(string fileName)

**Parâmetro**

Parâmetro	Significado
fileName	Nome do arquivo

**Valor de retorno**

-

***LoadCertificateFromSignedFile***

Carrega o certificado X.509 do arquivo especificado para enviar o e-mail.

**Sintaxe**

LoadCertificateFromSignedFile(string fileName)

**Parâmetro**

Parâmetro	Significado
fileName	Nome do arquivo

**Valor de retorno**

-

***RemoveAttachment***

Remove um anexo de um email.

**Sintaxe**

RemoveAttachment(string fileName)

**Parâmetro**

Parâmetro	Significado
fileName	Nome do arquivo a ser removido dos anexos.

**Valor de retorno**

-

***RemoveFromHeader***

Remove uma entrada de um cabeçalho de email.

**Sintaxe**

RemoveFromHeader(string name)

**Parâmetro**

Parâmetro	Significado
name	Nome da entrada a ser removida.

**Valor de retorno**

-

**Exemplo**

```
Email.RemoveFromHeader("Organisation")
```

***Send***

Envia o email com as configurações atuais.

**Sintaxe**

Send()

**Parâmetro**

-

**Valor de retorno**

-

***SetCredentials***

Define as credenciais de usuário (nome de usuário, senha e domínio).

**Sintaxe**

```
SetCredentials(string userName, string password, string domain)
```

**Parâmetro**

Parâmetro	Significado
userName	Nome de usuário
password	Senha A senha deve ser codificada. Para codificar a senha, use a ferramenta <b>Encoder.exe</b> no diretório bin.
domain	Domínio

**Valor de retorno**

-

***SetSmtpServer***

Define as propriedades do servidor SMTP para host, porta e UseDefaultCredentials.

**Sintaxe**

```
SetSmtpServer(string host, int port, bool useDefaultCredentials)
```

**Parâmetro**

Parâmetro	Significado
host	Host no qual o servidor SMTP está em execução.
port	Porta do servidor SMTP.
useDefaultCredentials	True: Autenticar com o servidor SMTP usando o usuário conectado. False: Autenticar com o servidor SMTP usando SmptUserName /

Smtppassword.

### Valor de retorno

-

### Script de exemplo

```
; Define email settings such as sender, priority, etc.
Email.MailFrom = ""info" <info@perceptivesoftware.com>"
Email.MailPriority = OdinMailPriority.High
;
; Was a document converted to body text?
if ((Email.BodyHtml = "") or (Email.BodyText = ""))
    ; No, then set the body text as plain text
    Email.BodyText = "this is the body text!!"
    Email.IsBodyHtml = false
;
end-if
;
; Was a document converted to the subject?
if (Email.MailSubject = "")
    ; No, so set a subject
    Email.MailSubject = "subject line"
end-if
```

## Contexto OdinEmailConvert

### Propriedades do contexto OdinEmailConvert

[▶ Mostrar tudo](#)

Propriedade	Significado				
ConvertOptions	<p>String separada por vírgula contendo os metadados e opções a serem usadas durante a conversão.</p> <p>Essas opções também podem ser definidas no evento OnConvertStart para que os dados de pilha, por exemplo, possam ser usados para definir os metadados.</p> <p><b>Metadados</b></p> <table border="1"> <tbody> <tr> <td>TITLE</td> <td>Título do documento.</td> </tr> <tr> <td>SUBJECT</td> <td>Assunto do documento.</td> </tr> </tbody> </table>	TITLE	Título do documento.	SUBJECT	Assunto do documento.
TITLE	Título do documento.				
SUBJECT	Assunto do documento.				



AUTHOR	Autor do documento.
KEYWORDS	Palavras-chave do documento.

**Opções**

TEXTNOTE_CHANGES	(Pdf 1.3/1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de segurança de PDF para Montagem de documento, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como verdadeiro/falso.
CHANGES	(Pdf 1.3/1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de segurança de PDF para Montagem de documento, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como verdadeiro/falso.
FILL	(Pdf 1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de segurança de PDF para Montagem de documento, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como verdadeiro/falso.
COPY	(Pdf 1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de segurança de PDF para Montagem de documento, Conteúdo copiado, Conteúdo copiado para acessibilidade, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como verdadeiro/falso.
DIGITAL_COPY	(Pdf 1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de segurança de PDF para Montagem de documento, Extração de página, Comentário, Assinatura e Criar

	modelo serão definidos como verdadeiro/falso.
ASSEMBLE	(Pdf 1.4) TRUE/FALSE A montagem de um ou mais documentos não é permitida. Com a especificação deste parâmetro, os atributos de segurança de PDF para Montagem de documento, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como verdadeiro/falso.
ACCESSIBILITY	(Pdf 1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de segurança de PDF para Conteúdo copiado, Conteúdo copiado para acessibilidade, Extração de página serão definidos como verdadeiro/falso.
DUPLEX_ADDPAGE	TRUE: Esta opção especifica se uma página vazia deve ser adicionada a um documento duplex com um número ímpar de páginas. FALSE: Configuração padrão. Esta opção define o modo simplex para a última página de um documento duplex com um número ímpar de páginas.
DUPLEX_OPTIMIZE	TRUE/FALSE Quando verdadeiro, as configurações de duplex são definidas apenas no fluxo de saída, quando o modo duplex muda.
MAP_COLOR	TRUE/FALSE Este parâmetro é usado para definir se o mapeamento de cores da definição AFP-HighlightColor será executado ou não. As definições de cores individuais podem ser especificadas como MapColor_1 até MapColor_n. <b>Exemplo:</b> Map_Color=TRUE,"MapColor_1=-1,20,black","MapColor_2=0,20,red" <b>Observação:</b> Como os parâmetros MapColor contêm valores

		<p>separados por vírgulas, a string completa deve estar entre aspas.</p> <p><b>A estrutura é a seguinte:</b></p> <p>MapColor_???= HSB-Color,Threshold,ColorName</p> <p>HSB-Color=Ângulo da cor HSB</p> <p>Threshold= Valor limite da cor HSB</p> <p>ColorName= Nome da cor como definido no perfil AFP, em HighlightColorList.</p> <p><b>O exemplo acima requer estas definições no perfil AFP:</b></p> <pre> &lt;/highlightcolorlist&gt;   &lt;highlightcolor name='red' type='OCA' number='0x0002' foreground-rgb="#00FF0000" /&gt;   &lt;highlightcolor name='black' type='OCA' number='0x0008' foreground-rgb="#00000000" /&gt; &lt;/highlightcolorlist&gt; </pre> <p>e</p> <pre> &lt;optimizepa value="never"/&gt; &lt;color value="FALSE" /&gt; &lt;usehighlightcolor value="TRUE" /&gt; &lt;enablebilevelimage value="TRUE" /&gt; </pre>
	SHOW_BOOKMARKS	<p>TRUE: Os bookmarks são mostrados quando o arquivo PDF é aberto.</p> <p>FALSE: Os bookmarks não são mostrados quando o arquivo PDF é aberto.</p>
	OVERLAYS	<p>TRUE: São criadas OVERLAYS.</p> <p>FALSE: Padrão. Não são criadas OVERLAYS.</p>
	MERGEFONTS	<p>TRUE: Padrão</p> <p>FALSE</p>
	MERGE_SPLITWORDS	Ativa e desativa a otimização da saída PCL.

		<p>Defina esta opção como FALSE se você tiver problemas de espaçamento de caracteres em um PCL gerado.</p> <p>A configuração padrão é TRUE.</p> <p>TRUE: Padrão, a otimização da saída PCL é ativada.</p> <p>FALSE: A otimização da saída PCL é desativada.</p>						
CurrentDocument	<p>Retorna o documento atual (IOdinStreamDocument-Interface) a ser convertido.</p> <p>Métodos do IOdinStreamDocument-Interface</p> <p>AddBookmark</p> <p><a href="#">EditPage</a></p> <p><a href="#">GetInputBin</a></p> <p><a href="#">Image</a></p> <p><a href="#">IsDuplex</a></p> <p><a href="#">IsPortrait</a></p> <p><a href="#">Line</a></p> <p><a href="#">PageCount</a></p> <p><a href="#">Post</a></p> <p><a href="#">Rectangle</a></p> <p><a href="#">SetFont</a></p> <p><a href="#">SetInputBin</a></p> <p><a href="#">TextOut</a></p>							
InputBinName	<p>Retorna o nome da bandeja que está sendo usada para converter ou imprimir o documento.</p>							
InputBinType	<p>Essa propriedade corresponde à propriedade BinType da bandeja de email.</p> <p>Tipo de bandeja que está sendo usada para converter o documento.</p> <p><b>Valores possíveis:</b></p> <table border="1"> <thead> <tr> <th>Valor</th> <th>Significado</th> </tr> </thead> <tbody> <tr> <td>OdinEmailPrinterBinType.Attachment</td> <td>O documento é convertido ao formato de saída correspondente (por exemplo, PDF) e, em seguida, adicionado ao email como um anexo separado.</td> </tr> <tr> <td>OdinEmailPrinterBinType.AttachmentCollect</td> <td>Todos os documentos são convertidos a um único</td> </tr> </tbody> </table>	Valor	Significado	OdinEmailPrinterBinType.Attachment	O documento é convertido ao formato de saída correspondente (por exemplo, PDF) e, em seguida, adicionado ao email como um anexo separado.	OdinEmailPrinterBinType.AttachmentCollect	Todos os documentos são convertidos a um único	
Valor	Significado							
OdinEmailPrinterBinType.Attachment	O documento é convertido ao formato de saída correspondente (por exemplo, PDF) e, em seguida, adicionado ao email como um anexo separado.							
OdinEmailPrinterBinType.AttachmentCollect	Todos os documentos são convertidos a um único							

		arquivo anexado no formato de saída correspondente.
	OdinEmailPrinterBinType.Body	O documento é convertido ao texto ou formato HTML e usado como o corpo do texto do email.
	OdinEmailPrinterBinType.Subject	Permite que uma única linha de documento seja convertida a um texto de referência individual do email.  O documento é convertido a um formato de texto e usado como o texto do assunto.
OutputFileName	Retorna ou define o nome do anexo.  O nome de arquivo padrão do documento convertido é a ID do trabalho com uma extensão correspondente ao formato de saída definido pela bandeja.  Esse nome pode ser alterado, mas qualquer especificação e extensão de caminho será ignorada.	

## Métodos do contexto OdinEmailConvert

### **SetPdfSigner**

O evento **OnConvert** de uma impressora de email é chamado para cada documento a ser convertido.

O método **SetPdfSigner()** do contexto **OdinEmailConvert** pode passar um objeto PDF-Signer inicializado, que pode ser usado para assinar um documento quando o formato de saída for PDF.

Se for definida uma senha em **ConvertOptions**, o objeto PDF-Signer deve ser inicializado com a senha ou o acesso ao PDF não será concedido.

### Sintaxe

```
SetPdfSigner(IOdinPdfSigner pdfSigner)
```

### Parâmetro

Parâmetro	Significado
pdfSigner	O objeto PDF-Signer inicializado criado pelo componente OdinFileConvert. Consulte: <a href="#">CreatePdfSigner</a>

**Valor de retorno**

-

**Métodos da interface IOdinStreamDocument*****AddBookmark***

Adiciona um bookmark a um documento. Se o formato de saída for PDF, um bookmark é criado para cada bookmark definido.

As IDs deve ser exclusivas para cada bookmark.

**Sintaxe**

```
AddBookmark(string id, string parentId, string title)
```

```
AddBookmark(string id, string parentId, string title, double x, double y)
```

**Parâmetro**

Parâmetro	Significado
id	ID do Bookmark
parentId	Uma string vazia aqui significa que o bookmark será inserido no nível mais elevado. Caso contrário, o bookmark é inserido como um filho do bookmark especificado.
title	Texto para o bookmark
x	Posição X na página
y	Posição Y na página

**Valor de retorno**

-

**Exemplo**

```
document.AddBookmark("doc1_page1", "doc1", "Page 1")
document.AddBookmark("doc1_page2", "doc1", "Page 2", 0, 0)
```

***EditPage***

EditPage define a página determinada para o modo de edição. Os comandos de caractere agora podem ser executados na página.

**Sintaxe**

```
EditPage(int pageNumber)
```

**Parâmetro**

Parâmetro	Significado
pageNumber	Número da página a ser processada, começando com 0.

**Valor de retorno**

-

***GetInputBin***

Retorna o recipiente de entrada de uma página.

Se não estiver definido, retorna a bandeja da primeira página ou a bandeja subsequente, dependendo do número da página.

**Sintaxe**

```
GetInputBin(int pageNumber)
```

**Parâmetro**

Parâmetro	Significado
pageNumber	Número da página

**Valor de retorno**

recipiente de entrada

***Imagem***

Insere um arquivo de imagem.

**Sintaxe**

```
Image(double x, double y, string fileName)
```

```
Image(double x, double y, string imageFileName, OdinPageLayer layer)
```

**Parâmetro**

Parâmetro	Significado
x	Posição X
y	Posição Y
filename	Nome do arquivo
layer	<b>Valores possíveis:</b>

OdinPageLayer.Foreground: O desenho é feito por cima do documento original.  
 OdinPageLayer.Background: O desenho é feito por baixo do documento original.

**Valor de retorno**

-

**Exemplo*****IsDuplex***

Retorna ao modo duplex.

**Sintaxe**

IsDuplex()

**Parâmetro****Valor de retorno**

True: Duplex

False: Simplex

***IsPortrait***

Retorna ao modo retrato.

**Sintaxe**

IsPortrait(int pageNumber)

**Parâmetro**

Parâmetro	Significado
pageNumber	Número da página

**Valor de retorno**

True: Retrato

False: Paisagem

***Line***

Desenha uma linha na página atual.



**Sintaxe**

Line(double x1, double y1, double x2, double y2)

Line(double x1, double y1, double x2, double y2, double lineWidth, OdinLineStyle lineStyle, string color)

Line(double x1, double y1, double x2, double y2, double lineWidth, OdinLineStyle lineStyle, string color, OdinPageLayer layer)

**Parâmetro**

Parâmetro	Significado
x1	Posição X da posição inicial da linha.
y1	Posição Y da posição inicial da linha.
x2	Posição X da posição final da linha.
y2	Posição Y da posição final da linha.
lineWidth	Largura de linha
lineStyle	Valores possíveis OdinLineStyle.Solid OdinLineStyle.Dashed OdinLineStyle.Dotted
color	Cor
Layer	<b>Valores possíveis</b> OdinPageLayer.Foreground OdinPageLayer.Background

**Valor de retorno**

-

***PageCount***

Retorna o número de páginas do documento atual.

**Sintaxe**

PageCount()

**Parâmetro**

-

**Valor de retorno**

Número de páginas.

***Publicar***

Grava comandos de caractere na camada do documento.

**Sintaxe**

Post()

**Parâmetro**

-

**Valor de retorno**

-

***Rectangle***

Desenha um retângulo na página atual.

**Sintaxe**

Rectangle(double x, double y, double width, double height, bool fill, string fillColor)

Rectangle(double x, double y, double width, double height, double lineWidth, OdinLineStyle lineStyle, string lineColor, bool fill, string fillColor)

Rectangle(double x, double y, double width, double height, double lineWidth, OdinLineStyle lineStyle, string lineColor, bool fill, string fillColor, OdinPageLayer layer)

**Parâmetro**

Parâmetro	Significado
x	Posição X do canto superior esquerdo do retângulo.
y	Posição Y do canto superior esquerdo do retângulo.
width	Largura do retângulo
height	Altura do retângulo
fill	True: o retângulo será preenchido (valor padrão = false)
fillColor	cor de preenchimento
lineWidth	Largura de linha
lineStyle	Valores possíveis

	OdinLineStyle.Solid OdinLineStyle.Dashed OdinLineStyle.Dotted
lineColor	Cor de linha
Layer	Valores possíveis OdinPageLayer.Foreground: O desenho é feito por cima do documento original. OdinPageLayer.Background: O desenho é feito por baixo do documento original.

**Valor de retorno**

-

***SetFont***

Define a especificação de fonte para a saída de texto.

**Sintaxe**

SetFont(string fontName, int fontSize)

**Parâmetro**

Parâmetro	Significado
fontName	Nome da fonte, por exemplo, Arial.
fontSize	Tamanho da fonte, por exemplo, 10.

**Valor de retorno**

-

***SetInputBin***

Define a bandeja de entrada de uma página.

O método substitui a bandeja predefinida para a primeira página ou para as páginas seguintes, conforme o caso.

**Sintaxe**

SetInputBin(int pageNumber, string binName)

**Parâmetro**

Parâmetro	Significado
-----------	-------------

pageNumber	Número da página
binName	Nome da bandeja

**Valor de retorno**

-

**Exemplo**

```
;Set the input tray for the last page of the Document
streamDoc.SetInputBin(streamdoc.PageCount - 1, "testBin")
```

***TextOut***

Escreve um texto na página atual.

**Sintaxe**

TextOut(double x, double y, string text, bool bold, bool italic)

TextOut(double x, double y, string text, bool bold, bool italic, string color, int angle)

TextOut(double x, double y, string text, bool bold, bool italic, string color, int angle, OdinPageLayer layer)

**Parâmetro**

Parâmetro	Significado
x	Posição X do início
y	Posição Y do início
text	Texto a ser escrito
bold	True: Formatação em "negrito" (valor padrão = false)
italic	True: Formatação em "itálico" (valor padrão = false)
color	Cor
angle	Ângulo de rotação
layer	<b>Valores possíveis</b> OdinPageLayer.Foreground: O desenho é feito por cima do documento original. OdinPageLayer.Background: O desenho é feito por baixo do documento original.

**Valor de retorno**

-

## Contexto OdinFilter

### Geral

Contexto para scripts de filtro.

### Métodos FilterObject

#### ***Geral***

A variável **filter** torna um objeto de filtro disponível com os seguintes métodos:

#### ***DefinePostageDefinitionByName***

Define a definição de postagem pelo nome.

#### **Sintaxe**

DefinePostageDefinitionByName(PostageDefinitionName)

#### **Parâmetro**

Parâmetro	Significado
PostageDefinitionName	Nome da definição de postagem

#### **Valor de retorno**

-

#### ***DefinePostageDefinitionByOld***

Define a definição e postagem por sua ID de objeto.

#### **Sintaxe**

DefinePostageDefinitionByOld(postageOld)

#### **Parâmetro**

Parâmetro	Significado
postageOld	ID do objeto da definição de postagem

#### **Valor de retorno**

-

#### ***DefinePrinterByName***

Define uma impressora por seu nome e tipo.

### Sintaxe

DefinePrinterByName(printerName,type)

### Parâmetro

Parâmetro	Significado
printerName	Nome da impressora
type	Tipo da impressora. <b>Os tipos disponíveis são:</b> ModusSuiteObjectType.Printer ModusSuiteObjectType. EMailPrinter ModusSuiteObjectType. OutsourcingPrinter

### Valor de retorno

-

### ***DefinePrinterByOld***

Define a impressora por uso de suas ID de objeto.

### Sintaxe

DefinePrinterByOld(printerOld)

### Parâmetro

Parâmetro	Significado
printerOld	ID do objeto da impressora.

### Valor de retorno

-

### ***EnumEnvelopes***

Invoca uma instância de um objeto de envelope para criar envelopes.

Antes que um objeto de envelope possa ser usado, primeiramente ele deve ser invocado chamando o método **Init**. Se o objeto de envelope não for mais necessário, ele deve ser revogado pela função **FreeObject** no script de filtro.

### Sintaxe

EnumEnvelopes()

### Parâmetro

-

### Valor de retorno

Instância de um objeto de envelope.

### **Gen\_Stack**

Cria uma pilha e atribui o conjunto de dados de trabalho e envelope relevantes a ela. O parâmetro do objeto de filtro 'UPDATE\_SQL' deve ser definido antes que esse método seja chamado. Esse parâmetro contém a instrução de atualização atribuindo os conjuntos de dados de trabalho e envelope à nova pilha.

### Sintaxe

### Parâmetro

Parâmetro	Significado
updateParameter	Contem os pares nome/valor, por exemplo: 'Postage=100'. Se mais de um par for definido, cada par deve ser separado por uma vírgula. Esses pares são usados para substituir com valores reais os espaços reservados na instrução de atualização SQL. Espaços reservados em uma instrução de atualização SQL são identificados por ':'
customStackColumns	<p>Esse parâmetro permite que você insira dados em colunas específicas do cliente na tabela da pilha quando uma pilha é criada. Se não for necessário, uma string vazia deve ser definida.</p> <p>Como UpdateParameter, a string enviada aqui contém pares nome/valor. Por exemplo, para preencher as colunas específicas ao cliente ODS_INFO e ODS_EXT com dados na inserção, os valores</p> <p>'ODS_INFO=SpecialInfo,ODS_EXT=10'</p> <p>devem ser enviados.</p> <p>Se o valor tiver aspas sinais de igual ou vírgulas, o par nome/valor deve ser incluído entre aspas</p> <p><b>Exemplo:</b></p> <p>ODS_CUSTOM1=123,"ODS_CUSTOM2=test,123=0815"</p> <p>Aspas devem ser duplicadas em um script.</p> <p><b>Exemplo:</b></p> <pre>StackResult = Filter.Gen_Stack(UpdateParameter, "ODS_CUSTOM1=123,""ODS_CUSTOM2=test,123=0815"")</pre>

**Valor de retorno**

O resultado aqui é uma estrutura, não o tipo básico de dados, contendo os seguintes atributos:

Parâmetro	Significado
NewStackId	ID do conjunto de dados da nova pilha ou uma string vazia, se nenhuma pilha tiver sido criada (nenhuma tarefa ou envelopes foram atribuídos).
UpdatedRecordCount	O número de conjuntos de dados atualizados pelo Update-SQL. Isso faz referência ao número de conjunto de dados de trabalho ou envelope atribuídos à pilha.

***GetParam***

Recupera o valor de um parâmetro de filtro.

**Sintaxe**

GetParam(ParameterName)

**Parâmetro**

Parâmetro	Significado
ParameterName	Nome do parâmetro de filtro.

**Valor de retorno**

Valor do parâmetro.

***RunFilter***

Permite que você execute outro script de filtro a partir de um script de filtro.

**Sintaxe**

RunFilter(FilterScriptName)

**Parâmetro**

Parâmetro	Significado
FilterScriptName	Nome do script a ser executado.

**Valor de retorno**

-



## ***SetParam***

Define um parâmetro de filtro.

### **Sintaxe**

SetParam(ParameterName, ParameterValue)

### **Parâmetro**

Parâmetro	Significado
ParameterName	Nome do parâmetro de filtro.
ParameterValue	Novo valor do parâmetro de filtro.

### **Valor de retorno**

-

Visão geral dos possíveis parâmetros dos métodos GetParam e SetParam

### **DBALIAS\_NAME**

Lê o nome da alias de banco de dados definida nas configurações do componente OdinFilter.

### **FILTERNAME**

Permite que você defina o valor inserido no campo **ODS\_Filtername** da pilha nova. É considerado durante a criação de uma pilha com **Gen\_Stack**.

### **HOLDTIME**

Permite que você substitua o valor do tempo de impressão padrão de uma pilha nova. É considerado durante a criação de uma pilha com **Gen\_Stack**.

### **MESSAGETEXT**

Permite que você defina o valor inserido no campo **ODS\_Messagetext** da pilha nova. É considerado durante a criação de uma pilha com **Gen\_Stack**.

### **OMR\_TYP**

Permite que você defina o valor inserido no campo **ODS\_OMR\_TYPE** de uma pilha nova. É considerado durante a criação de uma pilha com **Gen\_Stack**.

### **POSTAGEDEF\_STR**

Lê o nome da classe de postagem da definição da postagem definida pelo uso de uma variante de **DefinePostageDefinition**. Todos os nomes devem ser separado por vírgulas.

### **PRIORITY**

Permite que você substitua o valor de prioridade padrão de uma pilha nova. É considerado durante a criação de uma pilha com **Gen\_Stack**.

## STACKOBJECTSQL

Definição opcional de uma chave na tabela de pesquisa **OdinStackObjectSql**.

O valor da chave é uma ordem de instrução a ser executada em vez do SQL padrão enquanto os trabalhos ou envelopes de pilha são carregados.

Essa informação é armazenada no campo **ODJ\_STACKOBJECTSQL** da pilha correspondente criada e é avaliada durante os trabalhos ou envelopes de pilha.

### Exemplo:

Se a chave **Test** é usada pela pilha para carregar os envelopes, as ordens de instruções são recuperadas da tabela de pesquisa **OdinStackObjectSql** nas chaves **Test.Envelope\_OrderBy** e **Test.Job\_OrderBy**.

### Estrutura da tabela de pesquisa OdinStackObjectSQL

2 valores devem ser definidos para cada chave, um para os envelopes e um para os trabalhos:

```
<Key>.Envelope_OrderBy=<ORDERBY-STATEMENT>
<Key>.Job_OrderBy=<ORDERBY-STATEMENT>
```

### Amostra de tabela de pesquisa

```
default.Envelope_OrderBy=ODE_SEQUENCE ASC
default.Job_OrderBy=ODJ_SEQUENCE ASC
Test.Envelope_OrderBy= ODE_REC_NAME DSC
Test.Job_OrderBy=ODJ_SORTING ASC
```

## STACKTYPE

Permite que você substitua o valor padrão do tipo de pilha para uma pilha nova. É considerado durante a criação de uma pilha com **Gen\_Stack**.

## STATE

Permite que você substitua o valor padrão do status da pilha para uma pilha nova. É considerado durante a criação de uma pilha com **Gen\_Stack**.

## UPDATE\_SQL

Este parâmetro define a instrução da atualização do SQL a ser usado para trabalhos / envelopes atribuídos para a nova pilha quando o método **Gen\_Stack** é executado.

### Script de exemplo: Filtro simples incluindo geração de pilha

```
*****
; Filter Script Demo
; simple Filter including Stack generation
;*****
;
filterName = "FilterOnline"
```

```

printerName = "PrinterOnline"
stackState = 0
;
;Get number of open Jobs
DatabaseAliasName = filter.GetParam("DBALIAS_NAME")
InitDataset("OdinJob", DatabaseAliasName)
;
StringListAdd("JobCountSql", "Select count(*) JobCount from Odin_job Where
(ODJ_Stack_Id is null) and (Obj_Env_Id is null)")
SetSQL("OdinJob", "JobCountSql")
OpenSQL("OdinJob")
JobCount = GetField("OdinJob", "JobCount")
;
if (JobCount > 0)
;
;Open Jobs exist -> generate Stack
Protocol("{0} Jobs for processing found", 10, JobCount)
;
; SQL to update open Jobs
UpdateSql =
"Update Odin_Job "+
"Set ODJ_Stack_Id = :StackId, ODJ_Printer_OID = :PrinterOID,
ODJ_Printer_Name = :PrinterName, ODJ_Printer_Type = :PrinterType " +
"Where (ODJ_Stack_Id is null) and (Obj_Env_Id is null) and (ODJ_State =
1)"
;
filter.SetParam("UPDATE_SQL", UpdateSql)
;
filter.SetParam("holdtime", "0")
filter.SetParam("priority", "200")
filter.SetParam("STACKTYPE", "0")
filter.SetParam("OMR_TYP", "")
filter.SetParam("STATE", stackState)
filter.SetParam("filtername", filterName)
;
filter.DefinePrinterByName(printerName, ModusSuiteObjectType.Printer)
;
;Alternative, if information has to be inserted into custom fields of the
stack table
;my_result = filter.Gen_Stack("RecCity=NSU",
"ODS_INFO=test,ODS_INT_INFO=10")
my_result = filter.Gen_Stack("", "")
;
stackId = my_result.NewStackId
updCount = my_result.UpdatedRecordCount
if (updCount > 0)

```

```

        ;
        Protocol("Stack {0} with {1} jobs created", 8, stackId, updCount)
        ;
    end-if
    ;
    Protocol("UpdatedRecordCount: " + my_result.UpdatedRecordCount, 10)
    Protocol("NewStackId: " + my_result.NewStackId, 10)
    ;
else
    ;
    Protocol("No Jobs for processing found", 10)
    ;
end-if

```

## Métodos do objeto Envelope

### ***Geral***

O objeto do envelope permite que você crie conjuntos de dados de envelope.

### ***AppendVJob***

Insera um trabalho **virtual** atrás da posição interna do cursor na tabela interna de trabalho.

### **Sintaxe**

AppendVJob(renditionFileName, pageCount, formName)

### **Parâmetro**

Parâmetro	Significado
renditionFileName	Instrução com o nome completo do caminho para o arquivo XPS a ser usado como o documento para o trabalho virtual.
pageCount	Número de páginas do documento
formName	Nome do formulário usado

### **Valor de retorno**

-

### ***CalcEnvelopePages***

Calcula o número de páginas para o envelope atual (1 folha tem 2 páginas).

Leva em consideração todos os documentos de conjuntos de dados de trabalho e seus documentos anexos respectivos (Obj\_Annex).

Os documentos anexos na definição de postagem não são levados em consideração.

**Sintaxe**

CalcEnvelopePages()

**Parâmetro**

-

**Valor de retorno**

Número de páginas.

***CalcEnvelopePapers***

Calcula o número de folhas necessárias para o envelope atual (1 folha = 2 páginas).

Leva em consideração todos os documentos de conjuntos de dados do trabalho e seus documentos anexos respectivos (Odj\_Annex).

Os documentos anexos na definição de postagem não são levados em consideração.

**Sintaxe**

CalcEnvelopePapers()

**Parâmetro**

-

**Valor de retorno**

Número de folhas.

***CalcEnvelopePostage***

Calcula a classe de postagem do envelope atual. A base necessária para o cálculo da classe de postagem é o peso de todos os documentos nos conjuntos de dados do trabalho e seus documentos anexos (conforme definido em Odj\_Annex).

Os documentos anexos na definição de postagem não são levados em consideração.

**Sintaxe**

CalcEnvelopePostage()

**Parâmetro**

-

**Valor de retorno**

Nome da classe de postagem usada.

***CalcEnvelopeWeight***

Calcula o peso do envelope atual

Leva em consideração todos os documentos de conjuntos de dados de trabalho e seus documentos anexos respectivos (Obj\_Annex).

Os documentos anexos na definição de postagem não são levados em consideração.

**Sintaxe**

CalcEnvelopeWeight()

**Parâmetro**

-

**Valor de retorno**

Peso em gramas

***Confirmar***

Finaliza a transação de dados de base iniciada com o comando StartTransaction.

Todas as modificações de dados executadas desde o início da transação são transformadas em parte permanente do banco de dados.

**Sintaxe**

Commit()

**Parâmetro**

-

**Valor de retorno**

-

***Contagem***

Calcula o número de trabalhos contido no envelope.

**Sintaxe**

Count()

**Parâmetro**

-

**Valor de retorno**

Número de trabalhos.

***CreateEnv***

Cria um novo envelope no banco de dados e atribui os trabalhos selecionados no momento na tabela de trabalho interna do envelope.

Leva em consideração quaisquer marcadores de divisão configurados.

**Sintaxe**

CreateEnv()

**Parâmetro**

-

**Valor de retorno**

A ID do banco de dados do registro do envelope criado.

Se mais de um envelope for criado, a string retornada contém uma lista separada por vírgulas de todas as IDs.

***EndTimer***

Para o temporizador iniciado anteriormente com o comando 'StartTimer' e retorna o tempo decorrido em milissegundos.

**Sintaxe**

EndTimer()

**Parâmetro**

-

**Valor de retorno**

Número de milissegundos decorridos.

***Eof***

Verifica se o cursor está posicionado, no momento, em um conjunto de dados na tabela interna de trabalho.

**Sintaxe**

Eof()

**Parâmetro**

-

**Valor de retorno**

True: Eof alcançado

False: Eof não alcançado

***First***

Posiciona o cursor no primeiro registro na tabela interna de trabalho.

**Sintaxe**

First()

**Parâmetro**

-

**Valor de retorno**

-

***GetField***

Lê o conteúdo de um campo no conjunto de dados atual na tabela interna de trabalho.

**Sintaxe**

GetField(fieldName)

**Parâmetro**

Parâmetro	Significado
fieldName	Nome da coluna

**Valor de retorno**

Conteúdo atual do campo

***GotoJobTillWeight***

Começa da posição atual do cursor na tabela de trabalho interna e move o cursor para frente, contanto que o peso total de cada tarefa individual não exceda o peso definido.

Definir um marcador de divisão nessa posição permite que você divida o envelope em envelopes menores.

**Sintaxe**

GotoJobTillWeight(tillWeight)

**Parâmetro**

Parâmetro	Significado
tillWeight	O peso máximo não deve ser excedido pela soma dos pesos de trabalhos individuais.

**Valor de retorno**

True: O cursor alcançou o fim da tabela de trabalho interna antes do peso máximo ser alcançado.

False: O cursor não alcançou o fim da tabela de trabalho interna antes do peso máximo ser alcançado.



**GotoRecNum**

Move o cursor para um conjunto de dados definido na tabela de trabalho interna.

**Sintaxe**

GotoRecNum(recNumber)

**Parâmetro**

Parâmetro	Significado
recNumber	Índice com base em 0 do conjunto de dados onde o cursor será posicionado.

**Valor de retorno**

-

**Init**

Inicializa o objeto de envelope.

Esse método deve ser o primeiro método chamado, após o objeto de envelope e objeto de filtro terem sido definidos. Objetos de envelope inicializados devem ser liberados no final de um script de filtro com a função **FreeObject**.

**Sintaxe**

Init(groupFields, jobSqlQuery, takeOverFields, takeOverValues)

**Parâmetro**

Parâmetro	Significado
groupFields	Os nomes de coluna da tabela de trabalho (separados por vírgula) definindo a alteração de grupo.
jobSqlQuery	A instrução selecionada para determinar as gravações de trabalho para enchimento de envelopes lógicos.
takeOverFields	Campos adicionais na tabela de envelope ODIN_Envelope a serem preenchidos quando um conjunto de dados de envelope for criado.
takeOverValues	Informações sobre a origem dos valores para as colunas adicionais. É possível usar os nomes de colunas da tabela de trabalho (inserido com ':' antes do nome), bem como os valores constantes.

**Valor de retorno**

-

***InsertVJob***

Insere um trabalho 'virtual' antes da posição do cursor interno na tabela de trabalhos interna.

**Sintaxe**

InsertVJob(renditionFileName, pageCount, formName)

**Parâmetro**

Parâmetro	Significado
renditionFileName	Instrução com o nome completo do caminho para o arquivo XPS a ser usado como o documento para o trabalho virtual.
pageCount	Número de páginas do documento
formName	Nome do formulário usado

**Valor de retorno**

-

***Último***

Posiciona o cursor no último registro da tabela de trabalhos interna.

**Sintaxe**

Last()

**Parâmetro**

-

**Valor de retorno**

-

***Próximo***

Posiciona o cursor no próximo registro da tabela de trabalhos interna.

**Sintaxe**

Next()

**Parâmetro**

-

**Valor de retorno**

-

***Anterior***

Posiciona o cursor no registro anterior da tabela de trabalhos interna.

**Sintaxe**

Prev()

**Parâmetro**

-

**Valor de retorno**

-

***ReadNextEnvelope***

Calcula os conjuntos de dados para o envelope seguinte, de acordo com os campos de alteração de grupo especificados.

Consulte o método [Init](#), parâmetro **groupFields**.

**Sintaxe**

ReadNextEnvelope()

**Parâmetro**

-

**Valor de retorno**

O número de trabalhos nesse envelope ou '0' se o final da tabela de trabalhos for atingido (Eof).

***Recnum***

Lê o índice de registros da posição atual do cursor na tabela de trabalhos internos.

**Sintaxe**

Recnum()

**Parâmetro**

-

**Valor de retorno**

Índice do registro atual. O valor de retorno tem base 0:

O valor de retorno '-1' significa Eof

***Rollback***

Termina a transação de banco de dados que foi iniciada pelo comando [StartTransaction](#).  
 Todas as modificações de dados realizadas desde o início da transação são descartadas.  
 Esse método deve ser chamado no caso de um erro.

**Sintaxe**

```
RollBack()
```

**Parâmetro**

-

**Valor de retorno**

-

***SetField***

Grava um valor no conjunto de dados atual da tabela de trabalhos interna. O novo valor é inserido no banco de dados quando o trabalho é atualizado.

**Sintaxe**

```
SetField(fieldName, value)
```

**Parâmetro**

Parâmetro	Significado
fieldName	Nome da coluna
value	Novo valor

**Valor de retorno**

-

***Split***

Define um marcador de divisão na posição atual do cursos na tabela de trabalhos interna.  
 Permite que você divida os conjuntos de dados incluídos atualmente na tabela de trabalhos em vários envelopes.

**Sintaxe**

```
Split()
```

**Parâmetro**

-

**Valor de retorno**

-

***StartTimer***

Inicia o temporizador interno.

**Sintaxe**

StartTimer()

**Parâmetro**

-

**Valor de retorno**

-

***StartTransaction***

Inicia uma transação de banco de dados.

**Sintaxe**

StartTransaction()

**Parâmetro**

-

**Valor de retorno**

-

**Sample Script: Simple Enveloping Filter with Stack generation**

```

;*****
; Filter Script Demo
;Simple Enveloping Filter with Stack Generation
;*****
;
filterName = "FilterOutsourcing"
printerName = "PrinterOutsourcing"
postageDefinition = "PostageDefinition"
;
#region Postage/Printer
;
;set the Postage Definition
filter.DefinePostageDefinitionByName(postageDefinition)
;
;set the Printer assigned to the Jobs when the Envelope is created

```

```

filter.DefinePrinterByName(printerName,
ModusSuiteObjectType.OutsourcingPrinter)
;
#endregion
;
#region SQL-Definition
;
;Group field change to define Envelope end
GroupFields =
"ODJ_REC_NAME,ODJ_REC_AKZ,ODJ_REC_ZIP,ODJ_REC_CITY,ODJ_REC_STREET"
;
;Select to load the Jobs for enveloping
;datasets are locked using the function "Updlock"
; you must enter a space before and after " FROM "
; the where condition must at least contain the following minimum conditions:
;   ODJ_Stack_Id is NULL: Job is not assigned to a Stack
;   Odj_Env_Id is NULL: Job is not assigned to an Envelope
;   Odj_State = 1: Rendition created
JobSelect = "Select Odin_Job.* From Odin_Job with (Updlock) " +
"where (ODJ_Stack_Id is NULL) and (Odj_Env_Id is null) and (Odj_State = 1) "
+
"order by ODJ_REC_NAME,ODJ_REC_AKZ,ODJ_REC_ZIP,ODJ_REC_CITY,ODJ_REC_STREET"
;
;As an alternative, SQL statements can be read from a LookupTable
;Example:
;Sql-Statement should be read from the entry "JobSelect" in the LookupTable
"odin"
;JobSelect = GetLookupValue("odin", "JobSelect")
;
;Content of columns from the Jobs to be inserted into the Envelope created
TakeOverFields =
"ODE_REC_NAME,ODE_REC_AKZ,ODE_REC_ZIP,ODE_REC_CITY,ODE_REC_STREET,ODE_DEVICE"
TakeOverValues =
":ODJ_REC_NAME,:ODJ_REC_AKZ,:ODJ_REC_ZIP,:ODJ_REC_CITY,:ODJ_REC_STREET,'Devic
e x'"
;
#endregion
;
;Create Envelope instance outside the try-block
EnvEnum = filter.EnumEnvelopes()
;
#region Envelope Stuffing
;
try
    ;initialize inside the block
    EnvEnum.Init(GroupFields, JobSelect, TakeOverFields, TakeOverValues)
    EnvEnum.StartTransaction()

```

```

;
JobsInEnvelope = EnvEnum.ReadNextEnvelope()
while (JobsInEnvelope > 0)
    ;
    EnvelopeId = EnvEnum.CreateEnv()
    JobsInEnvelope = EnvEnum.ReadNextEnvelope()
    ;
end-while
;
EnvEnum.Commit()
;
OnError
;
EnvEnum.Rollback()
Protocol("Exception: {0}", 10, Error.Message)
Protocol("StackTrace: {0}", 10, Error.StackTrace)
Raise(Error.Token, Error.Message)
;
finally
;
;free object to
;free up internal resources
FreeObject("EnvEnum")
;
end-try
;
#endregion
;
#region create Stack for Envelope with no Stack assignment
;
;read the DbAlias of the Filter component
DatabaseAliasName = filter.GetParam("DBALIAS_NAME")
;
;Get number of open Envelopes
InitDataset("OdinEnv", DatabaseAliasName)
StringListAdd("EnvCountSql", "Select count(*) EnvCount from Odin_envelope
Where (ODE_Stack_Id is null)")
SetSQL("OdinEnv", "EnvCountSql")
OpenSQL("OdinEnv")
OpenEnvCount = GetField("OdinEnv", "EnvCount")
;
;create Stack if open Envelopes found
;A Stack with Envelopes is created for each Postage Class of the selected
Postage Definition if open Envelopes are found where the column ODE_POSTAGE
;
;contains the name of the Postage Class

```

```

if (OpenEnvCount > 0)
  ;
  filter.SetParam("filtername", filterName)
  filter.SetParam("holdtime", "10")
  filter.SetParam("priority", "200")
  filter.SetParam("STACKTYPE", "20")
  ;filter.SetParam("OMR_TYP", "C4_OMR")
  filter.SetParam("OMR_TYP", "")
  ;
  ;"StackId" is a protected keyword and is replaced by the ID of the Stack
  created during processing.
  ;"Postage" is a user-defined keyword. The value is defined in the
  UpdateParameter of the method Gen_Stack.
  UpdateSql =
  "Update Odin_Envelope "+
  "Set ODE_Stack_Id = :StackId "+
  "Where (ODE_Stack_Id is null) and (ODE_POSTAGE = :Postage)"
  ;
  filter.SetParam("UPDATE_SQL", UpdateSql)
  ;
  ;returns the names of the Postage Classes defined in the loaded Postage
  Definition (comma separated)
  PostageNames = filter.GetParam("PostageDef_Str")
  ;
  while (PostageNames <> "")
    Postage = StrTok("PostageNames", ",")
    if (Postage = "")
      Postage = PostageNames
      PostageNames = ""
    end-if
    ;
    Filter.SetParam("FilterName", Postage)
    UpdateParameter = FormatStr("Postage=%s", Postage)
    StackResult = Filter.Gen_Stack(UpdateParameter, "")
    if (StackResult.NewStackId = "0")
      Protocol("No Stacks for Postage Class '{0}' created", 10, Postage)
    else
      Protocol("Stack for Postage Class '{0}' has the ID '{1}' and
      contains '{2}' Envelopes", 10, Postage, StackResult.NewStackId,
      StackResult.UpdatedRecordCount)
    end-if
    ;
  end-while
;
else
  Protocol("No Envelopes for processing found", 10)

```



```
end-if
;
#endregion
```

## Contexto OdinPrint

### Propriedades do contexto OdinPrint

Parâmetro	Significado				
CurrentEnvelope	<p>Envelope atual, se for encontrado. (ReadOnly)</p> <p><b>Métodos</b></p> <p>GetValue</p> <p>Retorna o valor da coluna dada.</p> <p><b>Sintaxe</b></p> <p>GetValue(string columnName)</p> <table border="1"> <thead> <tr> <th>Parâmetro</th> <th>Significado</th> </tr> </thead> <tbody> <tr> <td>columnName</td> <td>Nome da coluna</td> </tr> </tbody> </table> <p><b>Valor de retorno</b></p> <p>Valor da coluna</p> <p>Se o tipo de dados da coluna do banco de dados for DATE, Datetime ou qualquer tipo de dados válido, a data lida geralmente será retornada no formato MM/DD/AAAA hh:mm:ss.</p> <p><b>Exemplo</b></p> <pre>GetObject("DateVar", "MLDate") ... StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT") ;Converter a data no formato desejado com a ajuda do objeto MLDate IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy hh:mm:ss", StringDateVar ) v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy hh:mm:ss ", IntDateVar)</pre> <p>GetFieldNameEnumerator</p> <p>Retorna um enumerador.</p> <p><b>Sintaxe</b></p> <p>GetEnvelopeEnumerator()</p> <p><b>Parâmetro</b></p>	Parâmetro	Significado	columnName	Nome da coluna
Parâmetro	Significado				
columnName	Nome da coluna				

	<p>-</p> <p><b>Valor de retorno</b></p> <p>Enumerador com estes métodos:</p> <table border="1" data-bbox="431 430 1388 693"> <thead> <tr> <th>Método</th> <th>Função</th> </tr> </thead> <tbody> <tr> <td>Current</td> <td>Retorna o objeto atual.</td> </tr> <tr> <td>MoveNext</td> <td>Movimenta a posição até o próximo objeto a ser processado.</td> </tr> <tr> <td>Redefinir</td> <td>Redefine o enumerador de volta ao início do primeiro elemento.</td> </tr> </tbody> </table>	Método	Função	Current	Retorna o objeto atual.	MoveNext	Movimenta a posição até o próximo objeto a ser processado.	Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.
Método	Função								
Current	Retorna o objeto atual.								
MoveNext	Movimenta a posição até o próximo objeto a ser processado.								
Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.								
CurrentJob	<p>Trabalho atual, se for encontrado. (ReadOnly)</p> <p><b>Métodos</b></p> <p>GetValue</p> <p>Retorna o valor da coluna dada.</p> <p><b>Sintaxe</b></p> <p>GetValue(string columnName)</p> <table border="1" data-bbox="431 1003 865 1136"> <thead> <tr> <th>Parâmetro</th> <th>Significado</th> </tr> </thead> <tbody> <tr> <td>columnName</td> <td>Nome da coluna</td> </tr> </tbody> </table> <p><b>Valor de retorno</b></p> <p>Valor da coluna</p> <p>Se o tipo de dados da coluna do banco de dados for DATE, Datetime ou qualquer tipo de dados válido, a data lida geralmente será retornada no formato MM/DD/AAAA hh:mm:ss.</p> <p><b>Exemplo</b></p> <pre>GetObject("DateVar", "MLDate") ... StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT") ;Converter a data no formato desejado com a ajuda do objeto MLDate IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy hh:mm:ss", StringDateVar ) v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy hh:mm:ss ", IntDateVar)</pre> <p>GetFieldNameEnumerator</p> <p>Retorna um enumerador.</p> <p><b>Sintaxe</b></p> <p>GetEnvelopeEnumerator()</p>	Parâmetro	Significado	columnName	Nome da coluna				
Parâmetro	Significado								
columnName	Nome da coluna								

	<p><b>Parâmetro</b></p> <p>-</p> <p><b>Valor de retorno</b></p> <p>Enumerador com estes métodos:</p> <table border="1" data-bbox="431 478 1386 743"> <thead> <tr> <th>Método</th> <th>Função</th> </tr> </thead> <tbody> <tr> <td>Current</td> <td>Retorna o objeto atual.</td> </tr> <tr> <td>MoveNext</td> <td>Movimenta a posição até o próximo objeto a ser processado.</td> </tr> <tr> <td>Redefinir</td> <td>Redefine o enumerador de volta ao início do primeiro elemento.</td> </tr> </tbody> </table>	Método	Função	Current	Retorna o objeto atual.	MoveNext	Movimenta a posição até o próximo objeto a ser processado.	Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.
Método	Função								
Current	Retorna o objeto atual.								
MoveNext	Movimenta a posição até o próximo objeto a ser processado.								
Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.								
CurrentStack	<p>Pilha atual (ReadOnly)</p> <p><b>Métodos</b></p> <p>GetValue</p> <p>Retorna o valor da coluna dada.</p> <p><b>Sintaxe</b></p> <p>GetValue(string columnName)</p> <table border="1" data-bbox="431 1052 865 1184"> <thead> <tr> <th>Parâmetro</th> <th>Significado</th> </tr> </thead> <tbody> <tr> <td>columnName</td> <td>Nome da coluna</td> </tr> </tbody> </table> <p><b>Valor de retorno</b></p> <p>Valor da coluna</p> <p>Se o tipo de dados da coluna do banco de dados for DATE, Datetime ou qualquer tipo de dados válido, a data lida geralmente será retornada no formato MM/DD/AAAA hh:mm:ss.</p> <p><b>Exemplo</b></p> <pre>GetObject("DateVar", "MLDate") ... StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT") ;Converter a data no formato desejado com a ajuda do objeto MLDate IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy hh:mm:ss", StringDateVar ) v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy hh:mm:ss ", IntDateVar)</pre> <p>GetFieldNameEnumerator</p> <p>Retorna um enumerador.</p> <p><b>Sintaxe</b></p>	Parâmetro	Significado	columnName	Nome da coluna				
Parâmetro	Significado								
columnName	Nome da coluna								

GetEnvelopeEnumerator()	
<b>Parâmetro</b>	
-	
<b>Valor de retorno</b>	
Enumerador com estes métodos:	
Método	Função
Current	Retorna o objeto atual.
MoveNext	Move a posição até o próximo objeto a ser processado.
Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.

## Métodos do contexto OdinPrint

### ***HasData***

Verifica se os dados existem para o tipo determinado.

### **Sintaxe**

HasData(OdinDataRowType dataType)

### **Parâmetro**

Parâmetro	Significado	Valores possíveis
DataType	Tipo de dados	OdinDataRowType.Process OdinDataRowType.Stack OdinDataRowType.Envelope OdinDataRowType.Job OdinDataRowType.VirtualJob

### **Valor de retorno**

True: A linha de dados especificada existe.

False: A linha de dados especificada não existe.

## Contexto OdinRendition

### **Geral**

Contexto para conversão de arquivos de documentos dos trabalhos para o formato de arquivo de spool do Odin (formato XPS).

A variável **CurrentJobRecord** fornece as propriedades e métodos a seguir.

#### Propriedades do contexto OdinRendition

Propriedade	Significado
CurrentJobRecord	Lê/grava o acesso ao conjunto de dados do trabalho atual
CurrentProcessId	Especifica a ID do processo Odin sendo processado no momento

#### Métodos do contexto OdinRendition

##### ***GetFieldNameEnumerator***

Retorna um enumerador.

##### **Sintaxe**

GetEnvelopeEnumerator()

##### **Parâmetro**

-

##### **Valor de retorno**

Enumerador com estes métodos:

Método	Função
Current	Retorna o objeto atual.
MoveNext	Move a posição para o próximo objeto a ser processado.
Reset	Redefine o enumerador de volta para o início do primeiro elemento.

##### ***GetValue***

Retorna o valor da coluna fornecida.

##### **Sintaxe**

GetValue(string columnName)

Parâmetro	Significado
columnName	Nome da coluna

##### **Valor de retorno**

Valor da coluna

Se o tipo de dados da coluna do banco de dados DATE, Datetime ou qualquer tipo de dados válido, a data lida normalmente será retornada no formato MM/DD/AAAA hh:mm:ss.

### Exemplo

```
GetObject("DateVar", "MLDate")
...
StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT")
;Convert date to the required format with the help of the MLDate object
IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy hh:mm:ss", StringDateVar )
v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy hh:mm:ss ", IntDateVar)
```

### HasData

Verifica se os dados existem para o tipo determinado.

### Sintaxe

```
HasData(OdinDataRowType dataType)
```

### Parâmetro

Parâmetro	Significado	Valores possíveis
DataType	Tipo de dados	OdinDataRowType.Process OdinDataRowType.Stack OdinDataRowType.Envelope OdinDataRowType.Job OdinDataRowType.VirtualJob

### Valor de retorno

True: A linha de dados especificada existe.

False: A linha de dados especificada não existe.

### SetValue

Grava um valor em uma coluna de banco de dados do conjunto de dados de trabalho atual.

### Sintaxe

```
SetValue(string columnName, string newValue)
```

### Parâmetro

Parâmetro	Significado
columnName	Nome da coluna

newValue

Valor da coluna

**Valor de retorno**

-

**Exemplo**

```
Printer = CurrentJobRecord.GetValue("ODJ_PRINTER_OID")
if (Printer = "")
    CurrentJobRecord.SetValue("ODJ_PRINTER_OID", "test-pr-5")
end-if
```

## Contexto OdinStream

### Geral

O contexto OdinStream fornece scripts com um objeto de fluxo que pode ser acessado com o **Stream**.

### Propriedades do contexto OdinStream

▶ [Mostrar tudo](#)

Propriedade	Significado
Stream	<p>Retorna a interface IOdinStreamDocument.</p> <p>Alias de script: stream</p> <p>Métodos de IOdinStream-Interface</p> <p><a href="#">AppendDocument</a></p> <p><a href="#">GetProperty</a></p> <p><a href="#">LoadDocument</a></p> <p><a href="#">LoadEnvelopes</a></p> <p><a href="#">LoadJobs</a></p> <p><a href="#">SetProperty</a></p> <p><a href="#">Script de exemplo LoadEnvelopes / LoadJobs</a></p>
HasStream	Declara se um objeto de fluxo existe na propriedade do contexto <b>Stream</b> .

### Métodos da interface IOdinStream

#### ***AppendDocument***

Adiciona um documento carregado com LoadDocument a um fluxo.

#### **Sintaxe**

AppendDocument(IOdinStreamDocument document, bool insertOMR, bool insertDVFrei)

### Parâmetro

Parâmetro	Significado
document	Documento a ser adicionado
insertOMR	Define se os códigos de OMR serão impressos no documento. Os códigos de OMR serão impressos em uma página de capa da pilha.
insertDVFrei	Parâmetro reservado para implementação posterior. Nenhuma função atual.

### Valor de retorno

-

### Exemplo

```
document = stream.LoadDocument("d:\import\Empty_Page.xps",
OdinDuplexSetting.Simplex, "firstbin", "nextbin")
stream.AppendDocument(document, true, false)
```

### GetProperty

O método GetProperty pode recuperar valores de fluxo, por exemplo, contagem de páginas.

### Sintaxe

GetProperty(string propertyName)

### Parâmetro

Parâmetro	Significado
PropertyName	Nome da propriedade

### Valor de retorno

Valor da propriedade

As seguintes propriedades podem ser consultadas. Os identificadores nos colchetes são os definidos no modus 4 que continuam válidos para assegurar a compatibilidade com versões anteriores. Eles são convertidos automaticamente para os novos nomes correspondentes.

### Geral:

Propriedade	Significado
EnvelopeId	ID de envelope do envelope em fluxo no momento (Longa)



JobId	ID de trabalho do trabalho/documento em fluxo no momento (Longa)
OMRCode	Valor atual do código OMR (UInt) READONLY
OMRReset	<p>True/False</p> <p>Se a propriedade estiver definida como TRUE, um sinalizador de redefinição (todas as marcas de OMR definidas) é impresso em todos os documentos subsequentes ao invés do código OMR.</p> <p>O sinalizador de redefinição é impresso até que a propriedade seja definida como FALSE.</p> <p>O evento OnStackEnd permite que um documento ou página adicional seja inserido após a pilha, que contém um sinalizador de redefinição para o alimentador de produção.</p>
StackId	ID de pilha da pilha a ser colocada em fluxo (Longa)
StreamFileName	Nome do arquivo de fluxo
Sysdate	Data atual
Systime	Hora atual
VirtualPageDescription (VPageDesc)	Para uma página de capa da pilha = ' <b>StackCoverPage</b> ' para uma página de capa do envelope = ' <b>EnvelopeCoverPage</b> ', caso contrário, uma string vazia (String)

**Pilha:**

Propriedade	Significado
StackEnvelope	Contador de envelope para toda a pilha (inteiro)
StackJob	Contador de trabalho / documento para toda a pilha (inteiro)
StackPage	Contador de página para toda a pilha (inteiro)
StackPaper	Contador de folha para toda a pilha (inteiro)
TotalStackEnvelopes (EnvCount)	Número de envelopes em uma pilha (inteiro)
TotalStackJobs	Número de trabalhos/documentos em uma pilha (inteiro)
TotalStackPages (StackPage)	Número de páginas em uma pilha (inteiro)
TotalStackPapers	Número de folhas em uma pilha (inteiro)

**Envelope:**

Propriedade	Significado
EnvelopeEnd	Especifica se o documento atual é o último documento de um envelope (Booleano)
EnvelopeJob	Contador de trabalho / documento para todo o envelope (inteiro)
EnvelopePage	Contador de página para todo o envelope (inteiro)
EnvelopePaper	Contador de folha para todo o envelope (inteiro)
EnvelopeStart (EnvStart)	Especifica se o documento atual é o primeiro documento de um envelope (Booleano)
TotalEnvelopeJobs (JobCount)	Número de trabalhos/documentos em um envelope (inteiro)
TotalEnvelopePages	Número de páginas em um envelope (inteiro)
TotalEnvelopePapers	Número de folhas em um envelope (inteiro)

**Trabalho/documento**

Propriedade	Significado
JobPage (PageCount)	Contador de páginas para um documento (inteiro)
JobPaper (PageCountExtended)	Contador de folhas para um documento (inteiro)
TotalJobPages	Número de páginas em um documento (inteiro)
TotalJobPapers	Número de folhas em um documento (inteiro)

**Exemplo**

```

; Stack was processed by streaming...
; As an example, an additional page is inserted containing details of the
number of pages/sheets, envelopes/jobs
;
TotalPages = stream.GetProperty("TotalStackPages")
TotalPapers = stream.GetProperty("TotalStackPapers")

```

```

TotalEnvelopes = stream.GetProperty("TotalStackEnvelopes")
TotalJobs = stream.GetProperty("TotalStackJobs")
;
; Load cover page for tray "coverpage"
document =
stream.LoadDocument("c:\PerceptiveSoftware\modusSuite\data\EmptyPage.xps",
OdinDuplexSetting.Simplex, "coverpage", "coverpage")
;
; print text as a test
document.SetFont("Tahoma", 16)
document.EditPage(0)
document.TextOut(20.5, 10.5, FormatStr("TotalPages/TotalPapers: %d/%d",
TotalPages, TotalPapers), false, false)
document.TextOut(20.5, 17, "Number Envelopes/Jobs: %d/%d", TotalEnvelopes,
TotalJobs), false, false)
document.Post()
;
; Print OMR-Reset character
Stream.SetProperty("OMRReset", true)
;
; Add document to stream
stream.AppendDocument(document, true, false)
;
; Free up the loaded document
FreeObject("document")

```

### ***LoadDocument***

Carrega um documento XPS.

Este documento pode ser adicionado como uma pilha adicional, ou página de capa de envelope, ou um documento virtual usando o método AppendDocument.

#### **Sintaxe**

LoadDocument(string fileName, OdinDuplexSetting duplex, string firstBin, string nextBin)

#### **Parâmetro**

Parâmetro	Significado
fileName	Nome do documento a ser carregado
duplex	<b>Valores possíveis:</b> OdinDuplexSetting.Simplex: Modo simplex - sem duplex. OdinDuplexSetting.Horizontal: Duplex horizontal (borda maior) OdinDuplexSetting.Vertical: Duplex vertical (borda menor)

firstBin	Bandeja para a primeira página
nextBin	Bandeja para a página seguinte

**Valor de retorno**

Retorna a interface **IOdinStreamDocument** com os métodos exigidos para desempenhar operações de desenhos nas páginas.

**Exemplo**

```
document = stream.LoadDocument("d:\import\Empty_Page.xps",
OdinDuplexSetting.Simplex, "firstbin", "nextbin")
```

***LoadEnvelopes***

Carrega todos os conjuntos de dados do envelope de uma pilha na tabela virtual 'ENVELOPE' para processamento adicional.

**Sintaxe**

```
LoadEnvelopes()
```

**Parâmetro**

-

**Valor de retorno**

Número de conjuntos de dados do envelope (inteiro).

***LoadJobs***

Este método determina a ID de envelope do conjunto de dados do envelope atual, carrega todos os trabalhos do envelope correspondente e os disponibiliza na tabela virtual **Job**.

**Sintaxe**

```
LoadJobs()
```

**Parâmetro**

-

**Valor de retorno**

Número de conjuntos de dados do trabalho (inteiro).

***SetProperty***

Use SetProperty para especificar novas configurações para propriedades.

**Observe:**

A alteração dos valores afeta contadores, o que significa que a função deve ser usada com muito cuidado!

### Sintaxe

```
 SetProperty(string propertyName, object Value)
```

### Parâmetro

Parâmetro	Significado
Name	Nome da propriedade (string)
propertyValue	Valor da propriedade (string, inteiro, booleano)

### Valor de retorno

-

### *Script de exemplo: LoadEnvelopes / LoadJobs*

Os métodos do VirtualDatabase, GetField(), por exemplo, fornecem acesso a dados de pilha, envelope e trabalho.

#### Exemplo 1:

```
ods_id = GetField("STACK", "ODS_ID")
ode_id = GetField("ENVELOPE", "ODE_ID")
odj_id = GetField("JOB", "ODJ_ID")
```

#### Exemplo 2:

```
; load stack envelopes
envelopeCount = stream.LoadEnvelopes()
First("ENVELOPE")
while (Eof("ENVELOPE") = false)
;
    ; Envelope ID
    EnvelopeId = GetField("ENVELOPE", "ODE_ID")
    ; Number of jobs in the envelope
    EnvelopeJobCount = GetField("ENVELOPE", "ODE_JOB_COUNT")
    ; load all jobs to the envelope
    JobsInEnvelope = stream.LoadJobs();
;
    ; Check the number of jobs defined in the envelope matches the number of
    actual jobs
    if (EnvelopeJobCount <> JobsInEnvelope)
        Protocol("Number of jobs is different (ODE_JOB_COUNT={0} / Job-
        Anzahl={1}", 10, EnvelopeJobCount, JobsInEnvelope)
    end-if
```

```

;
; get all job IDs
while (Eof("JOB") = false)
    JobId = GetField("Job", "ODJ_ID")
    ;
    Next("Job")
end-while
;
Next("ENVELOPE")
end-while

```

## Métodos do contexto OdinStream

### **HasData**

Verifica se os dados existem para o tipo determinado.

### **Sintaxe**

HasData(OdinDataRowType dataType)

### **Parâmetro**

Parâmetro	Significado	Valores possíveis
DataType	Tipo de dados	OdinDataRowType.Process OdinDataRowType.Stack OdinDataRowType.Envelope OdinDataRowType.Job OdinDataRowType.VirtualJob

### **Valor de retorno**

True: A linha de dados especificada existe.

False: A linha de dados especificada não existe.

## Contexto OdinStreamDocument

### Geral

[▶ Mostrar tudo](#)

O contexto OdinStream fornece scripts com um objeto de fluxo que pode ser acessado com o **streamDoc**.

Todas as informações de posição, além da largura e altura, são especificadas em milímetros.

### **As cores são definidas como uma string:**

#AARRGGBB AA=Transparent, RR=Red, GG=Green, BB=Blue

por exemplo, para Black #00000000, para White#00FFFFFF

Nomes também podem ser especificados para cores predefinidas

AliceBlue,AntiqueWhite,Aqua,Aquamarine,Azure,Beige,Bisque,Black,BlanchedAlmond,Blue,BlueViolet,Brown,BurlyWood,CadetBlue,Chartreuse,Chocolate,Coral,CornflowerBlue,Cornsilk,Crimson,Cyan,DarkBlue,DarkCyan,DarkGoldenrod,DarkGray,DarkGreen,DarkKhaki,DarkMagenta,DarkOliveGreen,DarkOrange,DarkOrchid,DarkRed,DarkSalmon,DarkSeaGreen,DarkSlateBlue,DarkSlateGray,DarkTurquoise,DarkViolet,DeepPink,DeepSkyBlue,DimGray,DodgerBlue,Firebrick,FloralWhite,ForestGreen,Fuchsia,Gainsboro,GhostWhite,Gold,Goldenrod,Gray,Green,GreenYellow,Honeydew,HotPink,IndianRed,Indigo,Ivory,Khaki,Lavender,LavenderBlush,LawnGreen,LemonChiffon,LightBlue,LightCoral,LightCyan,LightGoldenrodYellow,LightGray,LightGreen,LightPink,LightSalmon,LightSeaGreen,LightSkyBlue,LightSlateGray,LightSteelBlue,LightYellow,Lime,LimeGreen,Linen,Magenta,Maroon,MediumAquamarine,MediumBlue,MediumOrchid,MediumPurple,MediumSeaGreen,MediumSlateBlue,MediumSpringGreen,MediumTurquoise,MediumVioletRed,MidnightBlue,MintCream,MistyRose,Moccasin,NavajoWhite,Navy,OldLace,Olive,OliveDrab,Orange,OrangeRed,Orchid,PaleGoldenrod,PaleGreen,PaleTurquoise,PaleVioletRed,PapayaWhip,PeachPuff,Peru,Pink,Plum,PowderBlue,Purple,Red,RosyBrown,RoyalBlue,SaddleBrown,Salmon,SandyBrown,SeaGreen,SeaShell,Sienna,Silver,SkyBlue,SlateBlue,SlateGray,Snow,SpringGreen,SteelBlue,Tan,Teal,Thistle,Tomato,Transparent,Turquoise,Violet,Wheat,White,WhiteSmoke,Yellow,YellowGreen

## Propriedades de IOdinStreamDocument-Interface

► [Mostrar tudo](#)

Propriedade	Significado
Stream	<p>Retorna a interface IOdinStreamDocument.</p> <p>Alias de script: stream</p> <p>Métodos de IOdinStream-Interface</p> <p><a href="#">AppendDocument</a></p> <p><a href="#">GetProperty</a></p> <p><a href="#">LoadDocument</a></p> <p><a href="#">LoadEnvelopes</a></p> <p><a href="#">LoadJobs</a></p> <p><a href="#">SetProperty</a></p> <p><a href="#">Script de exemplo LoadEnvelopes / LoadJobs</a></p> <p>Consulte também Contexto <a href="#">OdinStream</a></p>
StreamDoc	<p>Retorna a interface IOdinStreamDocument do trabalho/documento carregado no momento.</p> <p>Esse documento é inserido automaticamente depois que o script OnDocument é executado, a menos que ele seja adicionado por meio de AppendDocument().</p> <p>Portanto, é possível inserir um documento virtual antes ou depois do documento atual por meio de LoadDocument/AppendDocument().</p> <p>Alias de script: streamdoc</p>

## Métodos de IOdinStreamDocument-Interface

[AddBookmark](#)[AddComment](#)[AddPJLCommand](#)[EditPage](#)[GetInputBin](#)[Image](#)[IsDuplex](#)[IsPortrait](#)[Line](#)[PageCount](#)[Post](#)[Rectangle](#)[SetAfpCopyGroup](#)[SetFont](#)[SetInputBin](#)[TextOut](#)

## Métodos da interface IOdinStreamDocument

**AddBookmark**

Adiciona um bookmark a um documento. Se o formato de saída for PDF, um bookmark é criado para cada bookmark definido.

As IDs deve ser exclusivas para cada bookmark.

**Sintaxe**

AddBookmark(string id, string parentId, string title)

AddBookmark(string id, string parentId, string title, double x, double y)

**Parâmetro**

Parâmetro	Significado
id	ID do Bookmark
parentId	Uma string vazia aqui significa que o bookmark será inserido no nível mais elevado. Caso contrário, o bookmark é inserido como um filho do bookmark especificado.



title	Texto para o bookmark
x	Posição X na página
y	Posição Y na página

**Valor de retorno**

-

**Exemplo**

```
document.AddBookmark("doc1_page1", "doc1", "Page 1")
document.AddBookmark("doc1_page2", "doc1", "Page 2", 0, 0)
```

***AddComment***

Este método permite que você escreva um comentário no arquivo de saída (um registro NOP em AFP).

**Sintaxe**

```
AddComment(string text)
```

**Parâmetro**

Parâmetro	Significado
texto	Texto do comentário

**Valor de retorno**

-

**Exemplo**

```
; Adicionar um registro NOP à primeira página...
streamdoc.EditPage(0)
streamdoc.AddComment("meu registro NOP interno para testes")
streamdoc.Post()
```

***AddPJLCommand***

Esse método permite que você grave um comando PDL em um arquivo de saída Postscript/PCL.

**Sintaxe**

```
AddPJLCommand(string value)
```

**Parâmetro**

Parâmetro	Significado
value	Comando PJJ

### Valor de retorno

-

### Exemplo

```
; Add PJJ-Command for Staple/Punch to first page...
streamdoc.EditPage(0)
streamdoc.AddPJJCommand("@PJJ SET STAPLE=OFF")
streamdoc.AddPJJCommand("@PJJ SET PUNCH=OFF")
streamdoc.Post()
```

### *EditPage*

EditPage define a página determinada para o modo de edição. Os comandos de caractere agora podem ser executados na página.

### Sintaxe

EditPage(int pageNumber)

### Parâmetro

Parâmetro	Significado
pageNumber	Número da página a ser processada, começando com 0.

### Valor de retorno

-

### *GetInputBin*

Retorna o recipiente de entrada de uma página.

Se não estiver definido, retorna a bandeja da primeira página ou a bandeja subsequente, dependendo do número da página.

### Sintaxe

GetInputBin(int pageNumber)

### Parâmetro

Parâmetro	Significado
-----------	-------------

pageNumber	Número da página
------------	------------------

**Valor de retorno**

recipiente de entrada

***Imagem***

Inserir um arquivo de imagem.

**Sintaxe**

Image(double x, double y, string fileName)

Image(double x, double y, string imageFileName, OdinPageLayer layer)

**Parâmetro**

Parâmetro	Significado
x	Posição X
y	Posição Y
filename	Nome do arquivo
layer	<b>Valores possíveis:</b> OdinPageLayer.Foreground: O desenho é feito por cima do documento original. OdinPageLayer.Background: O desenho é feito por baixo do documento original.

**Valor de retorno**

-

**Exemplo*****IsDuplex***

Retorna ao modo duplex.

**Sintaxe**

IsDuplex()

**Parâmetro****Valor de retorno**

True: Duplex

False: Simplex

### ***IsPortrait***

Retorna ao modo retrato.

#### **Sintaxe**

IsPortrait(int pageNumber)

#### **Parâmetro**

Parâmetro	Significado
pageNumber	Número da página

#### **Valor de retorno**

True: Retrato

False: Paisagem

### ***Line***

Desenha uma linha na página atual.

#### **Sintaxe**

Line(double x1, double y1, double x2, double y2)

Line(double x1, double y1, double x2, double y2, double lineWidth, OdinLineStyle lineStyle, string color)

Line(double x1, double y1, double x2, double y2, double lineWidth, OdinLineStyle lineStyle, string color, OdinPageLayer layer)

#### **Parâmetro**

Parâmetro	Significado
x1	Posição X da posição inicial da linha.
y1	Posição Y da posição inicial da linha.
x2	Posição X da posição final da linha.
y2	Posição Y da posição final da linha.
lineWidth	Largura de linha
lineStyle	Valores possíveis

	OdinLineStyle.Solid OdinLineStyle.Dashed OdinLineStyle.Dotted
color	Cor
Layer	<b>Valores possíveis</b> OdinPageLayer.Foreground OdinPageLayer.Background

**Valor de retorno**

-

***PageCount***

Retorna o número de páginas do documento atual.

**Sintaxe**

PageCount()

**Parâmetro**

-

**Valor de retorno**

Número de páginas.

***Publicar***

Grava comandos de caractere na camada do documento.

**Sintaxe**

Post()

**Parâmetro**

-

**Valor de retorno**

-

***Rectangle***

Desenha um retângulo na página atual.

**Sintaxe**

Rectangle(double x, double y, double width, double height, bool fill, string fillColor)

Rectangle(double x, double y, double width, double height, double lineWidth, OdinLineStyle lineStyle, string lineColor, bool fill, string fillColor)

Rectangle(double x, double y, double width, double height, double lineWidth, OdinLineStyle lineStyle, string lineColor, bool fill, string fillColor, OdinPageLayer layer)

### Parâmetro

Parâmetro	Significado
x	Posição X do canto superior esquerdo do retângulo.
y	Posição Y do canto superior esquerdo do retângulo.
width	Largura do retângulo
height	Altura do retângulo
fill	True: o retângulo será preenchido (valor padrão = false)
fillColor	cor de preenchimento
lineWidth	Largura de linha
lineStyle	Valores possíveis OdinLineStyle.Solid OdinLineStyle.Dashed OdinLineStyle.Dotted
lineColor	Cor de linha
Layer	Valores possíveis OdinPageLayer.Foreground: O desenho é feito por cima do documento original. OdinPageLayer.Background: O desenho é feito por baixo do documento original.

### Valor de retorno

-

### ***SetAfpCopyGroup***

Define o CopyGroup de uma página.

### Sintaxe

SetAfpCopyGroup(string copyGroup)

**Parâmetro**

Parâmetro	Significado
copyGroup	CopyGroup a ser definido

**Valor de retorno**

-

**Exemplo**

```

; set the CopyGroup "CG1" on first page of each document
streamdoc.EditPage(0)
streamdoc.SetAfpCopyGroup("CG1")
streamdoc.Post()

```

***SetFont***

Define a especificação de fonte para a saída de texto.

**Sintaxe**

```
SetFont(string fontName, int fontSize)
```

**Parâmetro**

Parâmetro	Significado
fontName	Nome da fonte, por exemplo, Arial.
fontSize	Tamanho da fonte, por exemplo, 10.

**Valor de retorno**

-

***SetInputBin***

Define a bandeja de entrada de uma página.

O método substitui a bandeja predefinida para a primeira página ou para as páginas seguintes, conforme o caso.

**Sintaxe**

```
SetInputBin(int pageNumber, string binName)
```

**Parâmetro**

Parâmetro	Significado
-----------	-------------

pageNumber	Número da página
binName	Nome da bandeja

**Valor de retorno**

-

**Exemplo**

```
;Set the input tray for the last page of the Document
streamDoc.SetInputBin(streamdoc.PageCount - 1, "testBin")
```

***TextOut***

Escreve um texto na página atual.

**Sintaxe**

TextOut(double x, double y, string text, bool bold, bool italic)

TextOut(double x, double y, string text, bool bold, bool italic, string color, int angle)

TextOut(double x, double y, string text, bool bold, bool italic, string color, int angle, OdinPageLayer layer)

**Parâmetro**

Parâmetro	Significado
x	Posição X do início
y	Posição Y do início
text	Texto a ser escrito
bold	True: Formatação em "negrito" (valor padrão = false)
italic	True: Formatação em "itálico" (valor padrão = false)
color	Cor
angle	Ângulo de rotação
layer	<b>Valores possíveis</b> OdinPageLayer.Foreground: O desenho é feito por cima do documento original. OdinPageLayer.Background: O desenho é feito por baixo do documento original.

**Valor de retorno**



## Contexto OdinStreamPrint

### Propriedades do contexto Odin StreamPrint

[▶ Mostrar tudo](#)

Parâmetro	Significado				
CurrentDocument	<p>Documento atual quando encontrado (somente leitura).</p> <p><b>Métodos</b></p> <p>GetDocumentFileName</p> <p>GetDuplex</p> <p>GetFileName</p> <p>GetFirstInputBin</p> <p>GetId</p> <p>GetNextInputBin</p> <p>GetPageCount</p> <p>IsPortrait</p> <p>SetDuplex</p> <p>SetFirstInputBin</p> <p>SetNextInputBin</p>				
HasDocument	Documento carregado True/False (somente leitura)				
CurrentEnvelope	<p>Envelope atual, se encontrado. (Somente leitura)</p> <p><b>Métodos</b></p> <p>GetValue</p> <p>Retorna o valor da coluna dada.</p> <p><b>Sintaxe</b></p> <p>GetValue(string columnName)</p> <table border="1"> <thead> <tr> <th>Parâmetro</th> <th>Significado</th> </tr> </thead> <tbody> <tr> <td>columnName</td> <td>Nome da coluna</td> </tr> </tbody> </table> <p><b>Valor de retorno</b></p> <p>Valor da coluna</p> <p>Se o tipo de dados da coluna do banco de dados for DATE, Datetime ou</p>	Parâmetro	Significado	columnName	Nome da coluna
Parâmetro	Significado				
columnName	Nome da coluna				

qualquer tipo de dados válido, a data lida geralmente será retornada no formato MM/DD/AAAA hh:mm:ss.

### Exemplo

```
GetObject("DateVar", "MLDate")
...
StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT")
;Converter a data no formato desejado com a ajuda do objeto
MLDate
IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy
hh:mm:ss", StringDateVar)
v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy
hh:mm:ss ", IntDateVar)
```

GetFieldNameEnumerator

Retorna um enumerador.

### Sintaxe

GetEnvelopeEnumerator()

### Parâmetro

-

### Valor de retorno

Enumerador com estes métodos:

Método	Função
Current	Retorna o objeto atual.
MoveNext	Move a posição até o próximo objeto a ser processado.
Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.

CurrentJob

Trabalho atual, se encontrado. (Somente leitura)

### Métodos

GetValue

Retorna o valor da coluna dada.

### Sintaxe

GetValue(string columnName)

Parâmetro	Significado
columnName	Nome da coluna

### Valor de retorno

**Valor da coluna**

Se o tipo de dados da coluna do banco de dados for DATE, Datetime ou qualquer tipo de dados válido, a data lida geralmente será retornada no formato MM/DD/AAAA hh:mm:ss.

**Exemplo**

```
GetObject("DateVar", "MLDate")
...
StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT")
;Converter a data no formato desejado com a ajuda do objeto
MLDate
IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy
hh:mm:ss", StringDateVar )
v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy
hh:mm:ss ", IntDateVar)
```

**GetFieldNameEnumerator**

Retorna um enumerador.

**Sintaxe**

```
GetEnvelopeEnumerator()
```

**Parâmetro**

-

**Valor de retorno**

Enumerador com estes métodos:

Método	Função
Current	Retorna o objeto atual.
MoveNext	Movimenta a posição até o próximo objeto a ser processado.
Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.

**CurrentStack**

Pilha atual (Somente leitura)

**Métodos**

```
GetValue
```

Retorna o valor da coluna dada.

**Sintaxe**

```
GetValue(string columnName)
```

Parâmetro	Significado
-----------	-------------

columnName	Nome da coluna
------------	----------------

**Valor de retorno**

Valor da coluna

Se o tipo de dados da coluna do banco de dados for DATE, Datetime ou qualquer tipo de dados válido, a data lida geralmente será retornada no formato MM/DD/AAAA hh:mm:ss.

**Exemplo**

```
GetObject("DateVar", "MLDate")
...
StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT")
;Converter a data no formato desejado com a ajuda do objeto
MLDate
IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy
hh:mm:ss", StringDateVar )
v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy
hh:mm:ss ", IntDateVar)
```

GetFieldNameEnumerator

Retorna um enumerador.

**Sintaxe**

GetEnvelopeEnumerator()

**Parâmetro**

-

**Valor de retorno**

Enumerador com estes métodos:

Método	Função
Current	Retorna o objeto atual.
MoveNext	Movimenta a posição até o próximo objeto a ser processado.
Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.

**Métodos do contexto StreamPrint*****HasData***

Verifica se os dados existem para o tipo determinado.

**Sintaxe**

HasData(OdinDataRowType dataType)

### Parâmetro

Parâmetro	Significado	Valores possíveis
DataType	Tipo de dados	OdinDataRowType.Process OdinDataRowType.Stack OdinDataRowType.Envelope OdinDataRowType.Job OdinDataRowType.VirtualJob

### Valor de retorno

True: A linha de dados especificada existe.

False: A linha de dados especificada não existe.

### Contexto OnError

Propriedades do contexto OnError

Propriedade	Significado
LastErrorMessage	Mensagem de último erro

### Contexto OnNavigateObject

Propriedades do contexto OnNavigateObject

Propriedade	Significado
Block	Permite o acesso a propriedades de bloco de texto.
Cancellation	Cancela navegação: True/False
ComponentType	Tipo de objeto <b>Valores possíveis:</b> TextSystemObjectType.DocumentReference TextSystemObjectType.Block TextSystemObjectType.Variable

DocumentReference	Permite o acesso a propriedades de referência de documento.
SkipChildren	Ignorar objetos filhos: True/False
Variable	Permite o acesso a propriedades de variável.

## Exemplo

O script a seguir verifica quais objetos foram ativados manualmente por um usuário e os grava na variável **System.UsedBlocks**.

```
;Standard script to filter UsedBlocks according to standard modus4 logic
;
if (ComponentType = TextSystemObjectType.Block)
;
    ;This IF-query checks if the rule for the modus 4 filter is met.
    ;Is the text block Kann (BlockType = 1)
    ;Is the text block visible (also takes account of parent text blocks
(IsVisible)
    ;Was the block inserted? (IsInserted)
    if ((Block.BlockType = "1") and (Block.IsVisible = true) and
(Block.IsInserted = true))
        ;
        if (System.UsedBlocks = "")
            System.UsedBlocks = Block.ObjectId
        else
            System.UsedBlocks = System.UsedBlocks + ", " + Block.ObjectId
        end-if
    ;
end-if
;
;If the text block was not inserted, then its children do not need to be
checked
if ((Block.IsInserted = false) or (Block.IsVisible = false))
    SkipChildren = true
end-if
;
end-if
```

## Contexto OnProcessError

Propriedades do contexto OnProcessError

[▶ Mostrar tudo](#)

Propriedade	Significado
-------------	-------------

LastErrorMessage	Última mensagem de erro				
ModusProcess	<p>Fornece as seguintes propriedades e métodos:</p> <p><b>Propriedades</b></p> <p>Objectname: nomes do processo</p> <p>SystemOld: ID do sistema em que o processo é executado</p> <p><b>Métodos</b></p> <p>Cancel</p> <p><b>Sintaxe</b></p> <p>Cancel()</p> <p><b>Parâmetro</b></p> <p>-</p> <p><b>Valor de retorno</b></p> <p>-</p> <p><b>Exemplo</b></p> <pre>ModusProcess.Cancel()</pre> <p>GetComponent</p> <p>Permite o acesso a propriedades e métodos dos componentes.</p> <p><b>Sintaxe</b></p> <pre>GetComponent(string componentName)</pre> <p><b>Parâmetro</b></p> <table border="1"> <thead> <tr> <th>Parâmetro</th> <th>Significado</th> </tr> </thead> <tbody> <tr> <td>componentName</td> <td>Nome do componente</td> </tr> </tbody> </table> <p><b>Valor de retorno</b></p> <p>-</p> <p><b>Exemplo</b></p> <pre>DocumentCollection = ModusProcess.GetComponent("DocumentCollectionManager")</pre> <p>SendInfoMessage</p> <p>Envia uma mensagem de informação a clientes conectados por controle remoto. Um carimbo de data/hora é inserido antes de cada mensagem.</p> <p><b>Sintaxe</b></p>	Parâmetro	Significado	componentName	Nome do componente
Parâmetro	Significado				
componentName	Nome do componente				

	<p><code>SendInfoMessage(string message)</code></p> <p><b>Parâmetro</b></p> <table border="1" data-bbox="448 384 1321 516"> <thead> <tr> <th>Parâmetro</th> <th>Significado</th> </tr> </thead> <tbody> <tr> <td>message</td> <td>A mensagem de informação (InfoMessage) a ser enviada</td> </tr> </tbody> </table> <p><b>Valor de retorno</b></p> <p>-</p> <p><b>Exemplo</b></p> <pre>ModusProcess.SendInfoMessage("This is the InfoText")</pre> <p>Stop</p> <p>Termina os processos em execução.</p> <p><b>Sintaxe</b></p> <pre>Stop()</pre> <p><b>Parâmetro</b></p> <p>-</p> <p><b>Valor de retorno</b></p> <p>-</p> <p><b>Exemplo</b></p> <pre>ModusProcess.Stop()</pre>	Parâmetro	Significado	message	A mensagem de informação (InfoMessage) a ser enviada
Parâmetro	Significado				
message	A mensagem de informação (InfoMessage) a ser enviada				
ResetError	<p>Permite redefinir o estado de um erro em um script OnError.</p> <p>True: evita o cancelamento de um processo no caso de um erro.</p> <p>False: permite que o processo seja cancelado.</p> <p>Se as atividades forem executadas em um temporizador, o processo atual é executado novamente no intervalo seguinte do temporizador. Caso contrário, o processo é terminado.</p>				
ResumeOptions	<p><b>Valores possíveis:</b></p> <p><code>ProcessResumeOptions.NextActivity</code>: o processo é retomado com a atividade seguinte.</p> <p><code>ProcessResumeOptions.FromStart</code>: o processo começa do início.</p> <p><code>ProcessResumeOptions.FromStartWithIntervall</code>: o processo começa do início no próximo intervalo.</p> <p><b>Observe:</b></p> <p>Pools de variável não são redefinidas.</p>				



A definição das ResumeOptions aplica-se somente quando o processo é executado em um temporizador.

## Contexto Process

### Geral

Contexto para scripts de processo.

### Propriedades do contexto Process

Propriedade	Significado				
ModusProcess	<p>Fornece as seguintes propriedades e métodos:</p> <p><b>Propriedades</b>  Objectname: Nomes dos processos  SystemOld: ID do sistema em que o processo é executado</p> <p><b>Métodos</b>  Cancelar</p> <p><b>Sintaxe</b>  Cancel()</p> <p><b>Parâmetro</b>  -</p> <p><b>Valor de retorno</b>  -</p> <p><b>Exemplo</b>  ModusProcess.Cancel ( )  GetComponent  Permite o acesso às propriedades e métodos dos componentes.</p> <p><b>Sintaxe</b>  GetComponent(string componentName)</p> <p><b>Parâmetro</b></p> <table border="1"> <thead> <tr> <th>Parâmetro</th> <th>Significado</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table>	Parâmetro	Significado		
Parâmetro	Significado				

componentName	Nome do componente
---------------	--------------------

**Valor de retorno**

-

**Exemplo**

```
DocumentCollection =
ModusProcess.GetComponent("DocumentCollectionManager")
```

**SendInfoMessage**

Envia uma mensagem informativa para se conectar aos clientes de controle remoto. Um carimbo é inserido antes de cada mensagem.

**Sintaxe**

```
SendInfoMessage(string message)
```

**Parâmetro**

Parâmetro	Significado
message	A mensagem informativa (InfoMessage) a ser enviada

**Valor de retorno**

-

**Exemplo**

```
ModusProcess.SendInfoMessage("This is the InfoText")
```

**Interrompimento**

Finaliza as execuções dos processos.

**Sintaxe**

```
Stop()
```

**Parâmetro**

-

**Valor de retorno**

-

**Exemplo**

```
ModusProcess.Stop()
```

## Contexto ProcessTimer

### Geral

Contexto para scripts de temporizador.

## Propriedades do contexto ProcessTimer

Propriedade	Significado				
ExecuteTimer	Executar a lógica do temporizador: True / False				
Interval	Intervalo em milissegundos. O intervalo em que o serviço de temporizador aciona o temporizador do processo.				
ModusProcess	<p>Fornecer as seguintes propriedades e métodos:</p> <p><b>Propriedades</b>  Objectname: nomes do processo  SystemOld: ID do sistema em que o processo é executado</p> <p><b>Métodos</b>  Cancel</p> <p><b>Sintaxe</b>  Cancel()</p> <p><b>Parâmetro</b>  -</p> <p><b>Valor de retorno</b>  -</p> <p><b>Exemplo</b>  ModusProcess.Cancel ( )  GetComponent  Permite o acesso a propriedades e métodos dos componentes.</p> <p><b>Sintaxe</b>  GetComponent(string componentName)</p> <p><b>Parâmetro</b></p> <table border="1"> <thead> <tr> <th>Parâmetro</th> <th>Significado</th> </tr> </thead> <tbody> <tr> <td>componentName</td> <td>Nome do componente</td> </tr> </tbody> </table> <p><b>Valor de retorno</b>  -</p> <p><b>Exemplo</b>  DocumentCollection =</p>	Parâmetro	Significado	componentName	Nome do componente
Parâmetro	Significado				
componentName	Nome do componente				

```
ModusProcess.GetComponent("DocumentCollectionManager")
```

### SendInfoMessage

Envia uma mensagem de informação a clientes conectados por controle remoto. Um carimbo de data/hora é inserido antes de cada mensagem.

#### Sintaxe

```
SendInfoMessage(string message)
```

#### Parâmetro

Parâmetro	Significado
message	A mensagem de informação (InfoMessage) a ser enviada

#### Valor de retorno

-

#### Exemplo

```
ModusProcess.SendInfoMessage("This is the InfoText")
```

### Stop

Termina os processos em execução.

#### Sintaxe

```
Stop()
```

#### Parâmetro

-

#### Valor de retorno

-

#### Exemplo

```
ModusProcess.Stop()
```

## Contexto SelectionQueryScript

Propriedades do contexto SelectionQueryScript

Propriedade	Significado
Bd	Alias de BD

## Contexto TextComponent

### Geral

Contexto para blocos de texto.

### Propriedades do contexto TextComponent

O nome entre parênteses é o nome da propriedade no modus 4, o qual por razões de compatibilidade de manutenção descendente, também pode ser usado.

Propriedade	Significado
BlockprotectionS (Blockprotection)	Corresponde à propriedade TextComponentProtection para um bloco de texto. Consulte <a href="#">Text Block Properties</a>
BlockType (BlockType)	0 = Desativada 1 = Obrigatória 2 = Ativada 3 = Desabilitada
DataLink (objeto de dados)	Caminho para o objeto de dados. (Somente leitura)
Expandable	Define se o usuário pode ver a estrutura do bloco de texto e tem permissão para ativar ou desativar seus objetos. (True / False)
ExpandableS	Define se o usuário pode ver a estrutura do bloco de texto e tem permissão para ativar ou desativar seus objetos. (Y / N)
IsVisible	Visível - determinado pela propriedade do bloco de texto/documento pai. (somente leitura)
TextBlockProtection	TextBlockProtection (True / False)
Title (Title)	O valor definido aqui é o nome do bloco de texto que o usuário vê no cliente.
Visible	Visível (True / False)
VisibleS	Visível (Y / N)
XPathExpression	Consulte <a href="#">Text Block Properties</a>
ErrorMessage	Mensagem de erro. Se um valor é atribuído, a geração é abortada com a mensagem de erro correspondente.

InsertObject	Inserir objeto True / False
InsertObjectS (InsertObject)	Inserir objeto Y / N
IsInserted	O objeto foi inserido True / False (somente leitura)
ObjectId (ObjectIndex)	ID do objeto (somente leitura)
ObjectName	Nome do objeto (somente leitura)
ParentBlock	Bloco de texto superior (somente leitura)
ParentDocument	Documento superior (somente leitura)

## Métodos do contexto TextComponent

### ***LoadExternalDocument***

Carrega um documento existente de um dispositivo de armazenamento de dados e substitui o bloco de texto Modus.

A função deve ser usada em um script BeforeInsert.

O documento a ser carregado deve ser um arquivo docx criado com o Word2011 ou mais avançado.

O bloco de texto que utiliza o documento do Word não pode conter blocos de texto aninhados ou variáveis.

### **Sintaxe**

LoadExternalDocument(fileName)

LoadExternalDocument(bytes)

### **Parâmetro**

Parâmetro	Significado
filename	Nome do bloco de texto no documento
bytes	Para o carregamento de um documento de uma matriz de bytes

### **Valor de retorno**

-

### **Exemplo**

```
LoadExternalDocument("C:\Documents\How_To.docx")
```

```
LoadExternalDocument(System.IO.File.ReadAllBytes("c:\Documents\Account.docx"))
```

## Navegar

Permite que você navegue pelos objetos filhos do objeto atual.

### Sintaxe

```
Navigate(string onObjectScript, bool onlyActiveNodes, params TextSystemObjectType[] filterTypes)
```

### Parâmetro

Parâmetro	Significado
onObjectScript	Script executado para cada um dos tipos de objetos especificados encontrados. Este script deve ter o contexto OnNavigateObject atribuído a ele.
onlyActiveNodes	True: Somente objetos ativos navegam False: Todos os objetos navegam
filterTypes	Filtro que define os objetos a serem navegados. <b>Os tipos disponíveis são:</b> TextSystemObjectType.Variable TextSystemObjectType.Block TextSystemObjectType.DocumentReferences

### Valor de retorno

-

### Exemplo

```
Navigate("Scriptname", true, TextSystemObjectType.Block, TextSystemObjectType.Variable)
```

## Contexto TextDocument

### Geral

### Sintaxe

### Parâmetro

Parâmetro	Significado
-----------	-------------



### Valor de retorno

-

### Exemplo

### Propriedades do contexto TextDocument

O nome entre parênteses é o nome da propriedade no modus 4, o qual por razões de compatibilidade de manutenção descendente, também pode ser usado.

Propriedade	Significado
CustomIndex	É usado na conexão com pacotes dinâmicos para transferir os valores para documentos inseridos dinamicamente. (somente leitura)
MissingObjects	String separada por vírgula contendo as aliases dos objetos variáveis ativados não encontrados durante a geração de documentos. Uma string vazia é retornada quando nenhum objeto ausente é encontrado.
ErrorMessage	Mensagem de erro. Se um valor é atribuído, a geração é abortada com a mensagem de erro correspondente.
InsertObject	Inserir objeto True / False
InsertObjectS (InsertObject)	Inserir objeto Y / N
IsInserted	O objeto foi inserido True / False (somente leitura)
ObjectId (ObjectIndex)	ID do objeto (somente leitura)
ObjectName	Nome do objeto (somente leitura)

### Métodos do contexto TextDocument

#### ***LoadExternalDocument***

Carrega um documento de um dispositivo de armazenamento de dados e substitui o documento Modus estático.

#### **Sintaxe**

LoadExternalDocument(fileName)



LoadExternalDocument(bytes)

### Parâmetro

Parâmetro	Significado
filename	Nome do documento no pacote
bytes	Para o carregamento de um documento de uma matriz de bytes

### Valor de retorno

-

### Exemplo

```
LoadExternalDocument("C:\Documents\How_To.pdf")
```

```
LoadExternalDocument(System.IO.File.ReadAllBytes("c:\Documents\Account.pdf"))
```

### Navegar

Permite que você navegue pelos objetos filhos do objeto atual.

### Sintaxe

```
Navigate(string onObjectScript, bool onlyActiveNodes, params TextSystemObjectType[] filterTypes)
```

### Parâmetro

Parâmetro	Significado
onObjectScript	Script executado para cada um dos tipos de objetos especificados encontrados. Este script deve ter o contexto OnNavigateObject atribuído a ele.
onlyActiveNodes	True: Somente objetos ativos navegam False: Todos os objetos navegam
filterTypes	Filtro que define os objetos a serem navegados. <b>Os tipos disponíveis são:</b> TextSystemObjectType.Variable TextSystemObjectType.Block TextSystemObjectType.DocumentReferences

### Valor de retorno

-

**Exemplo**

```
Navigate("Scriptname", true, TextSystemObjectType.Block,
TextSystemObjectType.Variable)
```

**Contexto TextDocumentCollection****Geral**

Contexto para pacotes.

**Propriedades do contexto TextDocumentCollection**

O nome entre parênteses é o nome da propriedade no modus 4, o qual por razões de compatibilidade de manutenção descendente, também pode ser usado.

Propriedade	Significado
AllDocumentNames	Uma lista separa por vírgulas com os nomes de todos os documentos originais contidos em um pacote. (somente leitura)
AllDocuments	Uma lista separa por vírgulas com as IDs filho de todos os documentos originais contidos em um pacote. (somente leitura)
AllowPrint	Impressão permitida True/False
ArchiveData (Inhalt)	Dados de arquivo. Essa propriedade só é contida por razões de compatibilidade.
BlockBundle	Se o pacote deve ser encaminhado ou não: True/False
DataID	Retorna a DataID atribuída ao objeto de pacote (objeto de raiz no pacote). Retorna uma string vazia se nenhuma DataID for atribuída.
DestinationGroup (DestUserGroup)	Nome do grupo de usuários ao qual o pacote é encaminhado.
DestinationUser (DestUser)	Nome do usuário ao qual o pacote é encaminhado.
DocumentGroupIndex	BundleID (apenas leitura)
DocumentGroupName	Nome do pacote (somente leitura)
DocumentNames	Uma lista separa por vírgulas com os nomes de todos os documentos ativos. (somente leitura)
Documents	Uma lista separa por vírgulas com as IDs filho de todos os documentos ativos. (somente leitura)
HoldTime	Tempo de espera de uma pilha de impressão direta.

Priority	Prioridade de uma pilha de impressão direta.
State	Status do pacote (somente leitura) <b>Valores possíveis:</b> DocumentCollectionState.None: estado inicial DocumentCollectionState.DataProcessed: dados atribuídos e processados (pacote dinâmico) DocumentCollectionState.Created: documentos criados. DocumentCollectionState.Printed: documentos impressos.
ErrorMessage (FatalerFehler)	Mensagem de erro. Se um valor é atribuído, a geração é abortada com a mensagem de erro correspondente.
InsertObject	Inserir objeto True / False
InsertObjectS (ObjektEinfuegen)	Inserir objeto Y / N
IsInserted	O objeto foi inserido True / False (somente leitura)
ObjectId (ObjectIndex)	ID do objeto (somente leitura)
ObjectName	Nome do objeto (somente leitura)

## Métodos do contexto TextDocumentCollection

### **AddDocumentByName**

Adiciona um documento a um pacote. Esse documento é mencionado por seu nome no repositório.

#### **Sintaxe**

```
AddDocumentByName(string objectName, string childId, TextInsertionRule insertRule, string beforeInsertName, string afterInsertName, string conditionName, string defineSortName, string defineGroupName, string defineArchiveName, int numberOfCopies, string printerName, string formName, bool changingAllowed, OdinPrintTime printTime, bool doArchive, string customIndex, string documentName, string dataId, string password, bool isVisible, bool isExpandable, DocumentType documentType, string title, string customUIName)
```

```
AddDocumentByName(string objectName, string childId, TextInsertionRule insertRule, string beforeInsertName, string afterInsertName, string conditionName, string defineSortName, string defineGroupName, string defineArchiveName, int numberOfCopies, string printerName, string formName, bool changingAllowed, OdinPrintTime printTime, bool doArchive, string customIndex, string documentName, string dataId, string password, bool isVisible, bool isExpandable, DocumentType documentType, string title, string customUIName, OdinPrinterType printerType)
```

#### **Parâmetro**

Parâmetro	Significado
objectName	Nome do documento no pacote
childId	ID de documento definível livremente. Ela deve ser exclusiva no pacote.
insertRule	TextInsertionRule <b>Os tipos disponíveis são:</b> TextInsertionRule.Activated TextInsertionRule.Deactivated TextInsertionRule.Required TextInsertionRule.Disabled
beforeInsertName	Script BeforeInsert
afterInsertName	Script AfterInsert
conditionName	Condição
defineSortName	Script DefinitionSorting
defineGroupName	Script DefinitionGrouping
defineArchiveName	Script DefinitionArchiving
numberOfCopies	Número de cópias
printerName	Impressora
formName	Nome do formulário
changingAllowed	Pode ser alterado True/False
printTime	Hora da impressão <b>Os tipos disponíveis são:</b> OdinPrintTime.Direct: Imediatamente : ShiftedOdinPrintTime.Shifted OdinPrintTime.Unknown: Desconhecido
doArchive	Arquivo True / False
customIndex	Índice de definição livre, por exemplo, para consultas subsequentes

documentName	Nome do documento
dataId	Seleção
password	Senha
isVisible	Visível (True/False)
isExpandable	Expandível True/False
documentType	Tipo de documento <b>Os tipos disponíveis são:</b> DocumentType.Unknown DocumentType.DynamicWord DocumentType.StaticWord DocumentType.StaticPdf DocumentType.StaticTiff DocumentType.StaticXps
title	Título do documento no pacote. Se este parâmetro estiver definido, ele substitui <b>objectName</b> .
customUIName	CustomUI
printerType	Tipo de impressora <b>Os tipos disponíveis são:</b> OdinPrinterType.Online: Impressora online OdinPrinterType.Email: Impressora de email  Se este parâmetro não estiver definido, o tipo de impressora OdinPrinterType.Online é usado.

**Valor de retorno**

-

**Exemplo**

```
AddDocumentByName("DocumentOriginal", "ChildID_Original", TextInsertionRule.Activated, "BeforeInsertDocument", "AfterInsertDocument", "True_Document", "SortingDocument", "GroupingDocument", "ArchivingDocument", 1, "Printer", "Form", true, OdinPrintTime.Shifted, false, "CustomIndex", "StaticPdf", "Partner_V2", "Password", true, true, DocumentType.StaticPdf, "Title_Original", "CustomUIName")
```

## **AddDocumentByObjectId**

Adiciona um documento a um pacote. O documento é referenciado por sua ID de objeto.

### **Sintaxe**

AddDocumentByObjectId(string objectName, string childId, TextInsertionRule insertRule, string beforeInsertOld, string afterInsertOld, string conditionOld, string defineSortOld, string defineGroupOld, string defineArchiveOld, int numberOfCopies, string printerOld, string formOld, bool changingAllowed, OdinPrintTime printTime, bool doArchive, string customIndex, string documentOld, string dataId, string password, bool isVisible, bool isExpandable, DocumentType documentType, string title, string customUIOld)

### **Parâmetro**

<b>Parâmetro</b>	<b>Significado</b>
objectId	Nome do documento no pacote
childId	ID de documento definível livremente. Ela deve ser exclusiva no pacote.
insertRule	TextInsertionRule <b>Os tipos disponíveis são:</b> TextInsertionRule.Activated TextInsertionRule.Deactivated TextInsertionRule.Required TextInsertionRule.Disabled
beforeInsertOld	ID do objeto do script BeforeInsert
afterInsertOld	ID do objeto do script AfterInsert
conditionOld	ID do objeto de Condition
defineSortOld	ID do objeto do script DefinitionSorting
defineGroupOld	ID do objeto do script DefinitionGrouping
defineArchiveOld	ID do objeto do script DefinitionArchiving
numberOfCopies	Número de cópias
printerOld	ID do objeto da impressora
formOld	ID do objeto do formulário
changingAllowed	Pode ser alterado True/False

printTime	Hora da impressão <b>Os tipos disponíveis são:</b> OdinPrintTime.Direct: Imediatamente : ShiftedOdinPrintTime.Shifted OdinPrintTime.Unknown: Desconhecido
doArchive	Arquivo True/False
customIndex	Índice de definição livre, por exemplo, para consultas subsequentes
documentOid	ID do objeto do documento
dataId	Selection
password	Senha
isVisible	Visível (True/False)
isExpandable	Expansível True/False
documentType	Tipo de Documento <b>Os tipos disponíveis são:</b> DocumentType.Unknown DocumentType.DynamicWord DocumentType.StaticWord DocumentType.StaticPdf DocumentType.StaticTiff DocumentType.StaticXps
title	Título do documento no pacote. Se este parâmetro estiver definido, ele substitui <b>objectName</b> .
customUIName	ID do objeto do script CustomUI

**Valor de retorno**

-

**Exemplo**

```
AddDocumentById("Dokument", "ChildID_Original2",
TextInsertionRule.Activated, "15072011-226-sc-ts", "15072011-227-sc-
ts", "10072011-186-cd-ts", "15072011-230-sc-ts", "15072011-229-sc-ts", "15072011-
228-sc-ts", 0, "04092011-43-pr-ts", "18112011-482-fo-ts", true,
```

```
OdinPrintTime.Shifted, false, "CustomIndex_2", "14072011-204-td-ts",
"Partner_V2", "Kennwort", true, true, DocumentType.StaticPdf,
"Titel_Original2", "06092011-67-cui-ts")
```

### **AddDocumentCopyByName**

Adiciona um documento a um pacote. Esse documento é mencionado por seu nome no repositório.

#### **Sintaxe**

```
AddDocumentCopyByName(string objectName, string childId, TextInsertionRule insertRule, string
beforeInsertName, string afterInsertName, string conditionName, string defineSortName, string
defineGroupName, string defineArchiveName, int numberOfCopies, string printerName, string
formName, bool changingAllowed, OdinPrintTime printTime, bool doArchive, string customIndex, string
copyLabel, string originalId, bool isVisible, string title)
```

```
AddDocumentCopyByName(string objectName, string childId, TextInsertionRule insertRule, string
beforeInsertName, string afterInsertName, string conditionName, string defineSortName, string
defineGroupName, string defineArchiveName, int numberOfCopies, string printerName, string
formName, bool changingAllowed, OdinPrintTime printTime, bool doArchive, string customIndex, string
copyLabel, string originalId, bool isVisible, string title, OdinPrinterType printerType)
```

#### **Parâmetro**

Parâmetro	Significado
objectName	Nome do documento no pacote
childId	ID de documento definível livremente. Ela deve ser exclusiva no pacote.
insertRule	TextInsertionRule <b>Os tipos disponíveis são:</b> TextInsertionRule.Activated TextInsertionRule.Deactivated TextInsertionRule.Required TextInsertionRule.Disabled
beforeInsertName	Script BeforeInsert
afterInsertName	Script AfterInsert
conditionName	Condição
defineSortName	Script DefinitionSorting
defineGroupName	Script DefinitionGrouping
defineArchiveName	Script DefinitionArchiving



numberOfCopies	Número de cópias
printerName	Impressora
formName	Nome do formulário
changingAllowed	Pode ser alterado True/False
printTime	Hora da impressão <b>Os tipos disponíveis são:</b> OdinPrintTime.Direct: Imediatamente : ShiftedOdinPrintTime.Shifted OdinPrintTime.Unknown: Desconhecido
doArchive	Arquivamento True / False
customIndex	Índice de definição livre, por exemplo, para consultas subsequentes
copyLabel	Texto na cópia
originalId	ChildId do original
isVisible	Visível (True/False)
title	Título do documento no pacote. Se este parâmetro estiver definido, ele substitui <b>objectName</b> .
printerType	Tipo de impressora <b>Os tipos disponíveis são:</b> OdinPrinterType.Online: Impressora online OdinPrinterType.Email: Impressora de email  Se este parâmetro não estiver definido, o tipo de impressora OdinPrinterType.Online é usado.

**Valor de retorno**

-

**Exemplo**

```
AddDocumentCopyByName("Document_Copy", "ChildID_Copy",
TextInsertionRule.Activated, "BeforeInsertDocument", "AfterInsertDocument",
"True_Document", "SortingDocument", "GroupingDocument", "ArchivingDocument", 1
```

```
, "Printer", "Form", true, OdinPrintTime.Shifted, false, "CustomIndex_K",
"CopyLabel", "ChildID_Original", true, "CopyTitle")
```

### **AddDocumentCopyByObjectID**

Adiciona um documento a um pacote. Esse documento é mencionado por sua ID no repositório.

#### **Sintaxe**

AddDocumentCopyByObjectID(string objectName, string childId, TextInsertionRule insertRule, string beforeInsertOld, string afterInsertOld, string conditionOld, string defineSortOld, string defineGroupOld, string defineArchiveOld, int numberOfCopies, string printerOld, string formOld, bool changingAllowed, OdinPrintTime printTime, bool doArchive, string customIndex, string copyLabel, string originalId, bool isVisible, string title)

#### **Parâmetro**

<b>Parâmetro</b>	<b>Significado</b>
objectId	Nome do documento no pacote
childId	ID de documento definível livremente. Ela deve ser exclusiva no pacote.
insertRule	TextInsertionRule <b>Os tipos disponíveis são:</b> TextInsertionRule.Activated TextInsertionRule.Deactivated TextInsertionRule.Required TextInsertionRule.Disabled
beforeInsertOid	ID do objeto do script BeforeInsert
afterInsertOid	ID do objeto do script AfterInsert
conditionOid	ID do objeto de Condition
defineSortOid	ID do objeto do script DefinitionSorting
defineGroupOid	ID do objeto do script DefinitionGrouping
defineArchiveOid	ID do objeto do script DefinitionArchiving
numberOfCopies	Número de cópias
printerOid	ID do objeto da impressora
formOid	ID do objeto do formulário

changingAllowed	Pode ser alterado True/False
printTime	Hora da impressão <b>Os tipos disponíveis são:</b> OdinPrintTime.Direct: Imediatamente ShiftedOdinPrintTime.Shifted OdinPrintTime.Unknown: Desconhecido
doArchive	Arquivo True/False
customIndex	Índice de definição livre, por exemplo, para consultas subsequentes
copyLabel	Texto na cópia
originalId	ChildId do original
isVisible	Visível (True/False)
title	Título do documento no pacote. Se este parâmetro estiver definido, ele substitui <b>ObjectName</b> .

**Valor de retorno**

-

**Exemplo**

```
AddDocumentCopyByObjectId("Document_Copy", "ChildId_Copy", TextInsertionRule.Activated, "15072011-226-sc-ts", "15072011-227-sc-ts", "10072011-186-cd-ts", "15072011-230-sc-ts", "15072011-229-sc-ts", "15072011-228-sc-ts", 1, "04092011-43-pr-ts", "18112011-482-fo-ts", true, OdinPrintTime.Shifted, false, "CustomIndexCopy", "CopyLabel", "ChildID_Original", true, "Titlecopy")
```

**Navegar**

Permite que você navegue pelos objetos filhos do objeto atual.

**Sintaxe**

```
Navigate(string onObjectScript, bool onlyActiveNodes, params TextSystemObjectType[] filterTypes)
```

**Parâmetro**

Parâmetro	Significado
onObjectScript	Script executado para cada um dos tipos de objetos especificados encontrados. Este script deve ter o contexto OnNavigateObject atribuído a ele.

onlyActiveNodes	True: Somente objetos ativos navegam False: Todos os objetos navegam
filterTypes	Filtro que define os objetos a serem navegados. <b>Os tipos disponíveis são:</b> TextSystemObjectType.Variable TextSystemObjectType.Block TextSystemObjectType.DocumentReferences

**Valor de retorno**

-

**Exemplo**

```
Navigate("Scriptname", true, TextSystemObjectType.Block,
TextSystemObjectType.Variable)
```

***SetArchiveAttribute***

Define um atributo para o processo de arquivamento.

**Sintaxe**

```
SetArchiveAttribute(string name, string value)
```

**Parâmetro**

Parâmetro	Significado
name	Nome do atributo
value	Valor do atributo

**Valor de retorno**

-

**Exemplo**

Consulte [Caso especial: Atributos de arquivo idênticos para todos os documentos de um pacote](#)

***SetDataContext***

Define o contexto de dados atual.

Scripts que contêm instruções **SetVariable** se referem a esse conjunto de dados.

Se **dataid** retornar uma string nula ou vazia, o contexto de dados será redefinido.

Isso significa que as instruções `SetVariable` resultam em um erro.

Uma exceção é lançada, se nenhum contexto de dados existir para a ID especificada.

### Sintaxe

```
SetDataContext(string dataId)
```

### Parâmetro

Parâmetro	Significado
<code>dataId</code>	A ID do contexto de dados a ser definido.

### Valor de retorno

-

## Contexto `TextDocumentReference`

### Geral

Contexto para documentos em um pacote.

### Propriedades do contexto `TextDocumentReference`

O nome entre parênteses é o nome da propriedade no modus 4, o qual por razões de compatibilidade de manutenção descendente, também pode ser usado.

Propriedade	Significado
<code>CustomIndex</code>	É usado na conexão com pacotes dinâmicos para transferir os valores para documentos inseridos.
<code>DoArchive</code>	Arquivo True/False
<code>DocumentGroupIndex</code>	BundleId (somente leitura)
<code>DocumentGroupName</code>	Nome do pacote (somente leitura)
<code>Expandable</code>	Expansível True/False
<code>ExpandableS</code>	Expansível (Y / N)
<code>FormName</code>	Leia o nome do formulário atribuído ou defina o formulário especificando um nome.
<code>Agrupamento</code>	Informações de agrupamento
<code>InsertionRule</code>	0 = Desativada 1 = Obrigatória

	<p>2 = Ativada</p> <p>3 = Desabilitada</p> <p>Essa propriedade pode ser usada nos scripts OnNavigateObject.</p>
IsActivated	<p>A referência de documento é ativada no pacote True/False (somente leitura)</p> <p>O IsActivated é true quando a propriedade de documento TextInsertionRule é definida como Obrigatória ou Ativada.</p>
IsCopy	Cópia True/False (somente leitura)
MissingObjects	<p>String separada por vírgula contendo as aliases dos objetos variáveis ativados não encontrados durante a geração de documentos.</p> <p>Uma string vazia é retornada quando nenhum objeto ausente é encontrado.</p>
NumberOfCopies	Número de cópias
PrinterName	Nome da impressora atribuída (somente leitura)
Sorting	Informações de classificação
Title (Title)	O valor definido aqui é o nome do documento que o usuário vê no cliente.
Transportpools	<p>Pools cujas variáveis são transferidas para o cliente.</p> <p>Os scripts executados no contexto <b>TextDocumentReference</b> podem definir variáveis do <b>TransportPools</b> para transferir as variáveis dos pools especificados para o cliente.</p> <p>A avaliação das variáveis definidas no <b>TransportPools</b> é executada no final do processamento de um documento.</p> <p>Os pools são definidos por seus nomes em uma lista separada por vírgula. Isso pode referir aos pools que já existem, por exemplo, ModusUser, e a pools que são criados pelo comando 'CreatePool' no MonaLisa.</p> <p><b>Exemplo: Configuração no DefinitionArchiving</b></p> <pre>TransportPools = "MyPool,ModusUser"</pre> <p>A transferência de pools serializados é implementada como um subelemento da estrutura do documento no elemento &lt;archiving&gt;.</p> <p>Após o final do processamento, a estrutura de retorno do documento contém o subelemento &lt;archiving&gt;.</p> <pre>&lt;object title="Partner_Application" type="D"</pre>

	<pre> name="Partner_Application" index="02062008-823-tdr- itl1" reference="10DE6C09-79CF-4547-97AA-6302CF976185" state="1" enabled="1" print_asktime="0" form="TestForm" printer="" printer_type="" printtime="2" print_changeable="1" copycount="0"   copy="0" password="modus" filename="Partner_Application_10DE6C09-79CF-4547-97AA- 6302CF976185_O.docx"&gt;   &lt;archiving&gt;     &lt;pools&gt;       &lt;pool name="MyPool"&gt;         &lt;poolvar name="TESTVALUE" type="S"&gt;MyVariableInMyPool&lt;/poolvar&gt;       &lt;/pool&gt;     &lt;/pools&gt;   &lt;/archiving&gt; &lt;/object&gt; </pre>
Visible	Visível (True/False)
VisibleS	Visível (Y / N)
ErrorMessage (FatalerFehler)	Mensagem de erro. Se um valor é atribuído, a geração é abortada com a mensagem de erro correspondente.
InsertObject	Inserir objeto True / False
InsertObjectS (InsertObject)	Inserir objeto Y / N
IsInserted	O objeto foi inserido True / False (somente leitura)
ObjectId (ObjectIndex)	ID do objeto (somente leitura)
ObjectName	Nome do objeto (somente leitura)

## Métodos do contexto TextDocumentReference

### **DefinePrinterByName**

Define uma impressora por seu nome e tipo.

#### **Sintaxe**

```
DefinePrinterByName(string printerName, ModusSuiteObjectType printerType, string formName, bool allowChange, OdinPrintTime printTime);
```

**Parâmetro**

Parâmetro	Significado
printerName	Nome da impressora
printerType	Tipo da impressora. <b>Os tipos disponíveis são:</b> ModusSuiteObjectType.Printer ModusSuiteObjectType.EMailPrinter ModusSuiteObjectType.OutsourcingPrinter
formName	Nome do formulário
allowChange	Pode ser alterado True/False
printTime	Tempo da impressão <b>Os tipos disponíveis são:</b> OdinPrintTime.Direct OdinPrintTime.Shifted OdinPrintTime.Unknown

**Valor de retorno**

-

***DefinePrinterByOld***

Define a impressora por uso de suas ID de objeto.

**Sintaxe**

DefinePrinterByOld(string printerOld, string formOld, bool allowChange, OdinPrintTime printTime)

**Parâmetro**

Parâmetro	Significado
printerOid	ID do objeto da impressora.
formOid	ID do objeto do formulário.
allowChange	Pode ser alterado True/False
printTime	Tempo da impressão



**Os tipos disponíveis são:**

OdinPrintTime.Direct

OdinPrintTime.Shifted

OdinPrintTime.Unknown

#### Valor de retorno

-

### ***SetArchiveAttribute***

Define um atributo para o processo de arquivamento.

#### Sintaxe

SetArchiveAttribute(string name, string value)

#### Parâmetro

Parâmetro	Significado
name	Nome do atributo
value	Valor do atributo

#### Valor de retorno

-

#### Exemplo

Consulte [Caso especial: Atributos de arquivo idênticos para todos os documentos de um pacote](#)

## Contexto TextVariable

### Geral

Contexto para variáveis de texto.

### Propriedades do contexto TextVariable

O nome entre parênteses é o nome da propriedade no modus 4, o qual por razões de compatibilidade de manutenção descendente, também pode ser usado.

Propriedade	Significado
AttributeName (variável de banco de dados)	Nome do atributo (campo do banco de dados ou variável de pool) (somente leitura)
Content	Conteúdo da variável

DataLink (objeto de dados)	Caminho para o objeto de dados ou o nome de um pool de variável. (somente leitura)
FormattedContent	O conteúdo formatado da variável (somente leitura)
IsFreelInput	É avaliada quando uma ajuda de valor é atribuída. True: um valor pode ser inserido manualmente. False: apenas um valor da ajuda de valor pode ser selecionado.
IsFreelInputS (InputEnabled)	consulte IsFreelInput (Y / N)
IsInputRequired	A variável é obrigatória True / False
IsInputRequiredS (InputRequired)	A variável é obrigatória Y/N
IsManual	Entrada manual possível (True / False)
IsManualS (Input)	Entrada manual possível (Y/N)
Prompt	Texto do prompt para variáveis manuais.
XPathExpression	Expressão de XPath (somente leitura)
ParentBlock	Bloco de texto pai (somente leitura)
ParentDocument	Documento pai (somente leitura)
ErrorMessage	Mensagem de erro Se um valor é atribuído, a geração é abortada com a mensagem de erro correspondente.
InsertObject	Inserir objeto True / False
InsertObjectS (InsertObject)	Inserir objeto Y / N
IsInserted	O objeto foi inserido True / False (somente leitura)
ObjectId (ObjectIndex)	ID do objeto (somente leitura)
ObjectName	Nome do objeto (somente leitura)

## Métodos do contexto TextVariable

### **Navegar**

Permite que você navegue pelos objetos filhos do objeto atual.

### Sintaxe

```
Navigate(string onObjectScript, bool onlyActiveNodes, params TextSystemObjectType[] filterTypes)
```

### Parâmetro

Parâmetro	Significado
onObjectScript	Script executado para cada um dos tipos de objetos especificados encontrados. Este script deve ter o contexto OnNavigateObject atribuído a ele.
onlyActiveNodes	True: Somente objetos ativos navegam False: Todos os objetos navegam
filterTypes	Filtro que define os objetos a serem navegados. <b>Os tipos disponíveis são:</b> TextSystemObjectType.Variable TextSystemObjectType.Block TextSystemObjectType.DocumentReferences

### Valor de retorno

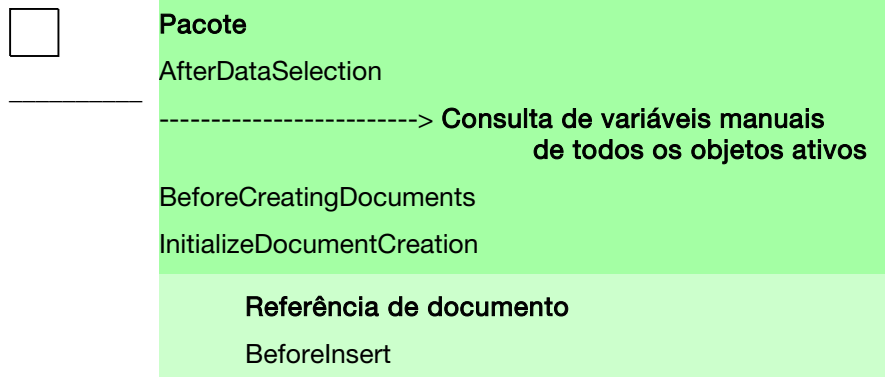
-

### Exemplo

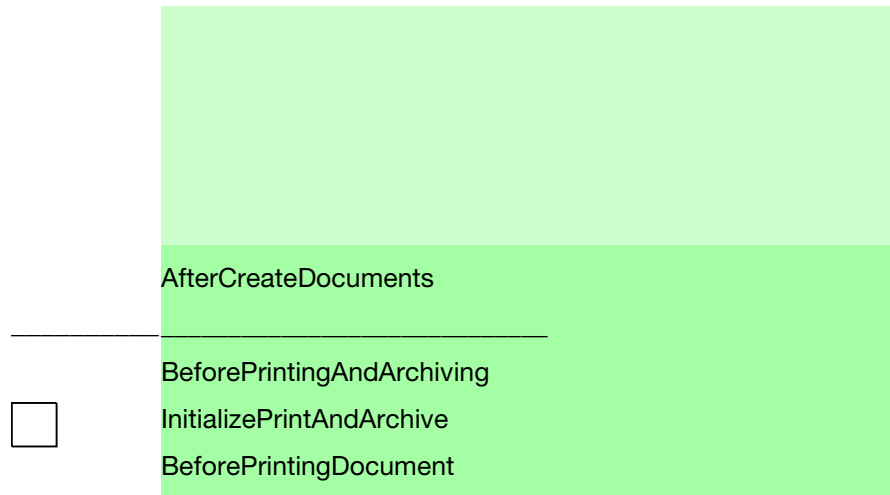
```
Navigate("Scriptname", true, TextSystemObjectType.Block,  
TextSystemObjectType.Variable)
```

## Ordem do script

A ilustração a seguir mostra a ordem na qual os scripts e as condições são executadas em um pacote:








## Tabelas de aliases

As tabelas de aliases são usadas para armazenar atribuições de blocos de texto e documentos variáveis.

**Para criar uma tabela de aliases para um bloco de texto variável, execute estas etapas:**

1. Selecione **Novo -> Tabela de aliases** do menu de contexto do Navigator.
2. Digite o nome da tabela de aliases na caixa de diálogo seguinte.
3. Vá para a guia **Tabela de aliases de pesquisa** e clique no botão **Nova chave** .  
Alternativamente, a função também pode ser selecionada do menu de contexto.
4. Insira o valor contido na variável correspondente como o nome da nova chave.  
**Exemplo:**  
uma variável contém um código de país. Dependendo do valor dessa variável, um bloco de texto diferente pode ser inserido cada vez em um documento.  
Crie uma chave para cada valor possível da variável, por exemplo, DE, GB, NL, FR, etc.
5. Selecione uma chave na janela do editor.
6. Vá para a guia **Tabela de aliases de pesquisa** e clique no botão **Nova referência** .
7. Como alternativa, a função também pode ser selecionada do menu de contexto.
8. No diálogo a seguir, selecione um bloco de texto a ser inserido quando a variável contém o valor da chave.

Para obter mais informações, consulte o capítulo [Objetos variáveis](#).

## Tabelas de pesquisa

É possível usar as tabelas de pesquisa para especificar as atribuições de valor para substituir os conteúdos das variáveis durante a criação do documento.

**Conclua as seguintes etapas para criar uma Tabela de pesquisa :**

1. Selecione **Novo -> Tabela de pesquisa** no menu de contexto do Navigator.
2. Insira o nome da tabela de pesquisa na caixa de diálogo seguinte.
3. Defina os pares de valor na janela do editor no formulário de Conteúdo da variável=valor a ser inserido.  
Cada par de valor deve ser definido em uma linha separada.

**Exemplo:**

Uma variável contém um código de saudação. Crie os seguintes pares de valor para a variável:

001=Sra.

002=Sr.

003=Empresa

Se essa tabela de pesquisa for atribuída a uma variável de documento que contenha o valor **003**, o conteúdo da variável é substituído pelo conteúdo **Empresa**.

# Trabalhos de transferência

## Informações gerais

O controle de versão é um componente opcional do ModusOne e está disponível apenas quando seu uso foi licenciado.

Quase todos os tipos de objeto administrados no ModusOne Studio podem ter versão.

### Os tipos de objeto a seguir não podem ter versão:

- Pastas
- Perfis
- Funções
- Mapeamentos de função
- Aliases de DB
- Trabalhos de transferência
- Sistemas

### Novos objetos

Um novo objeto é sempre criado como uma versão de trabalho com o número 0.1.

### Migração

Ao migrar do modus 4 para o ModusOne, todos os objetos migrados são criados como releases com o número 1.0.

### Repositórios som controle de versão

Se um repositório, editado como o ModusOne Versão 5.0.4, for transferido para um repositório ModusOne Versão 5.1., todos os objetos transferidos recebem o número de versão 1.0.

Se o controle de versão não estiver ativo para o ModusOne Studio, todos os objetos recebem o número de versão 1.0.

### Renomeando objetos com versão




Renomear um objeto não afeta o seu número de versão. Todas as versões e referências usam o novo nome do objeto.

## Criar um trabalho de transferência

### Para criar um trabalho de transferência, execute estas etapas:

1. Selecione **Novo -> Trabalho de transferência** do menu de contexto do Navigator.
2. Digite o nome do trabalho de transferência na caixa de diálogo seguinte.






3. Vá para a guia Ferramentas de transferência, clique no botão **Adicionar**  .  
Como alternativa, adicione objetos arrastando e soltando do Navigator.
4. Selecione o objeto que possa conter objetos filhos que ainda não foram adicionados ao trabalho de transferência.  
  
Vá para a guia Ferramentas de transferência, clique no botão **Mostrar objetos obrigatórios**  .
5. Na caixa de diálogo seguinte, selecione os objetos que precisam ser adicionados ao trabalho de transferência.
6. Selecione os sistemas alvo necessários da janela do lado direito.
7. Se os objetos tiverem que ser sinalizados como protegidos contra gravação no sistema alvo, especifique isso na coluna **Definir proteção contra gravação**.
8. Inicie a transferência com o botão **Executar**  .  
Se o controle de versão ainda não estiver licenciado, os objetos existentes no sistema alvo serão substituídos.

## Exportar objetos

Se objetos individuais precisarem ser transferidos para outro servidor, eles podem ser exportados para um sistema de arquivos e importados de volta para o servidor de destino.

### Para exportar objetos, execute estas etapas:

1. Selecione **Novo -> Trabalho de transferência** do menu de contexto do Navigator.
2. Digite o nome do trabalho de transferência na caixa de diálogo seguinte.
3. Vá para a aba Ferramentas de transferência e clique no botão **Adicionar**  .  
Como alternativa, adicione objetos arrastando e soltando do Navigator.
4. Selecione o objeto que possa conter objetos filhos que ainda não foram adicionados ao trabalho de transferência.  
  
Vá para a aba **Ferramentas de transferência** e clique no botão **Mostrar objetos obrigatórios**  .
5. Na caixa de diálogo seguinte, selecione os objetos que precisam ser adicionados ao trabalho de transferência.
6. Selecione os sistemas de destino necessários da janela do lado direito.
7. Inicie a transferência com o botão **Exportar**  .
8. Digite o nome do subdiretório para o qual os objetos devem ser exportados na caixa de diálogo seguinte.  
Esse diretório é criado diretamente abaixo do diretório de trabalho RWS.  
Seu nome é uma combinação do computador e nome de usuário, além da data e hora.

Informações sobre a importação de sistemas podem ser encontradas no capítulo sobre [Importação de repositório](#).

# Ajudas de valor

## Introdução

Ajudas de valor auxiliam os usuários durante a entrada de variáveis manuais

Quatro tipos diferentes de ajudas de valor estão disponíveis:

[Caixa de texto com máscara de entrada](#)

[Caixa de texto com opções](#)

[Calendário pop-up](#)

[Campo de entrada numérico](#)

## Caixa de texto com máscara de entrada

Essa ajuda de valor permite que você especifique uma máscara de entrada para definir como inserir variáveis manuais.

**Execute as etapas a seguir para criar uma nova caixa de texto com máscara de entrada:**

1. Selecione **Novo -> Ajuda de valor** no menu de contexto do Navigator.
2. Insira o nome da ajuda de valor na caixa de diálogo seguinte.
3. Na caixa de diálogo seguinte, selecione o tipo **Caixa de texto com máscara de entrada**.
4. No campo **Máscara de entrada**, crie a máscara de entrada usando os **Caracteres de formatação** necessários.

Você pode usar o campo da caixa de diálogo **Teste de máscara** para testar como a máscara de entrada afeta as variáveis inseridas.

Caractere	Significado na máscara de entrada
!	Especifica que os caracteres opcionais, como os espaços à esquerda, são exibidos. Se '!' não for especificado, caracteres opcionais são exibidos como espaços à direita na máscara.
>	Converte todos os seguintes caracteres em letras maiúsculas até que o fim da máscara seja alcançado ou o caractere '<' seja detectado.
<	Converte todos os seguintes caracteres em letras maiúsculas até que o fim da máscara seja alcançado ou o caractere '>' seja detectado.
<>	Se os dois sinais forem usados juntos, nenhuma verificação de diferenciação de maiúsculas e minúsculas é feita. Os dados inseridos são formatado exatamente da mesma maneira que o usuário o inseriu.
\	O caractere após „\“ deve ser usado como um espaço reservado na máscara de entrada.

L l	Somente letras são entradas válidas (A-Z, a-z). Não é obrigatória uma entrada.
A a	Somente caracteres alfanuméricos são válidos (A-Z, a-z, 0-9). Não é obrigatória uma entrada.
C c	Qualquer caractere é permitido. Não é obrigatória uma entrada.
0 9	Somente um caractere numérico é permitido. Não é obrigatória uma entrada.
#	Somente um caractere numérico ou sinal de mais ou menos é permitido. Não é obrigatória uma entrada.
:	Uma vírgula é usada para separar horas, minutos e segundos na definição de hora. Se um caractere diferente for especificado nas configurações regionais do Windows, será exibido em vez da vírgula.
/	Uma barra é usada para separar as entradas de data, dia, mês e ano umas das outras. Se um caractere diferente for especificado nas configurações regionais do Windows, será exibido em vez da vírgula.
;	Um ponto e vírgula é usado para separar os parâmetros das três máscaras uns dos outros.
_	O caractere sublinhado _ insere automaticamente um espaço no texto. Os cursores ignoram esses caracteres quando um dado é inserido.

- Qualquer caractere não listado na tabela acima pode ser usado como o primeiro parâmetro da máscara como espaços reservados para entradas.
- Esses caracteres são inseridos automaticamente e ignorados pelo cursor na entrada de um dado.
- Os caracteres de máscara especiais listados na tabela acima também podem ser usados como espaços reservados, desde de que o caractere „\" seja especificado primeiro.
- Insira uma descrição opcional no campo Descrição.
- Vá para a guia **Ajuda de valor** e clique no botão **Ajuda de valor de teste**.  
Teste a máscara de entrada na caixa de diálogo seguinte.

## Caixa de texto com opções

Esse tipo de ajuda de valor permite definir uma lista de valores que são exibidos em um menu suspenso durante a entrada manual de variáveis.

**Execute as etapas a seguir para criar uma nova caixa de texto suspensa:**

1. Selecione **Novo -> Ajuda de valor** no menu de contexto do Navigator.
2. Insira o nome da ajuda de valor na caixa de diálogo seguinte.
3. Na caixa de diálogo seguinte, selecione o tipo **Caixa de texto com lista de seleção**.
4. Insira a lista de valores desejados no campo **Entradas na lista de seleção**, cada valor em uma linha separada.  
Também é possível definir os valores no formato **[ModusUser.AttributeName]**.
5. Ative a opção **Lista de classificação** se quiser exibir a lista em ordem alfabética.
6. Insira uma descrição opcional no campo Descrição.
7. Vá para a guia **Ajuda de valor** e clique no botão **Ajuda de valor de teste**.  
Teste a máscara de entrada na caixa de diálogo seguinte.

## Calendário pop-up

Uma ajuda de valor que fornece um calendário como uma função seletora de datas.

**Conclua as seguintes etapas para criar um Calendário pop-up:**

1. Selecione Novo -> Ajuda de valor no menu de contexto do Navigator.
2. Insira o nome da ajuda de valor na caixa de diálogo seguinte.
3. Na caixa de diálogo seguinte, selecione o tipo Calendário pop-up.
4. No menu suspenso Formatos pré-definidos selecione o formato de data obrigatório:

**Longo**

Data no formato Segunda, 5 de novembro de 2011

**Curto**

Data no formato 05.11.2011

**Personalizado**

Definido pelo formato de data do usuário. Insira o formato de data desejado no campo Formato auto-definido.

**As seguintes definições são permitidas:**

Definição	Resultado
dd	Dia, dois dígitos numéricos
ddd	Dia da semana (abreviado)
dddd	Dia da semana (completo)
MM	Mês, dois dígitos numéricos
MMM	Dia do mês (abreviado)

MMMM	Dia do mês (completo)
aa	Ano, dois dígitos numéricos
aaaa	Ano, quatro dígitos numéricos
hh	Horas
mm	Minutos
ss	Segundos

**Exemplo**

dddd, dd. MMMM aaaa hh:mm:ss

**Resulta em:**

Segunda, 05. Novembro de 2011 10:58:59

1. Insira uma descrição opcional no campo Descrição.
2. •Vá para a guia **Ajuda de valor** e clique no botão **Ajuda de valor de teste**.  
Teste a máscara de entrada na caixa de diálogo seguinte.

## Campo de entrada numérico

Este objeto de ajuda de valor permite que você defina intervalos de valores de uma variável numérica.

### Conclua as seguintes etapas para criar um campo de entrada numérico:

1. Selecione **Novo -> Ajuda de valor** no menu de contexto do Navigator.
2. Insira o nome da ajuda de valor na caixa de diálogo seguinte.
3. Na caixa de diálogo seguinte, selecione o tipo **Campo de entrada numérico**.
4. Defina as configurações necessárias como:

**Valor mínimo**

Especifica o valor mínimo que pode ser inserido para a variável.

**Valor máximo**

Especifica o valor máximo que pode ser inserido para a variável.

**Número de casas decimais**

Especifica o número de casas decimais.

**Incremento**

Especifica o valor das etapas do incremento individual quando a seta é pressionada.




5. Insira uma descrição opcional no campo Descrição.
6. Vá para a guia **Ajuda de valor** e clique no botão **Ajuda de valor de teste**.  
Teste a máscara de entrada na caixa de diálogo seguinte.

## Interface do usuário do Word

As interfaces do usuário do Word podem ser usadas para restringir as funções de edição ao concluir documento no Word.

Para isso, a Interface do usuário do Word é atribuída a um documento em um pacote.

### Execute as etapas a seguir para criar uma Interface do usuário do Word:

1. Selecione **Novo -> Interface do usuário do Word** no menu de contexto do Navigator.
2. Insira o nome da Interface do usuário do Word.
3. Insira um modelo usando o botão **Carregar modelo XML**  na guia **Ferramentas de CustomUI**.
4. Edite o modelo de acordo com seus requisitos específicos.  
Para obter mais informações, consulte a documentação da Microsoft.
5. Formate o código com o botão **Formatar XML** .
6. Verifique o código com o botão **Verificar XML** .

### Execute as etapas a seguir para atribuir uma Interface do usuário do Word a um documento:

1. Abra o pacote que contém o respectivo documento.
2. Selecione o documento no editor de pacotes.
3. No Object Inspector, clique na propriedade **CustomUI Reference** e no botão com três pontos ao lado da propriedade.
4. Selecione a Interface do usuário do Word na caixa de diálogo seguinte.

# Processos

## Introdução

O objeto **Processo** permite que você faça representações gráficas simples de sequências de processos.

O designer processo fornece uma interface de usuário gráfica para criar, salvar e editar modelos de seus processos de trabalho.

É feita uma distinção entre **Processos normais**, que controlam processos e sequências dentro do ModusOne, e **Processos MWS**, que controlam processos e processamento no ModusOne Client.

## Criar um processo

### Criar um processo

**Para criar um processo, execute estas etapas:**

- Selecione **Novo -> Processo** do menu de contexto do Navigator.
- Digite o nome do processo na caixa de diálogo seguinte.
- Defina se deseja criar um projeto padrão ou um projeto MWS.
- Se necessário, modifique as propriedades do processo no object inspector.
- Arraste os [componentes](#) obrigatórios na janela do editor de processo.
- Arraste os componentes do designer de processo obrigatórios na janela do editor de processo.
- [Posicione e conecte](#) os componentes do designer de processo conforme necessário.
- Compile o processo.

**Consulte também:**

[Ferramentas do designer de processo](#)

## Ferramentas do designer de processo

A barra de ferramentas oferece várias funções para processos de edição.

**Alinhar** 

Ajusta as posições horizontal e vertical dos blocos.

**Recortar** 

Exclui os objetos selecionados e os copia para a área de transferência.

**Copiar** 

Copia o objeto selecionado.

**Inserir** 

Insere um objeto a partir da área de transferência.

**Desfazer** 

Desfaz a última ação.

**Repetir** 

Repete a última ação.

**Compilar** 

Compila o processo.

Mensagens de êxito ou falha são registradas na janela de saída.

**Executar** 

Executa o processo.

**Parar** 

Para o processo.

**Pausar** 

Pausa o processo.

**Cancelar** 

Cancela o processo.

**Mais zoom** 

Aumenta o zoom da janela do editor.

**Menos zoom** 

Diminui o zoom da janela do editor.



**Tamanho normal**

Retorna a visualização da janela do editor ao tamanho normal.

**Ajustar**

Ajusta o tamanho da visualização da janela do editor para tornar todos os blocos do processo visíveis.

**Propriedades de processos padrão**

Parâmetro	Significado
Description	Use esta opção para inserir uma descrição em texto livre do processo.
Context	O contexto de processos normais é sempre <b>Process</b> e não pode ser alterado.
OnError	Especifique aqui um script que será executado se um erro de processamento ocorrer. Contexto: <a href="#">OnProcessError</a>
ObjectName	Nome do processo.

**Propriedades de um processo MWS**

Parâmetro	Significado
Description	Use esta opção para inserir uma descrição em texto livre do processo.
Context	O contexto de um processo MWS é sempre <b>MwsProcess</b> e não pode ser alterado.
OnGetInfoScript	O script atribuído aqui é chamado pelo método <code>Process_GetInfo</code> e define informações para atividades.
OnNewProcessScript	O script atribuído aqui é executado quando um novo processo é criado.
OnProcessEndScript	O script é executado quando um processo é encerrado usando <code>Process_Close</code> .
OnSetOptionsScript	O script atribuído aqui é executado quando são definidas opções de XML para atividades. Ele sempre se refere a informações externas que não têm nada a ver com o processo MWS em si.

OnStatusChangeScript	O script atribuído aqui é executado quando a estrutura do componente de negócios é alterada.
ProcessRootDirectory	É usado para especificar o diretório no qual os dados do processo MWS são armazenados.  A especificação também pode conter variáveis, por exemplo, "\$server\processes".  O diretório temporário é usado se essa propriedade não for definida.
OnError	Especifique aqui um script que será executado se um erro de processamento ocorrer.  Contexto: <a href="#">OnProcessError</a>
ObjectName	Nome do processo.

## Designer de processo - Componentes

### Áreas de um bloco de processo

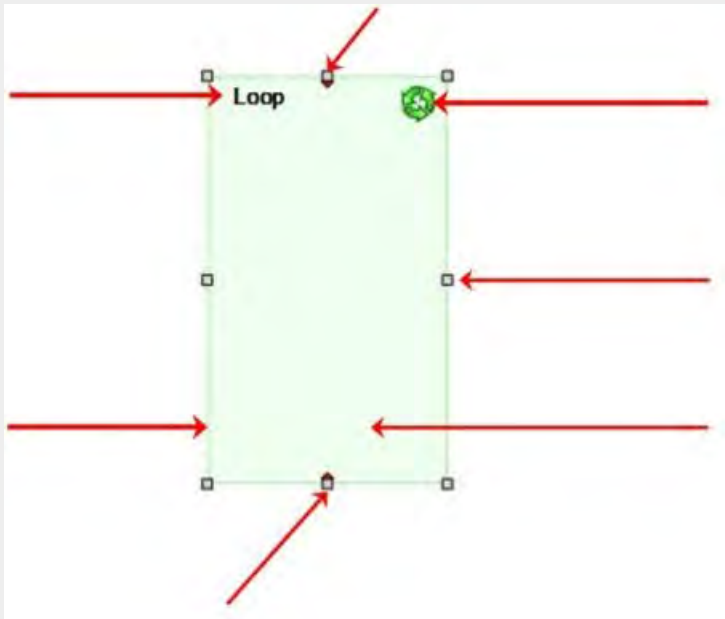
**Nome**

Bloco de nome de definição livre

Pode exibir automaticamente o nome do script anexo

**Porta**

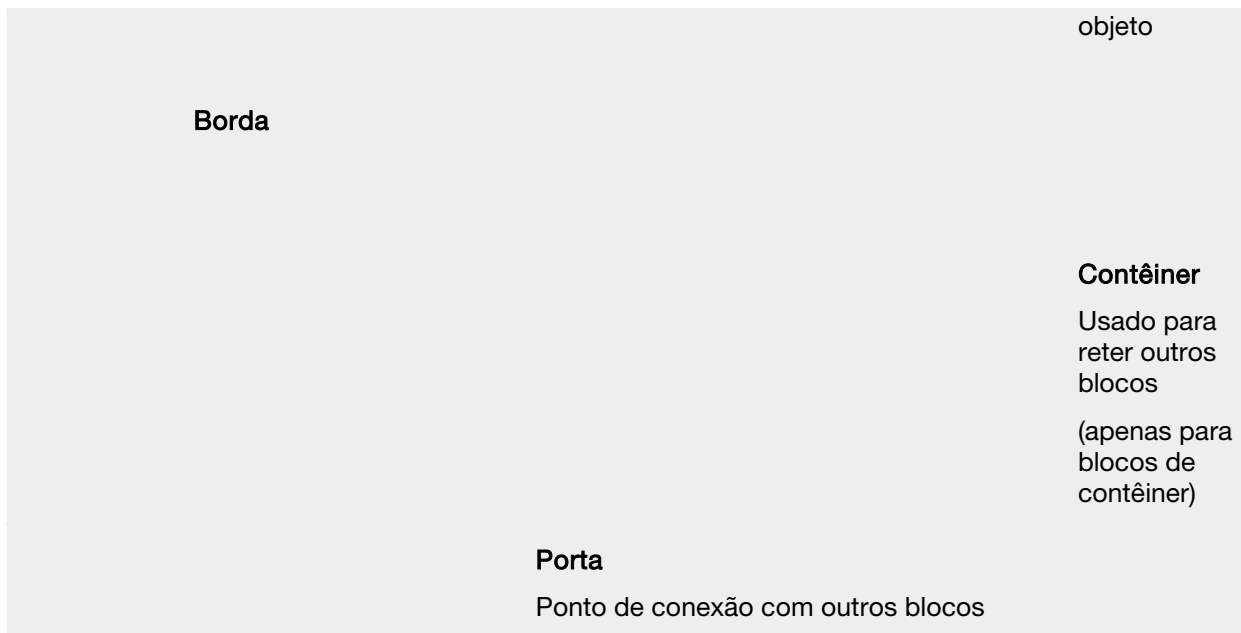
Ponto de conexão com outros blocos



**Ícone**

**Alça da superfície**

Permite que você altere o tamanho do



## Posicionar e conectar blocos de processo

[▶ Mostrar tudo](#)

- Blocos de processo são arrastados da caixa de ferramentas para a área de desenho por meio do recurso de arraste e solte.
- Os Scripts existentes também podem ser arrastados e soltos diretamente do Navigator para o designer de processo.
- É possível posicionar um bloco de processo quando ele é movido para outro bloco de processo. Isso é sinalizado pelo bloco de processo existente, por meio da mudança de cor do quadro.

### Exemplos

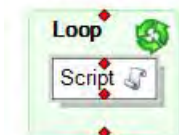
#### Adição de loop

Insira um loop arrastando-o da caixa de ferramentas do designer de processo e soltando-o em qualquer lugar na área de desenho.



#### Inserção de duas execuções de script em um loop

- Insira uma execução de script arrastando-o da caixa de ferramentas do designer do processo e soltando-o em qualquer lugar no loop já posicionado.



- Posicione a execução de script exatamente na área verde-claro no loop. Ao acessar o loop, a borda terá sua aparência alterada para traços vermelhos. Essa alteração nos sinais de aparência que o loop aceita o script como um antecessor.

### Observação

Antecessor e sucessor significa que o script está no mesmo nível de hierarquia que o loop.

Um antecessor é sinalizado por uma seta na direção da borda inferior do script para a borda superior do loop.

Um sucessor é sinalizado por uma seta na direção da borda inferior do loop para a borda superior do script.

Se o loop estiver pronto para aceitar o script, sua borda é revertida para sua aparência normal.

O cursor do mouse agora está diretamente sobre o loop e seus símbolos são alterados para um sinal de adição.

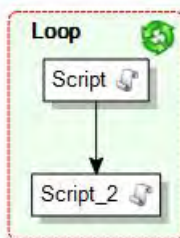
- Agora, arraste o segundo script no loop
- Dependendo se o script estiver posicionado antes ou depois do primeiro script, mova-o para a borda superior ou para a borda inferior do script. É possível soltar o script quando a borda do primeiro script altera para traços vermelhos.



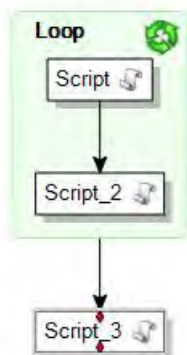
### Inserção de um script como sucessor do loop

Arraste uma execução de script da caixa de ferramentas do designer de processo e mova-a para a borda inferior do loop até a borda alterar sua aparência para traços vermelhos.

O loop está pronto para aceitar o novo script como um sucessor.



O script é vinculado ao loop:

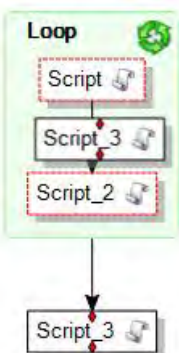


### Inserção de um script entre dois scripts já vinculados

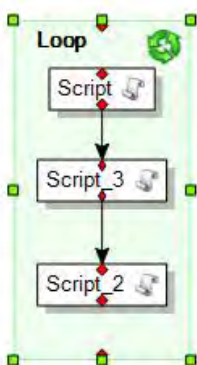
Agora, o Script\_3 será movido da sua posição lógica como sucessor do loop e inserido entre os scripts 1 e 2.

Para fazer isso, capture o Script\_3 pressionando a tecla CTRL e arrastando-o para a posição entre os dois scripts

As bordas dos dois scripts têm suas aparências alteradas para traços vermelhos:



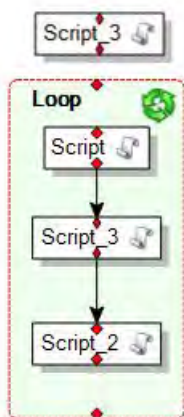
Agora, o terceiro script pode ser solto:



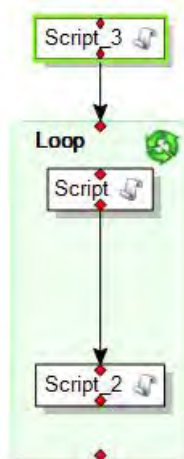
### Movimento de um script

Agora, o Script\_3 deve ser usado como um script antecessor e deixar sua posição lógica entre o Script\_1 e o Script\_2.

Para fazer isso, capture o Script\_3 pressionando a tecla CTRL e arrastando-o da borda superior do loop até a borda alterar para traços vermelhos.



Agora, o Script\_3 pode ser solto e posicionado antes do loop:

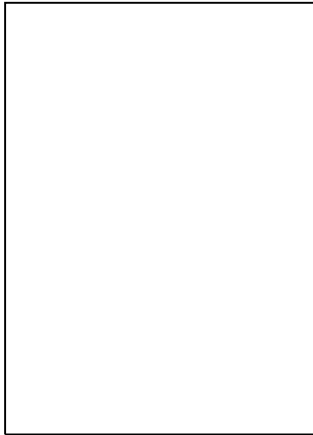


### Inserção de um script entre um script e um loop

Agora, o Script\_2 está prestes a ser inserido entre o Script\_3 e o loop.

Para fazer isso, capture o Script\_2 pressionando a tecla CTRL e arrastando-o para a posição entre o loop e o Script\_3.

As bordas desses dois objetos alteram para traços vermelhos, sinalizando que estão prontos para que o Script\_2 seja inserido.



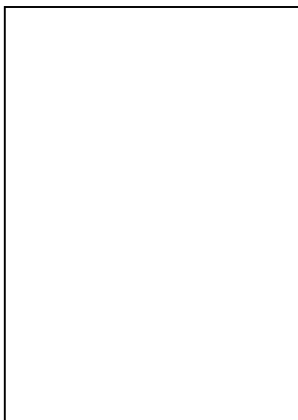
### Inserção de um script de volta no loop

Agora, o Script\_2 está prestes a ser inserido de volta no loop antes do Script.

Para fazer isso, capture o Script\_2 pressionando a tecla CTRL e arrastando-o pela borda superior do Script dentro do loop.

A borda do Script altera sua aparência para traços vermelhos, sinalizando que está pronto para aceitar o Script\_2 com um antecessor.

Após o Script\_2 ser solto, ele se torna o antecessor do Script dentro do loop.



Uma conexão direta entre o Script\_3 antes do loop foi estabelecida.

### Verificar

Um objeto de verificação permite que você defina diversas condições e verifique-as sequencialmente.

**Complete as etapas a seguir para inserir um bloco de Verificação em um processo:**

1. Abra o processo.
2. Clique em **Componentes do designer de processo** na [Caixa de ferramentas](#).
3. Arraste e solte o bloco de Verificação na janela do editor.
4. Se necessário, edite as propriedades do bloco de Verificação no Object Inspector.

### Tipo de verificação da propriedade

Propriedade	Definição
First	Uma das condições definidas deve ser verdadeira para continuar o processamento.
Any	Qualquer condição definida deve ser verdadeira para continuar o processamento.

### Condição

As condições são agrupadas em um objeto de verificação.

Isso define a ordem na qual as condições são verificadas.

#### Complete as etapas a seguir para inserir um bloco de Condições em um processo:

1. Abra o processo.
2. Clique em **Componentes do designer de processo** na [Caixa de ferramentas](#).
3. Arraste e solte o bloco de Condições na janela do editor.
4. No Object Inspector, clique na propriedade **Condição** e depois no botão com reticências ao lado da propriedade.
5. Selecione a condição na janela de diálogo seguinte.
6. Se necessário, edite as propriedades do bloco de Condições no Object Inspector.

#### Propriedade do tipo de condição

##### Complex

Você deve atribuir uma condição existente que retorne o resultado True ou False.

O tipo de condição corresponde a uma IF-query.

##### WhenNone

Escolha esse tipo de condição se nenhuma das condições do tipo complexa no objeto de verificação se aplique e você precise verificar o objeto que reaja com isso.

Depois, você pode atribuir um script a uma condição para passar uma mensagem de erro.

##### WhenAny

Escolha esse tipo de condição se uma das condições do tipo complexa no objeto de verificação se aplique e você precise verificar o objeto que reaja com isso.

##### WhenAll

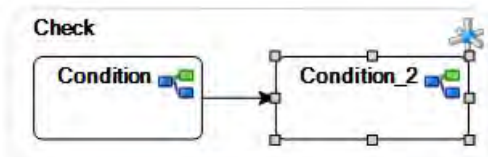
Escolha esse tipo de condição se todas as condições do tipo complexa no objeto de verificação se apliquem e você precise verificar o objeto que reaja com isso.

#### Organização de condições em sucessão

1. Arraste um novo objeto de condição na borda da condição para que a borda mude de aparência, para um tracejado vermelho, e um pequeno sinal de adição seja exibido.

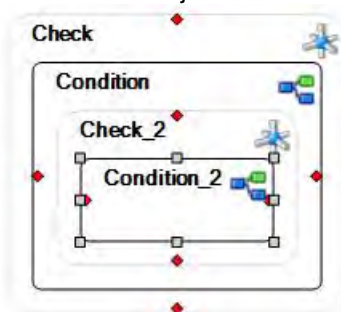


2. Arraste o objeto e ele será posicionado após a condição já inserida:

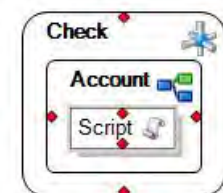


### Condições de aninhamento

1. Arraste o novo objeto de condição dentro da borda da condição - a borda permanece uma linha preta e um pequeno sinal de adição não é exibido.
2. Arraste o objeto e ele será posicionado dentro da condição já inserida:



you can place other objects in the condition that will be processed, depending on the result of the own condition.



### IF-Check

An IF-Check is composed of a Verification object that contains a Condition object.

**To insert an IF-Check block in a process, follow these steps:**

1. Open the process.
2. Click on **Components of the process designer** in the [Toolbox](#).
3. Drag and drop the IF-Check block into the editor window.
4. If necessary, edit the properties of the IF-Check block in the Object Inspector.
5. Insert additional objects in the condition as necessary.  
They will be executed if the condition is valid.

### IF-ELSE-Check

An IF-ELSE-Check is composed of a Verification object containing two Condition objects.

- A primeira condição é definida por padrão para o tipo de condição **Complex** e corresponde à If-query.
- A segunda condição é definida por padrão para o tipo de condição **WhenNone** e corresponde à Else-query.

**Para inserir um bloco IF-ELSE-Check em um processo, siga estas etapas:**

1. Abra o processo.
2. Clique em **Componentes do designer de processo** na [Caixa de ferramentas](#).
3. Arraste e solte o bloco IF-ELSE-Check na janela do editor.
4. Se necessário, edite as propriedades do bloco IF-ELSE-Check no Object Inspector.
5. Insira objetos adicionais na condição conforme necessário. Eles serão executados se a condição for válida.

## Comentário

Qualquer número de comentários pode ser adicionado a um processo.

**Complete as etapas a seguir para inserir um bloco de comentários em um processo:**

1. Abra o processo.
2. Clique em **Componentes do designer de processo** na [Caixa de ferramentas](#).
3. Arraste e solte o bloco de comentários na janela do editor.
4. Edite a propriedade do comentário do bloco de comentário no Object Inspector.

## Loop

Executar um loop permite a repetição de uma sequência de etapas até o ponto em que a condição não é mais válida.

**Conclua as seguintes etapas para inserir um bloqueio de Loop em um processo:**

1. Abra o processo.
2. Clique em **Componentes do designer de processo** na [Caixa de ferramentas](#).
3. Arraste e solte o bloqueio de Loop na janela do editor.
4. No Object Inspector, clique na propriedade **Condição** e depois no botão com reticências ao lado da propriedade.
5. Selecione a condição.
6. Insira objetos adicionais no Loop necessário.

## Execução de script

Este bloco funciona como um espaço reservado para um script e possibilita o início de um script a partir de um processo.

**Execute as etapas a seguir para inserir um bloco Script em um processo:**

1. Abra o processo.

2. Clique em **Componentes do designer de processo** na [Caixa de ferramentas](#).
3. Arraste e solte o bloco do script para a janela do editor.
4. No Object Inspector, clique na propriedade **Script** e no botão com três pontos ao lado da propriedade.
5. Selecione o script necessário na caixa de diálogo seguinte.

**Observação:**

Se o nome do script ultrapassar determinado comprimento, ele será truncado para clareza.

Com a ajuda do menu de contexto, é possível exibir o nome completo.

## Temporizador universal

O temporizador permite que você execute processos em um intervalo de tempo especificado.

**Execute as etapas a seguir para inserir um bloco de temporizador universal em um processo:**

1. Abra o processo.
2. Clique em **Componentes do designer de processo** na [Caixa de ferramentas](#).
3. Arraste e solte o bloco do temporizador universal para a janela do editor.
4. No object inspector, clique com o botão direito do mouse sobre a propriedade **UniversalTimer** e em seguida no botão **Criar**.  
Em seguida, clique no botão com os três pontos ao lado da propriedade.
5. Na caixa de diálogo seguinte, especifique a hora de início e o intervalo de iteração do processo:
  - Data e hora de início
  - Repetir Sim/Não em horas, minutos, segundos ou milissegundos
  - Repetição em ciclos de longo prazo diariamente, semanalmente, mensalmente
6. No propriedade **Script**, você pode atribuir um script para ser executado quando o temporizador inicia.  
Selecione o contexto [ProcessTimer](#) para o script.
7. No propriedade **OnError**, você pode atribuir um script para ser executado no caso de erro.
8. Arraste e solte os blocos necessários para o temporizador.

## Componentes do designer de processo

### Geral

Componentes fornecem as propriedades e métodos necessários para que um processo execute as tarefas de processamento específicas para as quais ele foi projetado.

### Componentes do Modus

DocumentCollectionManager

### **Informações gerais**

Esse componente fornece os métodos e variáveis necessários para gerar documentos de um pacote e encaminhá-los para outros sistemas descendentes.

Use a propriedade **DocumentCollection** para atribuir o pacote.

### ***DocumentCollectionManager Properties***

Propriedade	Significado
BundleSystemOld	Lê ou define a OID do sistema do sistema do qual o pacote será carregado.
MissingObjects	Reporta quais objetos não foram inseridos durante a geração do documento, ainda que, de acordo com a lógica de composição, tivessem que ser inseridos.
InsertTextComponentIndicators	Define se os indicadores do bloco de texto serão inseridos.
InsertCustomUI	Especifica se um objeto CustomUI atribuído a um documento será inserido.
IsDocumentCollectionLoaded	Declara se um componente carregou um pacote.
DestinationDirectory	Especifica o diretório de destino da geração do documento.

### ***Métodos de DocumentCollectionManager***

#### **AssignData**

Método de atribuição de dados a uma DataReference

#### **Sintaxe**

AssignData(string dataId, string dataXml)

#### **Parâmetro**

Parâmetro	Significado
DataId	ID de dados
DataXML	XML

#### **Valor de retorno**

-

#### **CreateDocuments**

Executa a geração de documentos.

### Sintaxe

CreateDocuments()

### Parâmetro

-

### Valor de retorno

-

### DefinePrinterForBundleByName

Define a impressora e tempo de impressão para todas as referências do pacote.

### Sintaxe

DefinePrinterForBundleByName(string printerName, ModusSuiteObjectType printerType, OdinPrintTime printTime)

DefinePrinterForBundleByName(string printerName, ModusSuiteObjectType printerType, OdinPrintTime printTime, string formName)

### Parâmetro

Parâmetro	Significado
printerName	Nome da impressora
printerType	Tipo de impressora <b>Os tipos disponíveis são:</b> ModusSuiteObjectType.Printer ModusSuiteObjectType. EMailPrinter ModusSuiteObjectType. OutsourcingPrinter
printTime	Tempo de impressão <b>Os tipos disponíveis são:</b> OdinPrintTime.Direct OdinPrintTime.Shifted OdinPrintTime.Unknown
formName	Nome do formulário

### Valor de retorno

-

**GetDataReferenceEnumerator**

Retorna um enumerador contendo todas as TextDataReferences.

**Sintaxe**

GetDataReferenceEnumerator()

**Parâmetro**

-

**Valor de retorno**

Enumerador com as seguintes propriedades e métodos:

Parâmetro	Significado
ObjectName	Nome do objeto
IsAssigned	Os dados já estão atribuídos a uma referência de dados: True / False
IsActiveDataSelection	É uma seleção ativa (dados são recuperados ativamente): True / False
IsPassiveDataSelection	É uma seleção passiva (dados não são recuperados ativamente): True / False
Assign(string xml)	Atribui dados à referência de dados. <b>Parâmetro</b> xml: XML string <b>Valor de retorno</b> -
GetXml()	Retorna o XML de dados dessa referência de dados ou uma string vazia se nenhum dado foi atribuído.

**GetManualVariables**

Retorna uma string de XML contendo as variáveis manuais de todos os documentos originais.

**Sintaxe**

GetManualVariables()

**Parâmetro**

-

**Valor de retorno**

String de XML

### **GetSelectionParameters**

Retorna uma string de XML contendo os parâmetros de seleção de todas as referências de dados.

#### **Sintaxe**

GetSelectionParameters()

#### **Parâmetro**

-

#### **Valor de retorno**

String de XML

### **LoadByAlias**

Carrega um pacote a partir de um alias como definido em uma tabela de aliases.

#### **Sintaxe**

LoadByAlias(aliasTableName, alias)

#### **Parâmetro**

Parâmetro	Significado
AliasTableName	Nome da tabela de aliases
Alias	Identificador de alias

#### **Valor de retorno**

-

### **LoadById**

Carrega um pacote a partir de sua ID de objeto no mesmo sistema em que o processo reside.

#### **Sintaxe**

LoadById(string objectId)

#### **Parâmetro**

Parâmetro	Significado
objectId	ID do objeto

**Valor de retorno**

-

**LoadByName**

Carrega um pacote a partir do seu nome no mesmo sistema em que o processo reside.

**Sintaxe**

```
LoadByName(string objectName)
```

**Parâmetro**

Parâmetro	Significado
objectName	Nome do objeto

**Valor de retorno**

-

**PrintAndArchive**

Imprime e arquiva documentos ou os encaminha ao sistema de saída descendente.

**Sintaxe**

```
PrintAndArchive(string processId, string processType, string processTitle)
```

```
PrintAndArchive(string processId, string processType, string processTitle, BundlePrintSequence bundlePrintSequence)
```

**Parâmetro**

Parâmetro	Significado	
processId	ID do processo	
processType	Tipo do processo	
processTitle	Título do processo	
bundlePrintSequence	<b>Valores possíveis:</b> BundlePrintSequence.BundleSequence  BundlePrintSequence.OriginalsFirst	Configuração padrão se o parâmetro não estiver definido.  Originais e cópias são passadas para o Odin na mesma ordem em que são posicionados no pacote.  O primeiro passa todos os originais para o Odin e, em seguida, as



		cópias.
--	--	---------

**Valor de retorno**

ModusProcessInfo XML (ImportXml for Odin) como string de XML

**ProcessData**

Executa o script AfterDataSelection.

**Sintaxe**

ProcessData()

**Parâmetro**

-

**Valor de retorno**

-

**ResetDocuments**

Redefine o status de todos os documentos originais para **não inserido**.

**Sintaxe**

ResetDocuments()

**Parâmetro**

-

**Valor de retorno**

-

**SetManualVariables**

Define variáveis manuais usando um string de XML.

**Sintaxe**

SetManualVariables(string manualVariables)

**Parâmetro**

Parâmetro	Significado
ManualVariables	String de XML

**Valor de retorno**

-

**SetSelectionParameters**

Define os parâmetros de seleção usando uma string de XML.

**Sintaxe**

SetSelectionParameters(string selectionParameters)

**Parâmetro**

Parâmetro	Significado
selectionParameters	String de XML

**Valor de retorno**

-

**Script de exemplo**

```
batchManager = ModusProcess.GetComponent("ModusBatchManager")
DocumentCollection = ModusProcess.GetComponent("DocumentCollectionManager")
;
;Load Bundle by specifying the name
bundleName = batchManager.DocumentCollection
documentCollection.BundleSystemOID = batchManager.SystemOID
documentCollection.LoadByName(bundleName)
;
documentCollection.DestinationDirectory = batchManager.Directory
```

**ModusBatchManager****Informações gerais**

Esse componente fornece os métodos e variáveis necessários para gerar documentos em processos em lote.

**Propriedades de ModusBatchManager**

Propriedade	Significado
BeginJobScript	Inicia o script de preparação da criação do documento.
DB Alias	Define o alias de banco de dados do banco de dados virtual dentro do componente.
EndJobScript	Especifica um script para terminar o processamento da criação de

	documento.
ProcessRootDirectory	Especifica o diretório no qual os pacotes gerados serão salvos.

## ***Métodos de ModusBatchManager***

### **DoWork**

Executa o script de início BatchManager.

### **Sintaxe**

DoWork()

DoWork(string xml)

### **Parâmetro**

Parâmetro	Significado
xml	Dados XML atribuídos ao contexto BatchManager.

### **Valor de retorno**

True: êxito

False: falha

### **Exemplo**

```
batchManager.DoWork( )
```

### **FinishWork**

Executa o script final BatchManager.

### **Sintaxe**

FinishWork()

FinishWork(string xml)

### **Parâmetro**

Parâmetro	Significado
xml	Dados XML atribuídos ao contexto BatchManager.

### **Valor de retorno**

-

**Exemplo**

```
batchManager.FinishWork()
```

**MwsAssistantManager****Informações gerais**

Esse componente exclui processos MWS concluídos.

**Propriedades do MwsAssistantManager**

Propriedade	Significado
Condition	Condição que define quais processos serão excluídos na forma de: MWS_STATUS=2  Nesse exemplo, todos os processos com o valor 2 no campo MWS_STATUS serão excluídos.
DB-Alias	Alias do banco de dados que permite o uso de um banco de dados virtual dentro desse componente.

**Métodos de MwsAssistantManager****DeleteFinishedProcesses**

Exclui os processos selecionados pela condição.

**Sintaxe**

```
DeleteFinishedProcesses()
```

**Parâmetro**

-

**Valor de retorno**

Número de processos excluídos.

**Exemplo**

```
deletedRecords = mwsAssistant.DeleteFinishedProcesses()
```

**Script de exemplo MWSAssistantManager**

```
mwsAssistant = ModusProcess.GetComponent("MWSAssistantManager")
deletedRecords = mwsAssistant.DeleteFinishedProcesses()
if (deletedRecords > 0)
    Protocol("==== MWSAssistant deleted {0} process(es) =====", 8,
    deletedRecords)
```

end-if

### ***MwsActivityStatus***

Esses componentes estão disponíveis apenas em processos MWS e são responsáveis pelo armazenamento de informações de atividades de processo (estados).

Um MwsActivityStatus é identificado por um nome de objeto exclusivo.

Scripts capazes de usar o [Context MwsProcess](#) podem acessar o MWSActivity com o comando **GetActivityStatus**.

## Componentes do Odin

### OdinArchive

#### ***Informações gerais***

Esse componente permite que você archive uma pilha.

O evento **OnArchive** tem acesso à pilha, envelope ou dados de trabalho.

Nesse script, por exemplo, o arquivo de apresentação pode ser convertido para um formato de arquivamento (por exemplo, Tiff ou PDF) usando o componente **OdinFileConvert** e, então, gravado no banco de dados.

#### ***Propriedades de OdinArchive***

Propriedade	Significado													
DB Alias	Define o banco de dados Odin no qual o componente irá executar suas ações.													
NextStackType	<p>Especifica o próximo tipo de pilha a ser definido após a execução com êxito e a chamada do método <b>End</b>.</p> <p>A lista suspensa exibe todos os tipos de pilha possíveis.</p> <p>Quaisquer tipos de pilha adicionais devem ser definidos no arquivo <b>ModusSuite.Common.PropertyEditors.dll.config</b>.</p> <p><b>Tipos de pilha padrão</b></p> <table border="1"> <tbody> <tr> <td>0</td> <td>Imprimir</td> <td>A pilha está pronta para ser impressa em uma impressora online ou de email (impressão online)</td> </tr> <tr> <td>10</td> <td>PortoOptimization</td> <td>A pilha está pronta para IT Franking (sem suporte no momento)</td> </tr> <tr> <td>20</td> <td>Streaming</td> <td>A pilha está pronta para fluxo.</td> </tr> <tr> <td>30</td> <td>Converter</td> <td>A pilha está pronta para ser convertida no formato de saída apropriado.</td> </tr> </tbody> </table>		0	Imprimir	A pilha está pronta para ser impressa em uma impressora online ou de email (impressão online)	10	PortoOptimization	A pilha está pronta para IT Franking (sem suporte no momento)	20	Streaming	A pilha está pronta para fluxo.	30	Converter	A pilha está pronta para ser convertida no formato de saída apropriado.
0	Imprimir	A pilha está pronta para ser impressa em uma impressora online ou de email (impressão online)												
10	PortoOptimization	A pilha está pronta para IT Franking (sem suporte no momento)												
20	Streaming	A pilha está pronta para fluxo.												
30	Converter	A pilha está pronta para ser convertida no formato de saída apropriado.												

	40	StreamPrint	A pilha (fluxo) está pronta para impressão em lote em uma impressora terceirizada.
	50	Arquivar	A pilha está pronta para arquivamento.
	99	Concluir	O processamento da pilha foi concluído, nenhuma ação adicional.
OnArchive	<p>O script especificado é executado durante a chamada do método 'ArchiveJob'.</p> <p>O script será executado somente quando o trabalho tiver que ser arquivado (ODJ_TO_ARCHIVE = "Y").</p> <p>O script deve verificar se o trabalho já foi arquivado (ODJ_ARCHIVED = "Y/N").</p> <p><b>Contexto disponível:</b></p> <p><a href="#">Contexto OdinArchive</a></p>		
StackSelect	<p>Seleção de SQL para identificar a pilha a ser processada.</p> <p>Ou insira a seleção de instrução diretamente ou consulte uma tabela de pesquisa (por exemplo, \$OdinSQL.PrintStackSelect\$).</p>		

### Eventos de OdinArchive

Evento	Significado
OnAfterStack	<p>Esse evento é executado quando o método <b>End()</b> é chamado após uma pilha ser processada.</p> <p>O conjunto de dados da pilha atual pode ser acessado na propriedade de contexto <b>CurrentStack</b>.</p> <p><b>Contexto disponível:</b></p> <p><a href="#">Contexto OdinArchive</a></p>
OnBeforeStack	<p>Esse evento é executado quando o método <b>Start()</b> é chamado antes de uma pilha ser processada.</p> <p>O conjunto de dados da pilha atual pode ser acessado na propriedade de contexto <b>CurrentStack</b>.</p> <p><b>Contexto disponível:</b></p> <p><a href="#">Contexto OdinArchive</a></p>

### Métodos de OdinArchive

#### ArchiveJob

Arquiva um determinado trabalho (de um envelope, se especificado).

O trabalho e o envelope são definidos no contexto `OdinArchiv` e, portanto, estão disponíveis para o evento `OnArchive`.

### Sintaxe

`ArchiveJob(job)`

`ArchiveJob(envelope, job)`

### Parâmetro

Parâmetro	Significado
envelope	O envelope a ser arquivado
job	O trabalho a ser arquivado

### Valor de retorno

-

### End

Encerra o arquivamento.

### Sintaxe

`End()`

### Parâmetro

-

### Valor de retorno

-

### GetCurrentStackRow

[▶ Mostrar tudo](#)

Após chamar `Start()` e antes de chamar `End()`, o método retorna um objeto cuja interface fornece acesso de leitura ao conjunto de dados atual da pilha.

### Sintaxe

`GetCurrentStackRow()`

### Parâmetro

-

### Valor de retorno

Interface de objeto que fornece acesso de leitura ao conjunto de dados atual da pilha.

### Os seguintes métodos estão disponíveis:

GetValue

Retorna o valor da coluna dada.

#### Sintaxe

GetValue(string columnName)

Parâmetro	Significado
columnName	Nome da coluna

#### Valor de retorno

Valor da coluna

Se o tipo de dados da coluna do banco de dados for DATE, Datetime ou qualquer tipo de dados válido, a data lida geralmente será retornada no formato MM/DD/AAAA hh:mm:ss.

#### Exemplo

```
GetObject("DateVar", "MLDate")
...
StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT")
;Converter a data no formato desejado com a ajuda do objeto MLDate
IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy hh:mm:ss", StringDateVar )
v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy hh:mm:ss ", IntDateVar)

GetFieldNameEnumerator
```

Retorna um enumerador.

#### Sintaxe

GetEnvelopeEnumerator()

#### Parâmetro

-

#### Valor de retorno

Enumerador com estes métodos:

Método	Função
Current	Retorna o objeto atual.
MoveNext	Move a posição até o próximo objeto a ser processado.
Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.

#### GetEnvelopeEnumerator



Retorna um enumerador de envelope. A função `GetValue()` recupera dados de envelope.

### Sintaxe

`GetEnvelopeEnumerator()`

### Parâmetro

-

### Valor de retorno

Enumerador com estes métodos:

Método	Significado
<code>Current</code>	Retorna o envelope atual.
<code>MoveNext()</code>	<p>Move a posição para o próximo envelope a ser processado.</p> <p><b>Valor de retorno:</b></p> <p>True: êxito</p> <p>False: o fim da coleta foi passado.</p> <p><b>Exemplo</b></p> <pre>... envEnumerator = OdinPrint.GetEnvelopeEnumerator() nextElement = envEnumerator.MoveNext() while (nextElement)     envelope = envEnumerator.Current; ... nextElement = envEnumerator.MoveNext()</pre>
<code>Reset</code>	Redefine o enumerador de volta para o início do primeiro elemento.

### GetJobEnumerator

Retorna um enumerador de trabalho.

A função `GetValue()` recupera dados de trabalho.

### Sintaxe

`GetJobEnumerator()`

`GetJobEnumerator(envelope)`

### Parâmetro

Parâmetro	Significado
envelope	O envelope cujos trabalhos serão acessados.

### Valor de retorno

Enumerador com estes métodos:

Método	Significado
Current	Retorna o trabalho atual.
MoveNext()	<p>Move a posição para o próximo trabalho a ser processado.</p> <p><b>Valor de retorno:</b></p> <p>True: êxito</p> <p>False: o fim da coleta foi passado.</p> <p><b>Exemplo</b></p> <pre>... jobEnumerator = OdinPrint.GetJobEnumerator() nextElement = jobEnumerator.MoveNext() while (nextElement)     job = jobEnumerator.Current; ... nextElement = jobEnumerator.MoveNext()</pre>
Reset	Redefine o enumerador de volta para o início do primeiro elemento.

### HasEnvelopes

Verifica se a pilha carregada pelo método **Start** contém envelopes.

### Sintaxe

HasEnvelopes

### Parâmetro

-

### Valor de retorno

True / False

### Start

Inicializa o arquivamento de uma pilha.

**Sintaxe**

Start(stackID)

**Parâmetro**

Parâmetro	Significado
stackId	ID de pilha

**Valor de retorno**

-

**Script de exemplo OdinArchive**

```

; If the in the OnArchive script the rendition file is to be converted by the
OdinFileConvert component
; the component must stored in the system pool
; SystemPool.FileConvert = ModusProcess.GetComponent("OdinFileConvert")
;
; Get archive components
OdinArchive = ModusProcess.GetComponent("OdinArchive")
;
; determine next stack
stackID = OdinArchive.GetNextStack()
;
if (stackID <> "")
;
; Yes, then initialize stack archiving
OdinArchive.Start(stackID)
; Does the stack have envelopes ?
if (OdinArchive.HasEnvelopes)
;
; Outsourcing Stack:
; Yes, get an envelope enumerator
envEnumerator = OdinArchive.GetEnvelopeEnumerator()
;
; run through all envelopes and archive jobs
nextEnv = envEnumerator.MoveNext()
while (nextEnv)
;
; get current envelope from enumerator
envelope = envEnumerator.Current
; as a test get envelope ID
envelopId = envelope.GetValue("ODE_ID")
;
; get an job enumerator for the envelope.

```

```

jobEnumerator = OdinArchive.GetJobEnumerator(envelope)
;
; run through all envelope jobs and archive
nextJob = jobEnumerator.MoveNext()
while (nextJob)
    ;
    ; get current job from enumerator
    job = jobEnumerator.Current
    ; as a test get job ID
    jobId = job.GetValue("ODJ_ID")
    ;
    ; and now archive, call the OnArchive script
    OdinArchive.ArchiveJob(envelope, job)
    ;
    ; process next Job
    nextJob = jobEnumerator.MoveNext()
    ;
end-while
;
; process next envelope
nextEnv = envEnumerator.MoveNext()
;
end-while
else
;
; Online Stack:
; get an job enumerator for the stack
jobEnumerator = OdinArchive.GetJobEnumerator()
;
; run through all stack jobs and archive
nextJob = jobEnumerator.MoveNext()
while (nextJob)
;
; get current job ID from enumerator
job = jobEnumerator.Current
; as a test get job ID
jobId = job.GetValue("ODJ_ID")
;
; and now archive, call the OnArchive script
OdinArchive.ArchiveJob(job)
;
; process next Job
nextJob = jobEnumerator.MoveNext()
;
end-while
end-if

```

```

;
; Finish archive
OdinArchive.End()
end-if

```

## OdinConvert

### Informações gerais

Use esse componente para converter um fluxo de impressão Odin para um formato de impressão (PDF, PCL, AFP...).

Todos o dados nos conjuntos de dados de trabalho são lidos no arquivo de descrição do fluxo, desde que o arquivo seja criado durante o fluxo.

Dessa maneira, os conjuntos de dados no banco de dados não são navegados durante a conversão, mas o invés disso, os dados obrigatórios são recuperados do arquivo de fluxo.

A lista da bandeja de saída no perfil Compart do respectivo formato de saída deve ser adaptado para mapear as 'bandejas virtuais' às bandejas físicas correspondentes.

### Exemplo de lista de bandeja de saída do perfil PCL 'mffpcl.pro':

A impressora de terceiros Odin tem 5 bandejas definidas, bandejas 1 - 3, além de firstbin e nextbin.

```

...
<output>
  <traylist>
    ...
    <!-- paper output bins -->
    <outputtray deviceid='15' name="Tray1" />
    <outputtray deviceid='259' name="Tray2"/>
    <outputtray deviceid='260' name="Tray3"/>
    <outputtray deviceid='15' name="firstbin"/>
    <outputtray deviceid='250' name="nextbin"/>
    ...
  </traylist>
</output>

```

### Propriedades de OdinConvert

[▶ Mostrar tudo](#)

Propriedade	Significado
ConvertOptions	<p>String separada por vírgula contendo os metadados e opções usadas durante a conversão.</p> <p>Essas opções também podem ser definidas no evento OnConvertStart para que os dados de pilha, por exemplo, possam ser usados para definir os metadados.</p> <p>Metadados</p>

TITLE	Título do documento.
SUBJECT	Assunto do documento.
AUTHOR	Autor do documento.
KEYWORDS	Palavras-chave do documento.

## Opções

TEXTNOTE_CHANGES	(Pdf 1.3/1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de segurança de PDF para Montagem de documento, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como verdadeiro/falso.
CHANGES	(Pdf 1.3/1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de segurança de PDF para Montagem de documento, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como verdadeiro/falso.
FILL	(Pdf 1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de segurança de PDF para Montagem de documento, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como verdadeiro/falso.
COPY	(Pdf 1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de segurança de PDF para Montagem de documento, Conteúdo copiado, Conteúdo copiado para acessibilidade, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como

	verdadeiro/falso.
DIGITAL_COPY	(Pdf 1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de segurança de PDF para Montagem de documento, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como verdadeiro/falso.
ASSEMBLE	(Pdf 1.4) TRUE/FALSE A montagem de um ou mais documentos não é permitida. Com a especificação deste parâmetro, os atributos de segurança de PDF para Montagem de documento, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como verdadeiro/falso.
ACCESSIBILITY	(Pdf 1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de segurança de PDF para Conteúdo copiado, Conteúdo copiado para acessibilidade, Extração de página serão definidos como verdadeiro/falso.
DUPLEX_ADDPAGE	TRUE: Esta opção especifica se uma página vazia deve ser adicionada a um documento duplex com um número ímpar de páginas. FALSE: Configuração padrão. Esta opção define o modo simplex para a última página de um documento duplex com um número ímpar de páginas.
DUPLEX_OPTIMIZE	TRUE/FALSE Quando verdadeiro, as configurações de duplex são definidas apenas no fluxo de saída, quando o modo duplex muda.
MAP_COLOR	TRUE/FALSE Este parâmetro é usado para definir se o mapeamento de cores da definição AFP-HighlightColor será executado ou não. As definições de cores individuais podem ser especificadas como MapColor_1 até MapColor_n.

	<p><b>Exemplo:</b></p> <pre>Map_Color=TRUE,"MapColor_1=-1,20,black","MapColor_2=0,20,red"</pre> <p><b>Observação:</b></p> <p>Como os parâmetros MapColor contêm valores separados por vírgulas, a string completa deve estar entre aspas.</p> <p><b>A estrutura é a seguinte:</b></p> <pre>MapColor_???= HSB-Color,Threshold,ColorName</pre> <p>HSB-Color=Ângulo da cor HSB</p> <p>Threshold= Valor limite da cor HSB</p> <p>ColorName= Nome da cor como definido no perfil AFP, em HighlightColorList.</p> <p><b>O exemplo acima requer estas definições no perfil AFP:</b></p> <pre>&lt;/highlightcolorlist&gt;   &lt;highlightcolor name='red' type='OCA' number='0x0002' foreground-rgb="#00FF0000" /&gt;   &lt;highlightcolor name='black' type='OCA' number='0x0008' foreground-rgb="#00000000" /&gt; &lt;/highlightcolorlist&gt;</pre> <p>e</p> <pre>&lt;optimizepa value="never" /&gt; &lt;color value="FALSE" /&gt; &lt;usehighlightcolor value="TRUE" /&gt; &lt;enablebilevelimage value="TRUE" /&gt;</pre>
SHOW_BOOKMARKS	<p>TRUE: Os bookmarks são mostrados quando o arquivo PDF é aberto.</p> <p>FALSE: Os bookmarks não são mostrados quando o arquivo PDF é aberto.</p>



	OVERLAYS	TRUE: São criadas OVERLAYS. FALSE: Padrão. Não são criadas OVERLAYS.	
	MERGEFONTS	TRUE: Padrão FALSE	
	MERGE_SPLITWORDS	Ativa e desativa a otimização da saída PCL. Defina esta opção como FALSE se você tiver problemas de espaçamento de caracteres em um PCL gerado. A configuração padrão é TRUE. TRUE: Padrão, a otimização da saída PCL é ativada. FALSE: A otimização da saída PCL é desativada.	
Alias de BD	Define o banco de dados Odin no qual o componente irá executar suas ações.		
ImagePath	Caminho principal para os arquivos de imagem que pode ser adicionado aos documentos com o comando <b>Imagem</b> . A declaração é opcional.		
LicenseFilename	Nome do arquivo de licença do Compart, por exemplo, cpsdk.lic.		
MasterPath	Caminho principal para os documentos / arquivos de spool. A declaração é opcional.		
NextStackType	Especifica o próximo tipo de pilha a ser definido após a execução com êxito e a chamada do método <b>End</b> . A lista suspensa exibe todos os tipos de pilha possíveis. Quaisquer tipos de pilha adicionais devem ser definidos no arquivo <b>ModusSuite.Common.PropertyEditors.dll.config</b> .  <b>Tipos de pilha padrão</b>		
	0	Imprimir	A pilha está pronta para ser impressa em uma impressora online ou de email (impressão online)
	10	PortoOptimization	A pilha está pronta para IT Franking (sem suporte no momento)
	20	Streaming	A pilha está pronta para fluxo.
	30	Converter	A pilha está pronta para ser convertida no formato de saída apropriado.

	40	StreamPrint	A pilha (fluxo) está pronta para impressão em lote em uma impressora terceirizada.
	50	Arquivar	A pilha está pronta para arquivamento.
	99	Concluir	O processamento da pilha foi concluído, nenhuma ação adicional.
OutputFileName	Retorna o nome atual do arquivo de saída. Essa propriedade é visível somente no componente e não no object inspector porque ele é inicializado chamando o método Start() no tempo de execução.		
OutputFormat	O formato de saída para o qual o fluxo é convertido. Corresponde à extensão de arquivo, ou seja, PCL, PS, AFP ou PDF.		
OutputPath	Caminho para o diretório de saída.		
ProfilePath	Diretório para os arquivos de perfil Compart.		
StackSelect	Seleção de SQL para identificar a pilha a ser processada. Ou insira a seleção de instrução diretamente ou consulte uma tabela de pesquisa (por exemplo, \$OdinSQL.PrintStackSelect\$).		
TraceFileName	Nome do arquivo trace Compart.		
Tracelevel	Especifica qual categoria de erro é escrita no arquivo trace.		

### **Eventos de OdinConvert**

Propriedade	Significado
OnConvertEnd	O script especificado é executado no final de um processo, enquanto chama o método <b>End</b> .
OnConvertStart	O script especificado é executado no início do processamento, enquanto chama o método <b>Start</b> . O script também pode ser usado para definir as propriedades do componente.
OnDocumentConvert	O script especificado é executado para cada documento.
OnEnvelopeEnd	O script especificado é executado no final do envelope.
OnEnvelopeStart	O script especificado é executado no início do envelope.
OnStackEnd	O script especificado é executado no final da pilha.

**OnStackStart**

O script especificado é executado no início da pilha.

## ***Métodos de OdinConvert***

### **ConvertDocument**

Converte um documento e grava-o para o fluxo de saída.

#### **Sintaxe**

ConvertDocument(doc)

#### **Parâmetro**

Parâmetro	Significado
doc	O documento a ser convertido.

#### **Valor de retorno**

-

#### **End**

Termina uma conversão e executa uma atualização na pilha. Um script atribuído ao evento 'OnConvertEnd' é executado aqui.

#### **Sintaxe**

End()

#### **Parâmetro**

-

#### **Valor de retorno**

-

### **GetCurrentStackRow**

[▶ Mostrar tudo](#)

Após chamar Start() e antes de chamar End(), o método retorna um objeto cuja interface fornece acesso de leitura ao conjunto de dados atual da pilha.

#### **Sintaxe**

GetCurrentStackRow()

#### **Parâmetro**

-

**Valor de retorno**

Interface de objeto que fornece acesso de leitura ao conjunto de dados atual da pilha.

**Os seguintes métodos estão disponíveis:**

GetValue

Retorna o valor da coluna dada.

**Sintaxe**

GetValue(string columnName)

Parâmetro	Significado
columnName	Nome da coluna

**Valor de retorno**

Valor da coluna

Se o tipo de dados da coluna do banco de dados for DATE, Datetime ou qualquer tipo de dados válido, a data lida geralmente será retornada no formato MM/DD/AAAA hh:mm:ss.

**Exemplo**

```
GetObject("DateVar", "MLDate")
...
StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT")
;Converter a data no formato desejado com a ajuda do objeto MLDate
IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy hh:mm:ss", StringDateVar )
v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy hh:mm:ss ", IntDateVar)
GetFieldNameEnumerator
```

Retorna um enumerador.

**Sintaxe**

GetEnvelopeEnumerator()

**Parâmetro**

-

**Valor de retorno**

Enumerador com estes métodos:

Método	Função
Current	Retorna o objeto atual.
MoveNext	Move a posição até o próximo objeto a ser processado.
Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.

**GetDocumentEnumerator**

Retorna um enumerador de documento.

**Sintaxe**

GetDocumentEnumerator()

**Parâmetro**

-

**Valor de retorno**

Enumerador com estes métodos:

Parâmetro	Significado
Current	Retorna o documento atual.
MoveNext	Movimenta a posição até o próximo documento a ser processado.
Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.

**GetNextStack**

Determina a próxima pilha a ser convertida, de acordo com o SQL correspondente e retorna sua ID.

**Sintaxe**

GetNextStack()

**Parâmetro**

-

**Valor de retorno**

ID de pilha

**SetStackError**

Define o status de uma pilha para erro.

**Sintaxe**

SetStackError(errorMessage)

**Parâmetro**

Parâmetro	Significado
-----------	-------------

errorMessage	Mensagem de erro
--------------	------------------

**Valor de retorno**

-

**Start**

Inicializa a pilha para conversão. Um script atribuído ao evento 'OnConvertStart' será executado aqui.

**Sintaxe**

Start(stackId)

**Parâmetro**

Parâmetro	Significado
stackId	ID de pilha

**Valor de retorno**

-

**Script de exemplo OdinConvert**

```

; get component for converting
OdinConvert = ModusProcess.GetComponent("OdinConvert")
;
; determine the next stack for streaming
stackId = OdinConvert.GetNextStack()
;
; was a stack found ?
if (stackID <> "")
;
    ; yes, now initialize stack / conversion.
    OdinConvert.Start(stackId)
    ;
    ;
; get a document enumerator
enumerator = OdinConvert.GetDocumentEnumerator()
;
; Are there more documents in the stream?
nextDoc = enumerator.MoveNext()
while (nextDoc = true)
;
    ; yes, now determine the document
    doc = enumerator.Current

```

```

;
; for tests determine the number of pages and name of the document
pageCount = doc.PageCount
fileName = doc.FileName
id = doc.Id
;
; and log
Protocol("id: {0} - pages: {1} - {2}", 0, id, pageCount, fileName)
;
; and write to output stream
OdinConvert.ConvertDocument(doc)
;
; Are there more documents in the stream?
nextDoc = enumerator.MoveNext()
;
end-while
; end converting, update stack
OdinConvert.End()
end-if

```

## OdinCustomProcess

### **Informações gerais**

Esse componente só está disponível quando o Odin foi licenciado corretamente.

Esse componente permite o processamento de pilha e funciona de maneira similar ao componente **OdinArchiv**.

A propriedade **StackType** define qual pilha será processada.

Você pode definir seus próprios tipos de pilha no arquivo

**ModusSuite.Common.PropertyEditors.dll.config** para que eles também apareçam na lista suspensa **StackType** e **NextStacktype**.

### **Sample ModusSuite.Common.PropertyEditors.dll.config:**

```

<PropertyEditor>
  <Odin>
    <StackTypes>
      <StackType value="60" name="Transfer" />
    </StackTypes>
    <OutputFormat>
      <Format value="pdf" name="Portable Document Format" />
      <Format value="pcl" name="Printer Command Language" />
      <Format value="pos" name="Postscript" />
      <Format value="afp" name="Advanced Function Presentation" />
      <Format value="tif" name="Tagged Image File Format" />
    </OutputFormat>
  </Odin>

```

&lt;/PropertyEditor&gt;

**Propriedades de OdinCustomProcess**

Propriedade	Significado																					
DB Alias	Define o banco de dados Odin no qual o componente irá executar suas ações.																					
NextStackType	<p>Especifica o próximo tipo de pilha a ser definido após a execução com êxito e a chamada do método <b>End</b>.</p> <p>A lista suspensa exibe todos os tipos de pilha possíveis.</p> <p>Quaisquer tipos de pilha adicionais devem ser definidos no arquivo <b>ModusSuite.Common.PropertyEditors.dll.config</b>.</p> <p><b>Tipos de pilha padrão</b></p> <table border="1"> <tbody> <tr> <td>0</td> <td>Imprimir</td> <td>A pilha está pronta para ser impressa em uma impressora online ou de email (impressão online)</td> </tr> <tr> <td>10</td> <td>PortoOptimization</td> <td>A pilha está pronta para IT Franking (sem suporte no momento)</td> </tr> <tr> <td>20</td> <td>Streaming</td> <td>A pilha está pronta para fluxo.</td> </tr> <tr> <td>30</td> <td>Converter</td> <td>A pilha está pronta para ser convertida no formato de saída apropriado.</td> </tr> <tr> <td>40</td> <td>StreamPrint</td> <td>A pilha (fluxo) está pronta para impressão em lote em uma impressora terceirizada.</td> </tr> <tr> <td>50</td> <td>Arquivar</td> <td>A pilha está pronta para arquivamento.</td> </tr> <tr> <td>99</td> <td>Concluir</td> <td>O processamento da pilha foi concluído, nenhuma ação adicional.</td> </tr> </tbody> </table>	0	Imprimir	A pilha está pronta para ser impressa em uma impressora online ou de email (impressão online)	10	PortoOptimization	A pilha está pronta para IT Franking (sem suporte no momento)	20	Streaming	A pilha está pronta para fluxo.	30	Converter	A pilha está pronta para ser convertida no formato de saída apropriado.	40	StreamPrint	A pilha (fluxo) está pronta para impressão em lote em uma impressora terceirizada.	50	Arquivar	A pilha está pronta para arquivamento.	99	Concluir	O processamento da pilha foi concluído, nenhuma ação adicional.
0	Imprimir	A pilha está pronta para ser impressa em uma impressora online ou de email (impressão online)																				
10	PortoOptimization	A pilha está pronta para IT Franking (sem suporte no momento)																				
20	Streaming	A pilha está pronta para fluxo.																				
30	Converter	A pilha está pronta para ser convertida no formato de saída apropriado.																				
40	StreamPrint	A pilha (fluxo) está pronta para impressão em lote em uma impressora terceirizada.																				
50	Arquivar	A pilha está pronta para arquivamento.																				
99	Concluir	O processamento da pilha foi concluído, nenhuma ação adicional.																				
StackSelect	<p>Seleção de SQL para identificar a pilha a ser processada.</p> <p>Ou insira a seleção de instrução diretamente ou consulte uma tabela de pesquisa (por exemplo, \$OdinSQL.PrintStackSelect\$).</p>																					
StackType	<p>Especifica o tipo de pilha, que será processado por este componente.</p> <p>Selecione o tipo de pilha definido anteriormente no processo personalizado.</p>																					

**Eventos de OdinCustomProcess**

Evento	Significado
--------	-------------



OnAfterStack	<p>O script especificado aqui é executado após uma pilha ter sido processada com êxito durante a chamada do método 'End'.</p> <p>O evento corresponde ao evento AfterStack de uma impressora no modus 4.</p> <p><b>Contexto disponível:</b>  <a href="#">Contexto OdinCustomProcess</a></p>
OnBeforeStack	<p>O script especificado aqui é executado durante a chamada do método Start(), ou seja, antes do processamento da pilha ser iniciado.</p> <p>O evento corresponde ao evento BeforeStack de uma impressora no modus 4.</p> <p><b>Contexto disponível:</b>  <a href="#">Contexto OdinCustomProcess</a></p>
OnProcessData	<p>O script especificado aqui é executado quando o método ProcessJob do componente é chamado, tornando os dados da pilha, do envelope ou do trabalho disponíveis.</p> <p><b>Contexto disponível:</b>  <a href="#">Contexto OdinCustomProcess</a></p>

## ***Métodos de OdinCustomProcess***

### **End**

Termina uma conversão e executa uma atualização na pilha. Um script atribuído ao evento 'OnConvertEnd' é executado aqui.

### **Sintaxe**

End()

### **Parâmetro**

-

### **Valor de retorno**

-

### **GetCurrentStackRow**

► [Mostrar tudo](#)

Após chamar Start() e antes de chamar End(), o método retorna um objeto cuja interface fornece acesso de leitura ao conjunto de dados atual da pilha.

### Sintaxe

GetCurrentStackRow()

### Parâmetro

-

### Valor de retorno

Interface de objeto que fornece acesso de leitura ao conjunto de dados atual da pilha.

### Os seguintes métodos estão disponíveis:

GetValue

Retorna o valor da coluna dada.

### Sintaxe

GetValue(string columnName)

Parâmetro	Significado
columnName	Nome da coluna

### Valor de retorno

Valor da coluna

Se o tipo de dados da coluna do banco de dados for DATE, Datetime ou qualquer tipo de dados válido, a data lida geralmente será retornada no formato MM/DD/AAAA hh:mm:ss.

### Exemplo

```
GetObject("DateVar", "MLDate")
```

...

```
StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT")
```

```
;Converter a data no formato desejado com a ajuda do objeto MLDate
```

```
IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy hh:mm:ss", StringDateVar )
```

```
v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy hh:mm:ss ", IntDateVar)
```

```
GetFieldNameEnumerator
```

Retorna um enumerador.

### Sintaxe

GetEnvelopeEnumerator()

### Parâmetro

-

### Valor de retorno

Enumerador com estes métodos:

Método	Função
Current	Retorna o objeto atual.
MoveNext	Move a posição até o próximo objeto a ser processado.
Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.

### GetEnvelopeEnumerator

Retorna um enumerador de envelope. A função GetValue() recupera dados de envelope.

#### Sintaxe

GetEnvelopeEnumerator()

#### Parâmetro

-

#### Valor de retorno

Enumerador com estes métodos:

Método	Significado
Current	Retorna o envelope atual.
MoveNext()	<p>Move a posição para o próximo envelope a ser processado.</p> <p><b>Valor de retorno:</b>            True: êxito            False: o fim da coleta foi passado.</p> <p><b>Exemplo</b></p> <pre> ... envEnumerator = OdinPrint.GetEnvelopeEnumerator() nextElement = envEnumerator.MoveNext() while (nextElement)     envelope = envEnumerator.Current; ... nextElement = envEnumerator.MoveNext() </pre>
Reset	Redefine o enumerador de volta para o início do primeiro elemento.

### GetNextStack

Determina a próxima pilha a ser convertida, de acordo com o SQL correspondente e retorna sua ID.

### Sintaxe

GetNextStack()

### Parâmetro

-

### Valor de retorno

ID de pilha

### GetJobEnumerator

Retorna um enumerador de trabalho.

A função GetValue() recupera dados de trabalho.

### Sintaxe

GetJobEnumerator()

GetJobEnumerator(envelope)

### Parâmetro

Parâmetro	Significado
envelope	O envelope cujos trabalhos serão acessados.

### Valor de retorno

Enumerador com estes métodos:

Método	Significado
Current	Retorna o trabalho atual.
MoveNext()	<p>Move a posição para o próximo trabalho a ser processado.</p> <p><b>Valor de retorno:</b></p> <p>True: êxito</p> <p>False: o fim da coleta foi passado.</p> <p><b>Exemplo</b></p> <pre>... jobEnumerator = OdinPrint.GetJobEnumerator() nextElement = jobEnumerator.MoveNext() while (nextElement)</pre>

	<pre> job = jobEnumerator.Current; ... nextElement = jobEnumerator.MoveNext(); </pre>
Reset	Redefine o enumerador de volta para o início do primeiro elemento.

**HasEnvelopes**

Verifica se a pilha carregada pelo método **Start** contém envelopes.

**Sintaxe**

HasEnvelopes

**Parâmetro**

-

**Valor de retorno**

True / False

**Process**

Processa o trabalho especificado.

O trabalho é definido no OdinCustomProcessContext e fica disponível para o evento OnProcessData.

**Sintaxe**

Process (job)

**Parâmetro**

Parâmetro	Significado
job	Trabalho

**Valor de retorno**

-

**ProcessJob**

Processa determinado trabalho do envelope especificado.

O trabalho e o envelope são definidos em OdinCustomProcessContext e portanto estão disponíveis para o evento OnProcessData.

**Sintaxe**

ProcessJob(envelope, job)

**Parâmetro**

Parâmetro	Significado
envelope	Envelope
job	Trabalho

### Valor de retorno

-

### ProcessJobs

Processa todos os trabalhos do envelope especificado.

O trabalho é definido no OdinCustomProcessContext e fica disponível para o evento OnProcessData.

### Sintaxe

ProcessJobs(envelope)

### Parâmetro

Parâmetro	Significado
envelope	Envelope

### Valor de retorno

-

### Start

Inicializa a pilha para conversão. Um script atribuído ao evento **OnConvertStart** será executado aqui.

### Sintaxe

Start(stackId)

### Parâmetro

Parâmetro	Significado
stackId	ID de pilha

### Valor de retorno

-

### Script de exemplo OdinCustomProcess

```
; Get CustomProcess component
OdinCustomProcess = ModusProcess.GetComponent("OdinCustomProcess")
```

```

;
; determine next stack
stackID = OdinCustomProcess.GetNextStack()
;
if (stackID <> "")
;
; Yes, then initialize stack processing,
; Execute the OnBeforeStack script.
OdinCustomProcess.Start(stackID)
;
; Does the stack have envelopes ?
if (OdinCustomProcess.HasEnvelopes)
;
; Outsourcing Stack:
; Yes, get an envelope enumerator
envEnumerator = OdinCustomProcess.GetEnvelopeEnumerator()
;
; run through all envelopes and process the jobs
nextEnv = envEnumerator.MoveNext()
while (nextEnv)
;
; get current envelope from enumerator
envelope = envEnumerator.Current
; as a test get envelope ID
envelopId = envelope.GetValue("ODE_ID")
;
; get an job enumerator for the envelope.
jobEnumerator = OdinCustomProcess.GetJobEnumerator(envelope)
;
; run through and process all envelope jobs
nextJob = jobEnumerator.MoveNext()
while (nextJob)
;
; get current job from enumerator
job = jobEnumerator.Current
; as a test get job ID
jobId = job.GetValue("ODJ_ID")
;
; Process job, call the OnProcessData script
OdinCustomProcess.ProcessJob(envelope, job)
;
; process next Job
nextJob = jobEnumerator.MoveNext()
;
end-while
;

```

```

        ; process next envelope
        nextEnv = envEnumerator.MoveNext()
        ;
    end-while
else
    ; Online Stack:
    ; get an job enumerator for the stack
    jobEnumerator = OdinCustomProcess.GetJobEnumerator()
    ;
    ; run through and process all stack jobs
    nextJob = jobEnumerator.MoveNext()
    while (nextJob)
        ; get current job from enumerator
        job = jobEnumerator.Current
        ; as a test get job ID
        jobId = job.GetValue("ODJ_ID")
        ;
        ; and process, the OnProcessData script is called
        OdinCustomProcess.ProcessJob(job)
        ;
        ; process next Job
        nextJob = jobEnumerator.MoveNext()
    end-while
end-if
; Stack processing finished, call the OnAfterStack script
OdinCustomProcess.End()
end-if

```

## OdinDbTasks

### **Informações gerais**

O componente OdinDbTasks fornece diversas funções para excluir e redefinir os conjuntos de dados Odin.

### **Propriedades de OdinDbTasks**

Propriedade	Significado
DB Alias	Define o banco de dados Odin no qual o componente irá executar suas ações.
OnDeleteJob	O script definido aqui é executado depois de OnDeleteStack para cada trabalho a ser excluído.
OnDeleteStack	O script definido aqui é executado antes de OnDeleteJob para cada pilha a ser excluída.



## Métodos de OdinDbTasks

### DeleteStack

Exclui uma pilha, incluindo todos os envelopes e trabalhos relacionados a ela. Os arquivos de documentos dos trabalhos também são excluídos. Se diretórios de importação vazios forem deixados após a exclusão dos arquivos de documentos, eles também serão excluídos.

### Sintaxe

DeleteStack(CustomStackSelect)

### Parâmetro

Parâmetro	Significado
customStackSelect	Instrução de seleção para encontrar as pilhas a serem excluídas. Se uma string vazia for definida, a instrução de seleção padrão é usada.

### Seleção padrão para um SQL Server:

```
Select TOP 1 ODS_ID FROM dbo.ODIN_STACK WITH (updlock, rowlock)
WHERE (ODS_STATE IN (2,4))
and (ODS_DELETE_AT <= GETDATE())
and (ODS_LOCKBY is NULL)
```

Seleção padrão para um banco de dados Oracle:

```
Select ODS_ID FROM ODIN_STACK
Where (ODS_STATE In (2,4))
and (ODS_DELETE_AT <= SYSDATE)
and (ODS_LOCKBY is NULL)
and (RowNum = 1) for update
```

### Valor de retorno

A ID da pilha excluída ou uma string vazia de nenhuma pilha foi excluída.

### Script de exemplo

```
;*****
; Delete Stack Data
;*****
;
DbTask = ModusProcess.GetComponent("OdinDbTasks")
;
;no select statement defined, the default statement is used
;
DeletedStackId = DbTask.DeleteStack("")
;
;if a stack is deleted, its ID is returned. Otherwise an empty string is
returned.
```

```

if (StrLen(DeletedStackId) > 0)
Protocol("Stack with the ID '{0}' was deleted", 10,DeletedStackId)
else
Protocol("No stack deleted", 10)
end-if

```

### DeleteProcessRecords

Exclui os conjuntos de dados do processo da tabela Odin\_Process, para a qual nenhum outro conjunto de dados de trabalho é atribuído.

### Sintaxe

DeleteProcessRecords(NumberOfDataSets)

### Parâmetro

Parâmetro	Significado
NumberOfDataSets	Número máximo de conjuntos de dados de processo a serem excluídos pela chamada de método. Quanto maior o número, maior o tempo de execução.

### Valor de retorno

Número de processos excluídos.

### Script de exemplo

```

;*****
Delete Process Data
;*****
;
DbTask = ModusProcess.GetComponent("OdinDbTasks")
ProcessCounter = DbTask.DeleteProcessRecords(100)
Protocol("'{0}' Process(es) were deleted", 10, ProcessCounter)

```

### SetBackStackState

Redefinição automática de pilhas com o status **Bloqueada** para o status **Aguardar**.

Cada chamada desse método busca um conjunto de dados de pilha que atende às seguintes condições:

ODS_STATE	= 3 (Locked)
ODS_RETRY_AT	<= current time
ODS_LOCKBY	is NULL

Se for encontrado um conjunto de dados, ele será redefinido.

Além disso, os seguintes campos de banco de dados serão atualizados:

ODS\_STATE = 0 (Wait)

ODS\_MESSAGE\_TEXT = 'automatic reset'

### Sintaxe

SetBackStackState()

### Parâmetro

-

### Valor de retorno

A ID da pilha redefinida ou uma string vazia, se nenhuma pilha for encontrada/processada.

### Script de exemplo

```
;*****
;Reset stack with status Locked to Wait
;*****
;
DbTask = ModusProcess.GetComponent("OdinDbTasks")
StackId = DbTask.SetBackStackState()
;
;if a stack is reset, its ID is returned. Otherwise an empty string is
returned.
if (StrLen(StackId) > 0)
Protocol("Stack with the ID {0} was reset", 10, StackId)
else
Protocol("No stack reset", 10)
end-if
```

### Script de exemplo OdinDbTasks

```
DbTask = ModusProcess.GetComponent("OdinDbTasks")
StackId = DbTask.SetBackStackState()
;
;if a stack is reset, its ID is
;returned. Otherwise an empty
;string is returned
if (StrLen(StackId) > 0)
Protocol("Stack with the ID '{0}' was reset", 10, StackId)
else
Protocol("No stack reset", 10)
end-if
```

OdinEmail

### Informações gerais

Esse componente permite que você envie emails por um servidor SMTP.

### ***Propriedades de OdinEmail***

Propriedade	Significado
BodyHtml	Define ou retorna o texto de corpo HTML do email.
BodyHmtlImages	<p>Define ou retorna o corpo das imagens HTML como uma string separada por vírgula.</p> <p>Durante a definição da string, é possível ter a seguinte estrutura:</p> <ul style="list-style-type: none"> <li>• id=nome do arquivo Exemplo: pic1=c:\images\pic1.jpg,picN=c:\images\picN.jpg</li> <li>• somente o nome do arquivo (aqui o nome sem uma extensão de arquivo é usado como ID) Exemplo: c:\images\pic1.jpg,c:\images\picN.jpg</li> <li>• ou uma combinação Exemplo: pic1=c:\images\pic1.jpg,c:\images\test.jpg,picN=c:\images\picN.jpg</li> </ul> <p>Valores retornados sempre são feitos no formulário da id=nome do arquivo</p>
BodyText	Define ou retorna o texto de corpo do email.
Encoding	Define ou retorna a codificação usada para o email.
EncodingCodepage	Define ou retorna a página de codificação / código usados para o email.
IsBodyHtml	<p>Define ou retorna se o email estiver no formato HTML.</p> <p>True: BodyHtml é usado como texto HTML e BodyText como alternativa de texto simples.</p> <p>False: Somente o BodyText é usado como texto simples.</p>
MailBcc	Define ou retorna o destinatário Cco do email.
MailCC	Define ou retorna o destinatário Cc do email (separado por vírgula).
MailFrom	Define ou retorna o remetente do email.
MailReplyTo	Define ou retorna a resposta para o endereço de email.
MailTo	Define ou retorna o destinatário do email. (separado por vírgula).
SmtpDomain	Especifica o domínio ou o computador no qual as informações de login serão verificadas.

SmtpHost	Host no qual o servidor SMTP está em execução.
SmtpPassword	Senha para autenticar no servidor SMTP. A senha deve ser codificada. Para codificar a senha, use a ferramenta <b>Encoder.exe</b> no diretório bin.
SmtpPickupDirectory	Define ou retorna o diretório em que os emails criados são salvos, se o SmtpDeliveryMethod é definido para <b>SpecifiedPickupDirectory</b> .
SmtpPort	Porta do servidor SMTP.
SmtpUserName	Nome de usuário para a autenticação do servidor SMTP.
Subject	Define ou retorna o assunto do email.
UseDefaultCredentials	True: Autenticar com o servidor SMTP usando o usuário conectado. False: Autenticar com o servidor SMTP usando SmtpUserName / SmtpPassword.

## ***Métodos de OdinEmail***

### **AddAttachment**

Adiciona um anexo a um email.

#### **Sintaxe**

AddAttachment(string fileName)

#### **Parâmetro**

Parâmetro	Significado
filename	Nome do arquivo anexado.

#### **Valor de retorno**

-

### **AddBodyHtmlImage**

Adiciona uma imagem ao corpo HTML.

#### **Sintaxe**

AddBodyHtmlImage(string id, string imageFileName)

**Parâmetro**

Parâmetro	Significado
id	ID da imagem (cid). Se nenhum ID for especificado, será usado o nome do arquivo sem a extensão.
imageFileName	Nome do arquivo da imagem

**Valor de retorno**

-

**AddToHeader**

Adiciona uma entrada de nome/valor a um cabeçalho de email.

**Sintaxe**

AddToHeader(string name, string value)

**Parâmetro**

Parâmetro	Significado
name	Nome da entrada. Se a entrada já existir, seu valor é sobrescrito. Em caso contrário, uma nova entrada é criada.
value	Valor da entrada.

**Valor de retorno**

-

**Exemplo**

```
Email.AddToHeader("Organization", "Perceptive Software")
```

**ClearAttachments**

Exclui todos os anexos carregados.

**Sintaxe**

ClearAttachments()

**Parâmetro**

-

**Valor de retorno**

-

**ClearBodyHtmlImages**

Exclui todas as imagens carregadas no corpo HTML.

**Sintaxe**

ClearBodyHtmlImages()

**Parâmetro**

-

**Valor de retorno**

-

**ClearCertificates**

Exclui todos os certificados carregados/definidos.

**Sintaxe**

ClearCertificates()

**Parâmetro**

**Valor de retorno**

-

**GetAttachments**

Retorna uma string separada por vírgulas contendo os nomes de arquivo de todos os anexos.

**Sintaxe**

GetAttachments()

**Parâmetro**

-

**Valor de retorno**

-

**Exemplo**

**GetDeliveryNotificationOptions**

Retorna as opções de notificações para emails.

**Sintaxe**

```
GetDeliveryNotificationOptions()
```

**Parâmetro**

-

**Valor de retorno**

Valor	Significado
OdinDeliveryNotificationOptions.None	Sem notificação
OdinDeliveryNotificationOptions.OnSuccess	Notificação de entrega com êxito
OdinDeliveryNotificationOptions.OnFailure	Notificação de entrega mal sucedida
OdinDeliveryNotificationOptions.Delay	Notificação de entrega atrasada
OdinDeliveryNotificationOptions.Never	Sem nenhuma notificação

**GetMailPriority**

Retorna a prioridade de um email.

**Sintaxe**

```
GetMailPriority()
```

**Parâmetro**

-

**Valor de retorno**

Valor	Significado
OdinMailPriority.Normal	O email tem prioridade normal.
OdinMailPriority.Low	O email tem baixa prioridade.
OdinMailPriority.High	O email tem alta prioridade.



**GetSmtpDeliveryMethod**

Retorna a opção de como os emails serão entregues.

**Sintaxe**

```
GetSmtpDeliveryMethod()
```

**Parâmetro**

-

**Valor de retorno**

Valor	Significado
OdinSmtpDeliveryMethod.Network	Os emails são enviados por meio do servidor SMTP configurado (SmtpHost / SmtpPort).
OdinSmtpDeliveryMethod.SpecifiedPickupDirectory	Os emails são copiados para o diretório especificado na propriedade SmtpPickupDirectory e devem ser enviados por um aplicativo externo.
OdinSmtpDeliveryMethod.PickupDirectoryFromIis	Os emails são copiados para o diretório configurado no Internet Information Server (IIS).

**LoadBodyFromFile**

Carrega uma caixa de texto a partir de um arquivo.

**Sintaxe**

```
LoadBodyFromFile(string fileName, bool isHtml)
```

**Parâmetro**

Parâmetro	Significado
fileName	Nome do arquivo
isHtml	True / False

**Valor de retorno**

-

**LoadCertificateFromCertFile**

Carrega o certificado X.509 do arquivo especificado para enviar o e-mail.

#### Sintaxe

LoadCertificateFromCertFile(string fileName)

#### Parâmetro

Parâmetro	Significado
fileName	Nome do arquivo

#### Valor de retorno

-

#### LoadCertificateFromSignedFile

Carrega o certificado X.509 do arquivo especificado para enviar o e-mail.

#### Sintaxe

LoadCertificateFromSignedFile(string fileName)

#### Parâmetro

Parâmetro	Significado
fileName	Nome do arquivo

#### Valor de retorno

-

#### RemoveAttachment

Remove um anexo de um email.

#### Sintaxe

RemoveAttachment(string fileName)

#### Parâmetro

Parâmetro	Significado
fileName	Nome do arquivo a ser removido dos anexos.

#### Valor de retorno

-

**RemoveFromHeader**

Remove uma entrada de um cabeçalho de email.

**Sintaxe**

RemoveFromHeader(string name)

**Parâmetro**

Parâmetro	Significado
name	Nome da entrada a ser removida.

**Valor de retorno**

-

**Exemplo**

```
Email.RemoveFromHeader("Organisation")
```

**Redefinir**

Redefine as configurações para definir os valores dos componentes.

**Sintaxe**

Reset()

**Parâmetro**

-

**Valor de retorno**

-

**Send**

Envia o email com as configurações atuais.

**Sintaxe**

Send()

**Parâmetro**

-

**Valor de retorno**

-

**SetCredentials**

Define as credenciais de usuário (nome de usuário, senha e domínio).

**Sintaxe**

SetCredentials(string userName, string password, string domain)

**Parâmetro**

Parâmetro	Significado
userName	Nome de usuário
password	Senha A senha deve ser codificada. Para codificar a senha, use a ferramenta <b>Encoder.exe</b> no diretório bin.
domain	Domínio

**Valor de retorno**

-

**SetDeliveryNotificationOptions**

Define a notificação de entrega de emails.

**Sintaxe**

SetDeliveryNotificationOptions(deliveryNotificationOptions)

**Parâmetro**

Parâmetro	Significado
deliveryNotificationOptions	<p>OdinDeliveryNotificationOptions.None : Sem notificação.</p> <p>OdinDeliveryNotificationOptions.OnSuccess: Notificação de entrega com êxito.</p> <p>OdinDeliveryNotificationOptions.OnFailure: Notificação de entrega com falha.</p> <p>OdinDeliveryNotificationOptions.Delay: Notificação de entrega com atraso.</p> <p>OdinDeliveryNotificationOptions.Never: Nenhum tipo de notificação.</p>

**Valor de retorno**

-

**SetMailPriority**

Define a prioridade de um email.

**Sintaxe**

SetMailPriority(mailPriority)

**Parâmetro**

Parâmetro	Significado
mailPriority	<b>Valores possíveis:</b> OdinMailPriority.Normal: A prioridade do email é normal. OdinMailPriority.Low: A prioridade do email é baixa. OdinMailPriority.High: A prioridade do email é alta.

**Valor de retorno**

-

**SetSmtpDeliveryMethod**

Define como os emails devem ser enviados.

**Sintaxe**

SetSmtpDeliveryMethod(smtpDeliveryMethod)

**Parâmetro**

Parâmetro	Significado
smtpDeliveryMethod	<b>Valores possíveis:</b> OdinSmtpDeliveryMethod.Network: Emails são enviados por meio do servidor SMTP configurado (SmptHost / Smtport). OdinSmtpDeliveryMethod.SpecifiedPickupDirectory: Emails são copiados ao diretório especificado na propriedade SmtppickupDirectory e devem ser enviado por um aplicativo externo. OdinSmtpDeliveryMethod.PickupDirectoryFromIis: Emails são copiados para o diretório configurado no Internet Information Server (IIS).

**Valor de retorno**

-

**SetSmtpServer**

Define as propriedades do servidor SMTP para host, porta e UseDefaultCredentials.

**Sintaxe**

SetSmtpServer(string host, int port, bool useDefaultCredentials)

**Parâmetro**

Parâmetro	Significado
host	Host no qual o servidor SMTP está em execução.
port	Porta do servidor SMTP.
useDefaultCredentials	True: Autenticar com o servidor SMTP usando o usuário conectado. False: Autenticar com o servidor SMTP usando SmtpUserName / SmtpPassword.

**Valor de retorno**

-

**Script de exemplo do OdinEmail**

```

; get Email component
email = ModusProcess.GetComponent("OdinEmail")
;
; Test: Encoding by display name / name / code page
email.Encoding = "Western European" (Windows)
;email.Encoding = "windows-1252"
;email.EncodingCodepage = 1252
;
; SMTP-Server + User
email.SmtpHost = "192,168.24,166"
email.SmtpPort = 25
email.UseDefaultCredentials = false
email.SmtpUserName = "test"
email.SmtpPassword = "test"
email.SmtpDomain = ""
;
; Can alternatively be set by the method SetSmtpServer/SetCredentials
;email.SetSmtpServer("192,168.24,166", 25, false)
;email.SetCredentials("test", "test", "")
;
; SetSmtpDeliveryMethod (OdinSmtpDeliveryMethod ->
Network/SpecifiedPickupDirectory/PickupDirectoryFromIis)
;email.SetSmtpDeliveryMethod(OdinSmtpDeliveryMethod.Network)
;email.SmtpPickupDirectory = "c:\temp\"
;
; Email From/To/CC/Bcc/ReplyTo

```

```

; The comma separated string for email addresses can have the following
structure:
; -----
; a) "DisplayName" <email@domain>,<email@domain>,email@domain
; or: "displayName-1" <email-1@domain-1.de>,"displayName-N" <email-N@domain-
N.de>,<email-X@domain-X.de>,email-Y@domain-Y.de
; b) in string list format: DisplayName=email address or just email address
(without DisplayName)
; or.: displayName-1=email-1@domain-1.de,displayName-N=email-N@domain-N.de
; Jo=email@test.de,email@test,Jo2=email2@test.de ...
email.MailFrom = ""Jo" <jo@mydomain.de>"
email.MailTo = ""Test" <test@mydomain.de>,jo@mydomain.de"
email.MailReplyTo = ""Jo" <jo@mydomain.de>"
;
; SetMailPriority (OdinMailPriority -> Normal/High/Low)
email.SetMailPriority(OdinMailPriority.High)
;
; SetDeliveryNotification (OdinDeliveryNotificationOptions ->
None/OnSuccess/OnFailure/Delay/Never)
email.SetDeliveryNotificationOptions(OdinDeliveryNotificationOptions.None)
;
; Subject + Body
email.Subject = "test-mail..."
email.BodyText = "plain-text body"
email.BodyHtml = "<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN"><HTML><HEAD><META http-equiv=Content-Type
content=""text/html; charset=iso-8859-1"><HEAD><BODY><DIV><FONT face=Arial
color=#ff0000 size=5>test: this is a html mail from Jo</FONT></DIV><img
alt="" hspace=0 src=""cid:No-Fear" align=baseline border=0></BODY></HTML>"
email.IsBodyHtml = true
;
; Alternatively, bodies can be loaded using LoadBodyFromFile()
;email.LoadBodyFromFile("plain.txt", false) ; = BodyText
;email.LoadBodyFromFile("html.txt", true) ; = BodyHtml
;
; image is inserted using cid:No-Fear, i.e insert image....
; If no ID is specified, the file name without its extension is used as the
ID.
email.AddBodyHtmlImage("", "c:\temp\no-fear.jpg")
;
; Insert PDF as attachment
email.AddAttachment("c:\temp\test.pdf")
;
; or possibly load X509 certificate (SSL is then used)
;email.LoadCertificateFromCertFile("certFile.cer")
;email.LoadCertificateFromSignedFile("signedFile.cer")
;

```

```
; and send email...
email.Send()
```

## OdinFileConvert

### Informações gerais

Use esse componente para converter e agrupar documentos 1-n no formato de destino (PDF, TIFF ...).

Os formatos de saída disponíveis dependem da licença.

Os documentos podem ser alterados durante a conversão em lote, se necessário.

Esse componente pode ser usado em processos de arquivamento para converter uma apresentação XPS para Tiff ou PDF, por exemplo, antes de ser arquivada.

### Propriedades de OdinFileConvert

Propriedade	Significado
LicenseFilename	Caminho e nome do arquivo de licença para o Compart .NET-Toolkit.
NumberOfFilesConverted	Retorna o número de arquivos já convertidos e anexados ao arquivo de saída durante a conversão em lote.
OutputFormat	<p>Retorna o valor do formato de saída do fluxo.</p> <p>A lista suspensa mostra todos os formatos de saída possíveis como definidos no arquivo ModusSuite.Common.PropertyEditors.dll.config.</p> <p>Os formatos dependem da licença Compart .NET-Toolkit correspondente.</p> <p><b>ModusSuite.Common.PropertyEditors.dll.config de exemplo:</b></p> <pre>&lt;PropertyEditor&gt;   &lt;Odin&gt;     &lt;StackTypes&gt;       &lt;StackType value="60" name="Transfer"/&gt;     &lt;/StackTypes&gt;     &lt;OutputFormat&gt;       &lt;Format value="pdf" name="Portable Document Format" /&gt;       &lt;Format value="pcl" name="Printer Command Language" /&gt;       &lt;Format value="pos" name="Postscript" /&gt;       &lt;Format value="afp" name="Advanced Function Presentation" /&gt;       &lt;Format value="tif" name="Tagged Image File Format" /&gt;     &lt;/OutputFormat&gt;   &lt;/Odin&gt;</pre>



	</PropertyEditor>
ProfilePath	Caminho para os perfis Compart definidos nos filtros de entrada e saída respectivos.
TraceFileName	Caminho e nome do arquivo de trace para o Compart .NET-Toolkit.

## ***Métodos de OdinFileConvert***

### **AddDocument**

Converte o documento Odin especificado e o adiciona ao arquivo de saída.

#### **Sintaxe**

AddDocument(document)

#### **Parâmetro**

Parâmetro	Significado
document	Documento a ser adicionado

#### **Valor de retorno**

-

### **AddFile**

Converte o arquivo especificado e o adiciona ao arquivo de saída durante a conversão em lote.

#### **Sintaxe**

AddFile(fileName)

#### **Parâmetro**

Parâmetro	Significado
Filename	Arquivo a ser adicionado

#### **Valor de retorno**

-

### **CreatePdfSigner**

[▶ Mostrar tudo](#)

Os métodos criam um PDF signer e retornam sua interface IOdinPdfSigner.

O certificado usado para a assinatura pode ser proveniente de um arquivo de certificado exportado (PFX) ou de um repositório de certificados.

O objeto PDF signer pode ser usado para atribuir uma assinatura digital a um arquivo PDF.

Ele pode ser usado após a criação de um arquivo de saída de uma conversão em um evento OnConvertEnd, ou pode ser usado por uma impressora de email no evento OnConvert (consulte Contexto OdinEmailConvert ).

### Sintaxe

CreatePdfSigner(string pfxFileName, string password)

CreatePdfSigner(OdinStoreName storeName, OdinStoreLocation storeLocation, OdinX509FindType findType, object findValue, bool validOnly)

### Parâmetro

Parâmetro	Significado																
pfxFileName	Nome do arquivo de certificado.																
password	A senha necessária para acessar os dados de certificado X.509. A senha deve ser criptografada. Para criptografar uma senha. Use a ferramenta Encoder.exe no diretório bin.																
storeName	Um dos valores OdinStoreName: <table border="1"> <thead> <tr> <th>Valor</th> <th>Significado</th> </tr> </thead> <tbody> <tr> <td>OdinStoreName.AddressBook</td> <td>O repositório de certificados X.509 para outros usuários.</td> </tr> <tr> <td>OdinStoreName.AuthRoot</td> <td>O repositório de certificados X.509 para autoridades de certificação de terceiros.</td> </tr> <tr> <td>OdinStoreName.CertificateAuthority</td> <td>O repositório de certificados X.509 para autoridades de certificação intermediárias.</td> </tr> <tr> <td>OdinStoreName.Disallowed</td> <td>O repositório de certificados X.509 para certificados revogados.</td> </tr> <tr> <td>OdinStoreName.My</td> <td>O repositório de certificados X.509 para certificados pessoais.</td> </tr> <tr> <td>OdinStoreName.Root</td> <td>O repositório de certificados X.509 para autoridades de certificação da raiz confiáveis.</td> </tr> <tr> <td>OdinStoreName.TrustedPeople</td> <td>O repositório de certificados X.509 para</td> </tr> </tbody> </table>	Valor	Significado	OdinStoreName.AddressBook	O repositório de certificados X.509 para outros usuários.	OdinStoreName.AuthRoot	O repositório de certificados X.509 para autoridades de certificação de terceiros.	OdinStoreName.CertificateAuthority	O repositório de certificados X.509 para autoridades de certificação intermediárias.	OdinStoreName.Disallowed	O repositório de certificados X.509 para certificados revogados.	OdinStoreName.My	O repositório de certificados X.509 para certificados pessoais.	OdinStoreName.Root	O repositório de certificados X.509 para autoridades de certificação da raiz confiáveis.	OdinStoreName.TrustedPeople	O repositório de certificados X.509 para
Valor	Significado																
OdinStoreName.AddressBook	O repositório de certificados X.509 para outros usuários.																
OdinStoreName.AuthRoot	O repositório de certificados X.509 para autoridades de certificação de terceiros.																
OdinStoreName.CertificateAuthority	O repositório de certificados X.509 para autoridades de certificação intermediárias.																
OdinStoreName.Disallowed	O repositório de certificados X.509 para certificados revogados.																
OdinStoreName.My	O repositório de certificados X.509 para certificados pessoais.																
OdinStoreName.Root	O repositório de certificados X.509 para autoridades de certificação da raiz confiáveis.																
OdinStoreName.TrustedPeople	O repositório de certificados X.509 para																

		peças e recursos diretamente confiáveis.
	OdinStoreName.TrustedPublisher	O repositório de certificados X.509 para editores diretamente confiáveis.
storeLocation	Um dos valores OdinStoreLocation:	
	<b>Valor</b>	<b>Significado</b>
	OdinStoreLocation.CurrentUser	O repositório de certificados X.509 usado pelo usuário atual.
	OdinStoreLocation.LocalMachine	O repositório de certificados X.509 atribuído à máquina local.
findType	Um dos valores OdinX509FindType:	
	<b>Valor</b>	<b>Significado</b>
	OdinX509FindType.FindByThumbprint	O parâmetro findValue deve ser uma string representando a impressão digital do certificado.
	OdinX509FindType.FindBySubjectName	O parâmetro findValue deve ser uma string representando o nome do assunto do certificado.
	OdinX509FindType.FindBySubjectDistinguishedName	O parâmetro findValue deve ser uma string representando o nome diferenciado do assunto do certificado.
	OdinX509FindType.FindByIssuerName	O parâmetro findValue deve ser uma string representando o nome do emissor do certificado.
	OdinX509FindType.FindByIssuerDistinguishedName	O parâmetro findValue deve ser uma string representando o nome diferenciado do emissor do certificado.
	OdinX509FindType.FindBySerialNumber	O parâmetro findValue deve ser uma string representando o número de série do certificado, conforme exibido pela interface do usuário. O número de série deve estar em ordem inversa, pois é um número inteiro.
	OdinX509FindType.FindByTimeValid	O parâmetro findValue deve ser um valor DateTime no horário local, por exemplo,

		Now.
	OdinX509FindType.FindByTimeNotYetValid	O parâmetro findValue deve ser um valor DateTime no horário local, por exemplo, Now.
	OdinX509FindType.FindByTimeExpired	O parâmetro findValue deve ser um valor DateTime no horário local, por exemplo, Now.
	OdinX509FindType.FindByTemplateName	O parâmetro findValue deve ser uma string representando o nome do modelo do certificado, por exemplo, "ClientAuth".  Um nome de modelo é uma extensão X509 versão 3 que especifica os usos do certificado.
	OdinX509FindType.FindByApplicationPolicy	O parâmetro findValue deve ser uma string representando o nome amigável da diretiva de aplicativo ou o identificador de objeto (Oid) do certificado. Por exemplo, "Encrypting File System" ou "1.3.6.1.4.1.311.10.3.4" podem ser utilizados. Para um aplicativo localizado, o valor OID deve ser usado, pois o nome amigável está localizado.
	OdinX509FindType.FindByCertificatePolicy	O parâmetro findValue deve ser uma string representando o nome amigável ou o identificados de objeto (Oid) da política de certificação. A prática recomendada é usar o OID, como "1.3.6.1.4.1.311.10.3.4". Para um aplicativo localizado, o valor OID deve ser usado, pois o nome amigável está localizado.
	OdinX509FindType.FindByExtension	O parâmetro findValue deve ser uma string descrevendo a extensão a ser encontrada. O identificador de objeto (OID) é mais comumente usado para direcionar o método Find para pesquisar por todos os certificados com extensão correspondente ao valor OID.
	OdinX509FindType.FindByKeyUsage	O parâmetro findValue deve ser uma string representando o uso da chave ou um número inteiro, representando uma bitmask contendo todos os usos de

		chave solicitados. Apenas um uso de chave pode ser especificado para o valor da string por vez.
	OdinX509FindType.FindBySubjectKeyIdentifier	O parâmetro findValue deve ser uma string representando o identificador da chave de entidade em hexadecimal, como "F3E815D45E83B8477B9284113C64EF208E897112", conforme exibido na interface de usuário.
findValue	Critério de pesquisa como string ou data e hora.	
validOnly	True, para que apenas certificados válidos sejam retornados pela pesquisa. Em caso contrário, False	

## Valor de retorno

### Interface IOdinPdfSigner

Métodos da interface IOdinPdfSigner

AddSignature

Initialize

RegisterFont

RegisterFontDirectory

SetFont

SetPosition

SetText

SetTimeStampServer

Propriedades da interface IOdinPdfSigner

Propriedade	Significado
saServer :string	URL do servidor do carimbo de data/hora, por exemplo, "http://dse200.ncipher.com/TSS/HttpTspServer"  Caso isso esteja vazio, a hora atual do computador é suada e não detectada/criptografada pelo servidor TSA.
TsaUserName :string	Nome de usuário do servidor TSA.
TsaPassword	Senha do servidor TSA.

:string									
FieldName :string	Nome do campo de assinatura do PDF. Por padrão, "Signature".								
ImageFileName :string	Especifica o nome do arquivo de imagem a ser usada para a assinatura.								
Text :string	Texto da assinatura. (usado por OdinSignatureStyle.Text ou OdinSignatureStyle.ImageAndText).								
Reason :string	Motivo da assinatura (usado por OdinSignatureStyle.Description ou OdinSignatureStyle.ImageAndDescription).								
Location :string	Local da assinatura (usado por OdinSignatureStyle.Description ou OdinSignatureStyle.ImageAndDescription).								
Contact :string	Informações de contato do assinante.								
PageNumber :int	Página na qual a assinatura será inserida.								
Visible :bool	Especifica se a assinatura é visível.								
XPos :double	Posição X da assinatura em pontos.								
YPos :double	Posição Y da assinatura em pontos.								
Width :double	Largura da assinatura em pontos.								
Height :double	Altura da assinatura em pontos.								
FontName :string	Nome da fonte usada para a assinatura.								
FontColor :string	Cor da fonte, ou "AARRGGBB" ou "Grey"								
FontSize :int	Tamanho da fonte.								
FontStyle :int	Estilo da fonte (uma combinação de: 0=Normal, 1=Negrito, 2=Itálico, 4=Sublinhado, 8=Riscado)								
	<table border="1"> <thead> <tr> <th>Valores possíveis</th> <th>Significado</th> </tr> </thead> <tbody> <tr> <td>OdinPdfFontStyle.Normal</td> <td>0</td> </tr> <tr> <td>OdinPdfFontStyle.Bold</td> <td>1 (negrito)</td> </tr> <tr> <td>OdinPdfFontStyle.Italic</td> <td>2 (itálico)</td> </tr> </tbody> </table>	Valores possíveis	Significado	OdinPdfFontStyle.Normal	0	OdinPdfFontStyle.Bold	1 (negrito)	OdinPdfFontStyle.Italic	2 (itálico)
Valores possíveis	Significado								
OdinPdfFontStyle.Normal	0								
OdinPdfFontStyle.Bold	1 (negrito)								
OdinPdfFontStyle.Italic	2 (itálico)								

	OdinPdfFontStyle.Underline	4 (sublinhado)	
	OdinPdfFontStyle.Strikethrough	8 (riscado)	
	Os valores inteiros podem ser definidos em combinação, por exemplo: OdinPdfFontStyle.Bold + OdinPdfFontStyle.Italic		
SignatureImage :string	Imagem da assinatura (exibida por OdinSignatureStyle.ImageAndText ou OdinSignatureStyle.ImageAndDescription).		
PdfPassword :string	Senha do proprietário do PDF		

### Script de amostra

```

; get OdinFileConvert component
OdinFileConvert = ModusProcess.GetComponent("OdinFileConvert")
;
; define variables
; TimeStamp-Server
tsaServer = "http://dse200.ncipher.com/TSS/HttpTspServer"
tsaUser = ""
tsaPassword = ""
;
; Pdf-/Image-Filename
inputPdf = "C:\Perceptive\modusSuite\data\Test.pdf"
outputPdf = "C:\Perceptive\modusSuite\data\Test_signed.pdf"
pdfPassword = ""
imageFileName = "C:\Perceptive\modusSuite\data\signature.JPG"
bgndImageFileName = ""
;
; Signature-Text
fieldName = "Signature"
reason = "Reason"
location = "Place"
contact = "Contact"
signatureText = "Signature"
signatureStyle = OdinSignatureStyle.Text
;
; Signature position
visible = true
pageNo = 1
xpos = 100
ypos = 100
width = 150

```

```

height = 50
;
; Font
fontName = "Verdana"
fontSize = 8
fontColor = "00808080"
fontStyle = OdinPdfFontStyle.Bold + OdinPdfFontStyle.Italic
;
; create Pdf-Signer with certificate from a PFX file
;pdfSigner =
OdinFileConvert.CreatePdfSigner("C:\Perceptive\modusSuite\Certificates\PS.pfx
", "PerceptiveSoftware")
; create Pdf-Signer with a certificate from the store (My Certificates /
Current Users).
pdfSigner = OdinFileConvert.CreatePdfSigner(OdinStoreName.My,
OdinStoreLocation.CurrentUser, OdinX509FindType.FindBySubjectName,
"PerceptiveSoftware", true)
;
; initialize signer
pdfSigner.Initialize(signatureStyle, pageNo, visible, fieldName,
imageFileName)
;
; initialize position
pdfSigner.SetPosition(xpos, ypos, width, height, bgndImageFileName)
;
; initialize text/description
;pdfSigner.SetText("", reason, location, contact)
pdfSigner.SetText(signatureText, reason, location, contact)
;
; initialize font
pdfSigner.SetFont(fontName, fontSize, fontStyle, fontColor)
;
; initialize time stamp server
pdfSigner.SetTimeStampServer(tsaServer, tsaUser, tsaPassword)
;
; insert signature to PDF
pdfSigner.AddSignature(inputPdf, outputPdf, pdfPassword)

```

### ConvertSingleFile

[► Mostrar tudo](#)

Converte um único arquivo em um formato especificado.

Os formatos disponíveis são dependentes da licença Compant para filtros de entrada e de saída.

### Sintaxe

```
ConvertSingleFile(inputFileName, outputFileName)
```



ConvertSingleFile(inputFileName, outputFileName, singlePage)

ConvertSingleFile(inputFileName, outputFileName, singlePage, convertOptions)

### Parâmetro

Parâmetro	Significado										
inputFileName	O arquivo a ser convertido										
outputFileName	O arquivo a ser criado										
singlePage	<p>Quando verdadeiro, grava um arquivo de saída por página, de acordo com a seguinte convenção de nomenclatura FILENAME_#PAGENUMBER#.outputformat</p> <p><b>Exemplo:</b> OutputFilename_001.pdf</p> <p>Se o parâmetro não estiver definido, o valor padrão é <b>False</b>.</p>										
convertOptions	<p>Usado para especificar opções de conversão e metadados.</p> <p>As opções são definidas na forma de uma string separada por vírgulas. Seu uso é dependente do formato de saída.</p> <p>Os parâmetros são opcionais.</p> <p>Metadados</p> <table border="1"> <tbody> <tr> <td>TITLE</td> <td>Título do documento.</td> </tr> <tr> <td>SUBJECT</td> <td>Assunto do documento.</td> </tr> <tr> <td>AUTHOR</td> <td>Autor do documento.</td> </tr> <tr> <td>KEYWORDS</td> <td>Palavras-chave do documento.</td> </tr> </tbody> </table> <p>Opções</p> <table border="1"> <tbody> <tr> <td>TEXTNOTE_CHANGES</td> <td>(Pdf 1.3/1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de</td> </tr> </tbody> </table>	TITLE	Título do documento.	SUBJECT	Assunto do documento.	AUTHOR	Autor do documento.	KEYWORDS	Palavras-chave do documento.	TEXTNOTE_CHANGES	(Pdf 1.3/1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de
TITLE	Título do documento.										
SUBJECT	Assunto do documento.										
AUTHOR	Autor do documento.										
KEYWORDS	Palavras-chave do documento.										
TEXTNOTE_CHANGES	(Pdf 1.3/1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de										

	segurança de PDF para Montagem de documento, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como verdadeiro/falso.
CHANGES	(Pdf 1.3/1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de segurança de PDF para Montagem de documento, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como verdadeiro/falso.
FILL	(Pdf 1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de segurança de PDF para Montagem de documento, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como verdadeiro/falso.
COPY	(Pdf 1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de segurança de PDF para Montagem de documento, Conteúdo copiado, Conteúdo copiado para acessibilidade, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como verdadeiro/falso.
DIGITAL_COPY	(Pdf 1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de segurança de PDF para Montagem de documento, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como

	verdadeiro/falso.
ASSEMBLE	(Pdf 1.4) TRUE/FALSE A montagem de um ou mais documentos não é permitida. Com a especificação deste parâmetro, os atributos de segurança de PDF para Montagem de documento, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como verdadeiro/falso.
ACCESSIBILITY	(Pdf 1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de segurança de PDF para Conteúdo copiado, Conteúdo copiado para acessibilidade, Extração de página serão definidos como verdadeiro/falso.
DUPLEX_ADDPAGE	TRUE: Esta opção especifica se uma página vazia deve ser adicionada a um documento duplex com um número ímpar de páginas. FALSE: Configuração padrão. Esta opção define o modo simplex para a última página de um documento duplex com um número ímpar de páginas.
DUPLEX_OPTIMIZE	TRUE/FALSE Quando verdadeiro, as configurações de duplex são definidas apenas no fluxo de saída, quando o modo duplex muda.
MAP_COLOR	TRUE/FALSE Este parâmetro é usado para definir se o mapeamento de cores da definição AFP-HighlightColor será executado ou não. As definições de cores individuais podem ser

especificadas como MapColor\_1 até MapColor\_n.

**Exemplo:**

```
Map_Color=TRUE,"MapColor_1=-1,20,black","MapColor_2=0,20,red"
```

**Observação:**

Como os parâmetros MapColor contêm valores separados por vírgulas, a string completa deve estar entre aspas.

**A estrutura é a seguinte:**

```
MapColor_???= HSB-Color,Threshold,ColorName
HSB-Color=Ângulo da cor HSB
Threshold= Valor limite da cor HSB
ColorName= Nome da cor como definido no perfil AFP, em HighlightColorList.
```

**O exemplo acima requer estas definições no perfil AFP:**

```
</highlightcolorlist>
  <highlightcolor
name='red' type='OCA'
number='0x0002' foreground-
rgb="#00FF0000" />
  <highlightcolor
name='black' type='OCA'
number='0x0008' foreground-
rgb="#00000000" />
</highlightcolorlist>
```

e

```
<optimizepa value="never" />
<color value="FALSE" />
<usehighlightcolor
```

	<pre>value="TRUE" /&gt; &lt;enablebilevelimage value="TRUE" /&gt;</pre>
SHOW_BOOKMARKS	<p>TRUE: Os bookmarks são mostrados quando o arquivo PDF é aberto.</p> <p>FALSE: Os bookmarks não são mostrados quando o arquivo PDF é aberto.</p>
OVERLAYS	<p>TRUE: São criadas OVERLAYS.</p> <p>FALSE: Padrão. Não são criadas OVERLAYS.</p>
MERGEFONTS	<p>TRUE: Padrão</p> <p>FALSE</p>
MERGE_SPLITWORDS	<p>Ativa e desativa a otimização da saída PCL.</p> <p>Defina esta opção como FALSE se você tiver problemas de espaçamento de caracteres em um PCL gerado.</p> <p>A configuração padrão é TRUE.</p> <p>TRUE: Padrão, a otimização da saída PCL é ativada.</p> <p>FALSE: A otimização da saída PCL é desativada.</p>

**Valor de retorno**

-

**ConvertXDocToHtml**

Converte xdoc para HTML.

**Sintaxe**

ConvertXDocToHtml(xDocFileName, htmlOutputFileName, saveImages)

**Parâmetro**

Parâmetro	Significado
-----------	-------------

xDocFileName	O arquivo xdoc a ser convertido.
htmlOutputFileName	O arquivo HTML a ser criado
savelmages	True / False True significa salvar imagens no mesmo diretório.

**Valor de retorno**

-

**ConvertXDocToString**

Converte um arquivo xdoc (Word dinâmico) para texto (string) usando um caractere de quebra de linha especificado para o ambiente ou definido como um parâmetro.

**Sintaxe**

```
ConvertXDocToString(string xDocFileName)
```

```
ConvertXDocToString(string xDocFileName, string newLine)
```

**Parâmetro**

Parâmetro	Significado
xDocFileName	O arquivo xdoc a ser convertido
newLine	Define o tipo de quebra de linha. <b>Valores possíveis:</b> NewLineCRLF: CarriageReturn/LineFeed ("\r\n" - \$0D \$0A) NewLineLF: LineFeed ("\n" - \$0A)

**Valor de retorno**

-

**ConvertXDocToText**

Converte o xdoc para texto.

**Sintaxe**

```
ConvertXDocToText(string xDocFileName, string txtOutputFileName)
```

```
ConvertXDocToText(string xDocFileName, string txtOutputFileName, string encoding)
```

```
ConvertXDocToText(string xDocFileName, string txtOutputFileName, string encoding, string newLine)
```

**Parâmetro**

Parâmetro	Significado																																	
xDocFileName	O arquivo xdoc a ser convertido.																																	
txtOutputFileName	O arquivo de texto a ser criado																																	
encoding	<p>O texto a seguir pode ser usado como parâmetro de codificação:</p> <ul style="list-style-type: none"> <li>String vazia. Nesse caso, a codificação padrão (página de código ANSI do sistema operacional) é usada.</li> <li>página de código</li> <li>name</li> <li>nome amigável</li> </ul> <p>Exemplo:</p> <ul style="list-style-type: none"> <li>"1252"</li> <li>"windows-1252"</li> <li>"Europeu Ocidental" (Windows)</li> </ul> <p>Possíveis parâmetros de codificação</p> <table border="1"> <thead> <tr> <th>Página de código</th> <th>Nome</th> <th>Nome para exibição</th> </tr> </thead> <tbody> <tr> <td>37</td> <td>IBM037</td> <td>IBM EBCDIC (EUA-Canadá)</td> </tr> <tr> <td>437</td> <td>IBM437</td> <td>OEM EUA</td> </tr> <tr> <td>500</td> <td>IBM500</td> <td>IBM EBCDIC (Internacional)</td> </tr> <tr> <td>708</td> <td>ASMO-708</td> <td>Árabe (ASMO 708)</td> </tr> <tr> <td>720</td> <td>DOS-720</td> <td>Árabe (DOS)</td> </tr> <tr> <td>737</td> <td>ibm737</td> <td>Grego (DOS)</td> </tr> <tr> <td>775</td> <td>ibm775</td> <td>Báltico (DOS)</td> </tr> <tr> <td>850</td> <td>ibm850</td> <td>Europeu Ocidental (DOS)</td> </tr> <tr> <td>852</td> <td>ibm852</td> <td>Europeu Oriental (DOS)</td> </tr> <tr> <td>855</td> <td>IBM855</td> <td>OEM Cirílico</td> </tr> </tbody> </table>	Página de código	Nome	Nome para exibição	37	IBM037	IBM EBCDIC (EUA-Canadá)	437	IBM437	OEM EUA	500	IBM500	IBM EBCDIC (Internacional)	708	ASMO-708	Árabe (ASMO 708)	720	DOS-720	Árabe (DOS)	737	ibm737	Grego (DOS)	775	ibm775	Báltico (DOS)	850	ibm850	Europeu Ocidental (DOS)	852	ibm852	Europeu Oriental (DOS)	855	IBM855	OEM Cirílico
Página de código	Nome	Nome para exibição																																
37	IBM037	IBM EBCDIC (EUA-Canadá)																																
437	IBM437	OEM EUA																																
500	IBM500	IBM EBCDIC (Internacional)																																
708	ASMO-708	Árabe (ASMO 708)																																
720	DOS-720	Árabe (DOS)																																
737	ibm737	Grego (DOS)																																
775	ibm775	Báltico (DOS)																																
850	ibm850	Europeu Ocidental (DOS)																																
852	ibm852	Europeu Oriental (DOS)																																
855	IBM855	OEM Cirílico																																

857	ibm857	Turco (DOS)
858	IBM00858	OEM Multilíngue Latino I
860	IBM860	Português (DOS)
861	ibm861	Islandês (DOS)
862	DOS-862	Hebraico (DOS)
863	IBM863	Francês do Canadá (DOS)
864	IBM864	Árabe (864)
865	IBM865	Nórdico (DOS)
866	cp866	Cirílico (DOS)
869	ibm869	Grego moderno (DOS)
870	IBM870	IBM EBCDIC (Multilíngue Latino 2)
874	windows-874	Tailandês (Windows)
875	cp875	IBM EBCDIC (Grego moderno)
932	shift_jis	Japonês (Shift-JIS)
936	gb2312	Chinês simplificado (GB2312)
949	ks_c_5601-1987	Coreano
950	big5	Chinês (tradicional) (Big5)
1026	IBM1026	IBM EBCDIC (Turco, Latino 5)
1047	IBM01047	IBM Latino 1
1140	IBM01140	IBM EBCDIC (EUA-Canadá-Europeu)



1141	IBM01141	IBM EBCDIC (Alemanha-Europeu)
1142	IBM01142	IBM EBCDIC (Dinamarca-Noruega-Europeu)
1143	IBM01143	IBM EBCDIC (Finlândia-Suécia-Europeu)
1144	IBM01144	IBM EBCDIC (Italiano-Europeu)
1145	IBM01145	IBM EBCDIC (Espanha-Europeu)
1146	IBM01146	IBM EBCDIC (Grã-Bretanha-Europeu)
1147	IBM01147	IBM EBCDIC (França-Europeu)
1148	IBM01148	IBM EBCDIC (Internacional-Europeu)
1149	IBM01149	IBM EBCDIC (Islandês-Europeu)
1200	utf-16	Unicode
1201	unicodeFFFE	Unicode (Big Endian)
1250	windows-1250	Centro-europeu (Windows)
1251	windows-1251	Cirílico (Windows)
1252	Windows-1252	Europeu Ocidental (Windows)
1253	windows-1253	Grego (Windows)
1254	windows-1254	Turco (Windows)
1255	windows-1255	Hebraico (Windows)

1256	windows-1256	Árabe (Windows)
1257	windows-1257	Báltico (Windows)
1258	windows-1258	Vietnamita (Windows)
1361	Johab	Coreano (Johab)
10000	macintosh	Europeu Ocidental (Mac)
10001	x-mac-japanese	Japonês (Mac)
10002	x-mac-chinesetrad	Chinês tradicional (Mac)
10003	x-mac-korean	Coreano (Mac)
10004	x-mac-arabic	Árabe (Mac)
10005	x-mac-hebrew	Hebraico (Mac)
10006	x-mac-greek	Grego (Mac)
10007	x-mac-cyrillic	Cirílico (Mac)
10008	x-mac-chinesesimp	Chinês simplificado (Mac)
10010	x-mac-romanian	Romeno (Mac)
10017	x-mac-ukrainian	Ucraniano (Mac)
10021	x-mac-thai	Tailandês (Mac)
10029	x-mac-ce	Centro-europeu (Mac)
10079	x-mac-icelandic	Islandês (Mac)
10081	x-mac-turkish	Turco (Mac)
10082	x-mac-croatian	Croata (Mac)
12000	utf-32	Unicode (UTF-32)

12001	utf-32BE	Unicode (UTF-32-Big Endian)
20000	x-Chinese-CNS	Chinês tradicional (CNS)
20001	x-cp20001	TCA Taiwan
20002	x-Chinese-Eten	Chinês tradicional (Eten)
20003	x-cp20003	IBM5550 Taiwan
20004	x-cp20004	TeleText Taiwan
20005	x-cp20005	Wang Taiwan
20105	x-IA5	Europeu Ocidental (IA5)
20106	x-IA5-German	Alemão (IA5)
20107	x-IA5-Swedish	Sueco (IA5)
20108	x-IA5-Norwegian	Norueguês (IA5)
20127	us-ascii	US-ASCII
20261	x-cp20261	T.61
20269	x-cp20269	ISO-6937
20273	IBM273	IBM EBCDIC (Alemanha)
20277	IBM277	IBM EBCDIC (Dinamarca-Noruega)
20278	IBM278	IBM EBCDIC (Finlândia-Suécia)
20280	IBM280	IBM EBCDIC (Itália)
20284	IBM284	IBM EBCDIC (Espanha)
20285	IBM285	IBM EBCDIC (Grã-Bretanha)
20290	IBM290	IBM EBCDIC (Japonês)

			katakana)
20297	IBM297		IBM EBCDIC (França)
20420	IBM420		IBM EBCDIC (Árabe)
20423	IBM423		IBM EBCDIC (Grego)
20424	IBM424		IBM EBCDIC (Hebraico)
20833	x-EBCDIC-KoreanExtended		IBM EBCDIC (Coreano estendido)
20838	IBM-Thai		IBM EBCDIC (Tailandês)
20866	koi8-r		Cirílico (KOI8-R)
20871	IBM871		IBM EBCDIC (Islandês)
20880	IBM880		IBM EBCDIC (Cirílico russo)
20905	IBM905		IBM EBCDIC (Turco)
20924	IBM00924		IBM Latino 1
20932	EUC-JP		Japonês (JIS 0208-1990 e 0212-1990)
20936	x-cp20936		GB2312-80 Chinês (simplificado)
20949	x-cp20949		Coreano Wansung
21025	cp1025		IBM EBCDIC (Cirílico servo-búlgaro)
21866	koi8-u		Cirílico (KOI8-U)
28591	iso-8859-1		Europeu Ocidental (ISO)
28592	iso-8859-2		Centro-europeu (ISO)
28593	iso-8859-3		Latino 3 (ISO)

28594	iso-8859-4	Báltico (ISO)
28595	iso-8859-5	Cirílico (ISO)
28596	iso-8859-6	Árabe (ISO)
28597	iso-8859-7	Grego (ISO)
28598	iso-8859-8	Hebraico (ISO-Visual)
28599	iso-8859-9	Turco (ISO)
28603	iso-8859-13	Estoniano (ISO)
28605	iso-8859-15	Latino 9 (ISO)
29001	x-Europa	Europa
38598	iso-8859-8-i	Hebraico (ISO-lógico)
50220	iso-2022-jp	Japonês (JIS)
50221	csISO2022JP	Japonês (JIS, 1 byte kana permitido)
50222	iso-2022-jp	Japonês (JIS, 1 byte kana permitido-SO/SI)
50225	iso-2022-kr	Coreano (ISO)
50227	x-cp50227	ISO-2022 Chinês simplificado
51932	euc-jp	Japonês (EUC)
51936	EUC-CN	Chinês simplificado (EUC)
51949	euc-kr	Coreano (EUC)
52936	hz-gb-2312	Chinês simplificado (HZ)
54936	GB18030	GB18030 Chinês simplificado

	57002	x-iscii-de	ISCII Devanágari
	57003	x-iscii-be	ISCII Bengalês
	57004	x-iscii-ta	ISCII Tâmil
	57005	x-iscii-te	ISCII Télego
	57006	x-iscii-as	ISCII Assamês
	57007	x-iscii-or	ISCII Oriá
	57008	x-iscii-ka	ISCII Kannada
	57009	x-iscii-ma	ISCII Malaiala
	57010	x-iscii-gu	ISCII Guzerate
	57011	x-iscii-pa	ISCII Punjabi
	65000	utf-7	Unicode (UTF-7)
	65001	utf-8	Unicode (UTF-8)
newLine	Define o tipo de quebra de linha. <b>Valores possíveis:</b> NewLineCRLF: CarriageReturn/LineFeed ("\r\n" - \$0D \$0A) NewLineLF: LineFeed ("\n" - \$0A)		

**Valor de retorno**

-

**ConvertXDocToXps**

Converte xdoc (Word dinâmico) para XPS.

**Sintaxe**

ConvertXDocToXps(string xDocFileName, string xpsOutputFileName)

**Parâmetro**

Parâmetro	Significado
xDocFileName	O arquivo xdoc a ser convertido.
xpsOutputFileName	O arquivo XPS a ser criado

**Valor de retorno**

-

**End**

Termina o processamento e libera os respectivos recursos.

**Sintaxe**

End()

**Parâmetro**

-

**Valor de retorno**

-

**LoadDocument**

Carrega o arquivo especificado e retorna um documento Odin (interface IOdinStreamDocument) com o qual, por exemplo, a saída de texto pode ser gravada em uma página do documento.

Consulte também [Contexto IOdinStreamDocument](#)

**Sintaxe**

LoadDocument(fileName)

**Parâmetro**

Parâmetro	Significado
fileName	O arquivo a ser carregado

**Valor de retorno**

Interface IOdinStreamDocument com as seguintes propriedades e métodos:

AddBookmark

[AddComment](#)

[AddPJLCommand](#)

[EditPage](#)

[GetInputBin](#)[Image](#)[IsDuplex](#)[IsPortrait](#)[Line](#)[PageCount](#)[Post](#)[Rectangle](#)[SetFont](#)[SetInputBin](#)[TextOut](#)

### NumberOfFilesConverted

Retorna o número de arquivos já convertidos e anexados ao arquivo de saída durante a conversão em lote.

#### Sintaxe

NumberOfFilesConverted()

#### Parâmetro

-

#### Valor de retorno

Número de arquivos convertidos.

#### Iniciar

[▶ Mostrar tudo](#)

Inicializa o processamento para conversão em lote. Usando os métodos AddFile e AddDocument, documentos 1-n podem ser gravados no arquivo de saída.

#### Sintaxe

Start(outputFileName)

Start(outputFileName, singlePage)

Start(outputFileName, singlePage, convertOptions)

#### Parâmetro

Parâmetro	Significado
-----------	-------------



outputFileName	Nome do arquivo de saída														
singlePage	<p>True / False</p> <p>Quando verdadeiro, escreve um arquivo de saída por página, de acordo com a seguinte convenção de nomenclatura</p> <p>FILENAME_#PAGENUMBER#.outputformat</p> <p><b>Exemplo:</b></p> <p>OutputFilename_001.pdf</p> <p>Se o parâmetro não estiver definido, o valor padrão é <b>False</b>.</p>														
convertOptions	<p>Usado para especificar opções de conversão e metadados.</p> <p>As opções são definidas na forma de uma string separada por vírgulas. Seu uso é dependente do formato de saída.</p> <p>Os parâmetros são opcionais, ou seja, não são totalmente necessários.</p> <p>Metadados</p> <table border="1"> <tr> <td>TITLE</td> <td>Título do documento.</td> </tr> <tr> <td>SUBJECT</td> <td>Assunto do documento.</td> </tr> <tr> <td>AUTHOR</td> <td>Autor do documento.</td> </tr> <tr> <td>KEYWORDS</td> <td>Palavras-chave do documento.</td> </tr> </table> <p>Opções</p> <table border="1"> <tr> <td></td> <td>(Pdf 1.3/1.4) TRUE/FALSE</td> </tr> <tr> <td>TEXTNOTE_CHANGES</td> <td>Com a especificação deste parâmetro, os atributos de segurança de PDF para Montagem de documento, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como verdadeiro/falso.</td> </tr> <tr> <td>CHANGES</td> <td>(Pdf 1.3/1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de</td> </tr> </table>	TITLE	Título do documento.	SUBJECT	Assunto do documento.	AUTHOR	Autor do documento.	KEYWORDS	Palavras-chave do documento.		(Pdf 1.3/1.4) TRUE/FALSE	TEXTNOTE_CHANGES	Com a especificação deste parâmetro, os atributos de segurança de PDF para Montagem de documento, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como verdadeiro/falso.	CHANGES	(Pdf 1.3/1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de
TITLE	Título do documento.														
SUBJECT	Assunto do documento.														
AUTHOR	Autor do documento.														
KEYWORDS	Palavras-chave do documento.														
	(Pdf 1.3/1.4) TRUE/FALSE														
TEXTNOTE_CHANGES	Com a especificação deste parâmetro, os atributos de segurança de PDF para Montagem de documento, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como verdadeiro/falso.														
CHANGES	(Pdf 1.3/1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de														

	segurança de PDF para Montagem de documento, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como verdadeiro/falso.
FILL	(Pdf 1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de segurança de PDF para Montagem de documento, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como verdadeiro/falso.
COPY	(Pdf 1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de segurança de PDF para Montagem de documento, Conteúdo copiado, Conteúdo copiado para acessibilidade, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como verdadeiro/falso.
DIGITAL_COPY	(Pdf 1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de segurança de PDF para Montagem de documento, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como verdadeiro/falso.
ASSEMBLE	(Pdf 1.4) TRUE/FALSE A montagem de um ou mais documentos não é permitida. Com a especificação deste parâmetro, os atributos de segurança de PDF para Montagem de documento, Extração de página, Comentário, Assinatura e Criar modelo serão definidos como verdadeiro/falso.
ACCESSIBILITY	(Pdf 1.4) TRUE/FALSE Com a especificação deste parâmetro, os atributos de segurança de PDF para Conteúdo copiado, Conteúdo copiado para acessibilidade, Extração de página serão definidos como verdadeiro/falso.
DUPLEX_ADDPAGE	TRUE: Esta opção especifica se uma página vazia deve ser adicionada a um documento duplex com um número

	<p>ímpar de páginas.</p> <p>FALSE: Configuração padrão. Esta opção define o modo simplex para a última página de um documento duplex com um número ímpar de páginas.</p>
DUPLEX_OPTIMIZE	<p>TRUE/FALSE</p> <p>Quando verdadeiro, as configurações de duplex são definidas apenas no fluxo de saída, quando o modo duplex muda.</p>
MAP_COLOR	<p>TRUE/FALSE</p> <p>Este parâmetro é usado para definir se o mapeamento de cores da definição AFP-HighlightColor será executado ou não. As definições de cores individuais podem ser especificadas como MapColor_1 até MapColor_n.</p> <p><b>Exemplo:</b></p> <pre>Map_Color=TRUE,"MapColor_1=-1,20,black","MapColor_2=0,20,red"</pre> <p><b>Observação:</b></p> <p>Como os parâmetros MapColor contêm valores separados por vírgulas, a string completa deve estar entre aspas.</p> <p><b>A estrutura é a seguinte:</b></p> <pre>MapColor_???= HSB-Color,Threshold,ColorName</pre> <p>HSB-Color=Ângulo da cor HSB</p> <p>Threshold= Valor limite da cor HSB</p> <p>ColorName= Nome da cor como definido no perfil AFP, em HighlightColorList.</p> <p><b>O exemplo acima requer estas definições no perfil AFP:</b></p> <pre>&lt;/highlightcolorlist&gt;   &lt;highlightcolor name='red' type='OCA' number='0x0002' foreground-rgb="#00FF0000" /&gt;   &lt;highlightcolor name='black' type='OCA'</pre>

	<pre>number='0x0008' foreground-rgb="#00000000" /&gt; &lt;/highlightcolorlist&gt;  e  &lt;optimizepa value="never"/&gt; &lt;color value="FALSE" /&gt; &lt;usehighlightcolor value="TRUE" /&gt; &lt;enablebilevelimage value="TRUE" /&gt;</pre>
SHOW_BOOKMARKS	<p>TRUE: Os bookmarks são mostrados quando o arquivo PDF é aberto.</p> <p>FALSE: Os bookmarks não são mostrados quando o arquivo PDF é aberto.</p>
OVERLAYS	<p>TRUE: São criadas OVERLAYS.</p> <p>FALSE: Padrão. Não são criadas OVERLAYS.</p>
MERGEFONTS	<p>TRUE: Padrão</p> <p>FALSE</p>
MERGE_SPLITWORDS	<p>Ativa e desativa a otimização da saída PCL.</p> <p>Defina esta opção como FALSE se você tiver problemas de espaçamento de caracteres em um PCL gerado.</p> <p>A configuração padrão é TRUE.</p> <p>TRUE: Padrão, a otimização da saída PCL é ativada.</p> <p>FALSE: A otimização da saída PCL é desativada.</p>

**Valor de retorno**

-

**Script de exemplo OdinFileConvert: 1 arquivo**

```
; get FileConverter component
OdinFileConverter = ModusProcess.GetComponent("OdinFileConvert")
;
StringListAdd("options", "USER_PASSWORD=ymM/Mgjelzq/rdPdf6q6rg==")
StringListAdd("options", "OWNER_PASSWORD=uiuCKFZ+QE1VHSQAA/xQQQ==")
StringListAdd("options", "LINEARIZED=true")
StringListAdd("options", "TITLE=document-title")
StringListAdd("options", "SUBJECT=document-subject")
StringListAdd("options", "AUTHOR=JKlinger")
```

```

StringListAdd("options", "KEYWORDS=document-keywords")
StringListAdd("options", "PRINT=true")
StringListAdd("options", "TEXTNOTE_CHANGES=false")
StringListAdd("options", "CHANGES=false")
StringListAdd("options", "FILL=false")
StringListAdd("options", "COPY=true")
StringListAdd("options", "DIGITAL_COPY=false")
StringListAdd("options", "ASSEMBLE=false")
StringListAdd("options", "ACCESSIBILITY=false")
;
options = StringListGetCommaText("options")

; convert single XPS file to PCL
OdinFileConverter.OutputFormat = "pcl"
OdinFileConverter.ConvertSingleFile("c:\temp\Document.xps",
"c:\temp\output.pcl", false, options)
;
; convert single XPS file to Postscript
OdinFileConverter.OutputFormat = "pos"
OdinFileConverter.ConvertSingleFile("c:\temp\Document.xps",
"c:\temp\output.pos", false, options)
;
; convert single XPS file to PDF
OdinFileConverter.OutputFormat = "pdf"

OdinFileConverter.ConvertSingleFile("c:\temp\Document.xps",
"c:\temp\output.pdf", false, options)

```

### Script de exemplo OdinFileConvert: 1-n arquivos

```

Navegação: Processos > Componentes do designer de processo > Odin >
OdinFileConvert > Métodos de OdinFileConvert >
Script de exemplo OdinFileConvert: 1-n arquivos

```

```

;
; get component for converting
OdinFileConverter = ModusProcess.GetComponent("OdinFileConvert")
;
; Start batch conversion
OdinFileConverter.OutputFormat = "pdf"
OdinFileConverter.Start("c:\temp\output.pdf")
;
; Load document #1 (2 pages)
document = OdinFileConverter.LoadDocument("c:\temp\TestDocument1.xps")
;
; Create bookmarks on both pages and output an additional text

```

```

document.EditPage(0)
document.AddBookmark("doc1", "", "Document 1")
document.AddBookmark("doc1_page1", "doc1", "Page 1")
document.SetFont("Tahoma", 16)
document.TextOut(20.5, 10.5, "Document#1 Page#1", false, false)
document.Post()
;
document.EditPage(1)
document.AddBookmark("doc1_page2", "doc1", "Page 2", 100, 100)
document.TextOut(20.5, 10.5, "Document#1 Page#2", false, false)
document.Post()
;
; Insert document in the output file
OdinFileConverter.AddDocument(document)
;
; Document #2: as test inserting a document with AddFile in between, no
bookmarks added
OdinFileConverter.AddFile("c:\temp\TestDocument2.xps")
;
; Add document #3
document = OdinFileConverter.LoadDocument("c:\temp\TestDocument3.xps")
;
; Create bookmarks on both pages and output an additional text
document.EditPage(0)
document.AddBookmark("doc3", "", "Dokument 3")
document.AddBookmark("doc3_page1", "doc3", "Seite 1")
document.SetFont("Tahoma", 16)
document.TextOut(20,5, 10,5, "Document#3 Page#1", false, false)
document.Post()
;
document.EditPage(1)
document.AddBookmark("doc3_page2", "doc3", "Seite 2", 100, 100)
document.TextOut(20.5, 10.5, "Document#3 Page#2", false, false)
document.Post()
;
; Insert document in the output file
OdinFileConverter.AddDocument(document)
;
; Finish batch conversion
OdinFileConverter.End()

```

### Script de exemplo OdinFileConvert: xDoc em XPS

```

; get FileConverter component
OdinFileConverter = ModusProcess.GetComponent("OdinFileConvert")
;
; convert an xDoc file (DocX) to XPS

```

```
OdinFileConverter.ConvertXDocToXps("c:\temp\TestDocument.docx",
"c:\temp\output.xps")
```

## ***Converter Doc em Docx***

### **ConvertDocToDocX**

Esse método pode ser invocado repetidamente.

#### **Sintaxe**

ConvertDocToDocX(string docFileName, string docXFilename)

#### **Parâmetro**

Parâmetro	Significado
docFileName	Arquivo Doc
docXFilename	Arquivo Docx

#### **Valor de retorno**

-

### **EndDocConvert**

Encerra a conversão.

#### **Sintaxe**

EndDocConvert()

#### **Parâmetro**

-

#### **Valor de retorno**

-

### **StartDocConvert**

Inicializa o Word para conversão.

#### **Sintaxe**

StartDocConvert(int wordRestart, bool showWord)

#### **Parâmetro**

Parâmetro	Significado
-----------	-------------

wordRestart	Define o número de operações de conversão que o Word realiza antes de ser reiniciado automaticamente.
showWord	Define se o Word fica visível durante a operação.

**Valor de retorno**

-

**OdinFilterManager*****Informações gerais***

Esse componente só está disponível quando o Odin foi licenciado corretamente.

Esse componente agrupa conjuntos de dados de trabalho para pilhas ou envelopes para processamento adicional.

O componente executa um script em seu contexto para fornecer os objetos específicos à tarefa necessários.

***Propriedades de OdinFilterManager***

Propriedade	Significado
DB Alias	Define o banco de dados Odin no qual o componente irá executar suas ações.
FilterScript	Define o script a ser executado no contexto do componente de filtro. O script deve ter o atributo 'OdinFilter' definido como seu contexto.

***Métodos de OdinFilterManager*****Executar**

Executa o script atribuído à propriedade 'Filter Script'.

**Sintaxe**

```
Run()
```

**Parâmetro**

-

**Valor de retorno**

-

**Script de exemplo OdinFilterManager**



```
OdinFilter = ModusProcess.GetComponent("OdinFilterManager")
OdinFilter.Run()
```

## OdinPrint

### **Informações gerais**

Esse componente permite que os arquivos de spool de trabalhos da pilha sejam saída para uma impressora física (impressora online).

Os trabalhos de uma única pilha podem ser atribuídos a impressoras diferentes.

A impressora e formulário a serem usados são recuperados de um conjunto de dados de trabalho individual. Tanto a impressora especificada quanto os objetos de formulário devem existir no repositório.

Eventos específicos da impressora são executados de acordo com as etapas específicas do processo executadas durante a saída do trabalho.

### **O componente suporta os seguintes cenários:**

- Processamento de trabalhos de uma pilha específica.
- Processamento de pilhas determinado por uma instrução de seleção configurável.

### **Propriedades de OdinPrint**

Propriedade	Significado
DB Alias	Define o banco de dados Odin no qual o componente irá executar suas ações.
IsRetryAble	Um enumerador que especifica se a pilha pode ser impressa novamente no caso de um erro.  <b>Valores possíveis:</b>  OdinRetryOptions.NotRetryAble: A pilha é definida como bloqueada no caso de um erro e não tenta imprimir novamente.  OdinRetryOptions.RetryAbleContinue: No caso de um erro, a pilha será redefinida para imprimir novamente. Somente trabalhos ou pilhas com falha ainda não impressos serão processados.  OdinRetryOptions.RetryAbleRestart: No caso de um erro, toda a pilha será redefinida para imprimir novamente. Todos os trabalhos da pilha são processados.
MaxRetryCount	Número máximo de novas tentativas antes de a pilha ser definida com o status <b>Bloqueado</b> .
NextStackType	Especifica o próximo tipo de pilha a ser definido após a execução com êxito e a chamada do método <b>End</b> .  A lista suspensa exibe todos os tipos de pilha possíveis.  Quaisquer tipos de pilha adicionais devem ser definidos no arquivo

<b>ModusSuite.Common.PropertyEditors.dll.config.</b>		
<b>Tipos de pilha padrão</b>		
0	Imprimir	A pilha está pronta para ser impressa em uma impressora online ou de email (impressão online)
10	PortoOptimization	A pilha está pronta para IT Franking (sem suporte no momento)
20	Streaming	A pilha está pronta para fluxo.
30	Converter	A pilha está pronta para ser convertida no formato de saída apropriado.
40	StreamPrint	A pilha (fluxo) está pronta para impressão em lote em uma impressora terceirizada.
50	Arquivar	A pilha está pronta para arquivamento.
99	Concluir	O processamento da pilha foi concluído, nenhuma ação adicional.
RetryPause	Tempo em minutos antes desta pilha ser impressa novamente ou redefinida por meio do componente OdinDbTasks.	
StackSelect	Define a instrução selecionada para detectar a próxima pilha a ser processada. Ou insira a seleção de instrução diretamente ou consulte uma tabela de pesquisa (por exemplo, \$OdinSQL.PrintStackSelect\$).	

### **Eventos de OdinPrint**

<b>Evento</b>	<b>Significado</b>
OnAfterStack	<p>O script especificado é executado após o processamento da pilha, enquanto chama o método 'End'.</p> <p>O evento corresponde ao evento AfterStack de uma impressora no modus 4.</p> <p><b>Contexto disponível:</b></p> <p><a href="#">Contexto OdinCustomProcess</a></p>
OnBeforeStack	<p>O script especificado é ao chamar o método Start(), antes de executar o processamento de pilha.</p> <p>O evento corresponde ao evento BeforeStack de uma impressora no modus 4.</p>

Contexto disponível:

[Contexto OdinCustomProcess](#)

## ***Métodos de OdinPrint***

### **AddBatchPrintJob**

Adiciona um trabalho à lista interna.

Uma exceção será lançada se a impressão em lote não estiver inicializada na impressora especificada no trabalho, ou se o trabalho for adicionado em redundância.

### **Sintaxe**

AddBatchPrintJob(job)

### **Parâmetro**

Parâmetro	Significado
job	Trabalho a ser inserido

### **Valor de retorno**

-

### **BatchJobCount**

Retorna o número atual de trabalhos de impressão adicionados a uma impressora específica pelo método **AddBatchPrintJob**.

Se nenhum número de impressora for especificado, retorna o número total de trabalhos de impressora adicionados a todas as impressões em lote inicializadas.

### **Sintaxe**

BatchJobCount()

BatchJobCount(string printerName)

### **Parâmetro**

Parâmetro	Significado
printerName	Nome da impressora

### **Valor de retorno**

Número de trabalhos.

**CurrentPrinter**

Retorna o nome da impressora executando, no momento, um trabalho de impressão em lote.

**Sintaxe**

CurrentPrinter()

**Parâmetro**

-

**Valor de retorno**

Nome da impressora

**Término**

Finaliza o processamento do componente, define a pilha para o próximo tipo de pilha e, se relevante, executa um script atribuído ao evento 'OnAfterStack'.

**Os seguintes valores são definidos:**

ODS\_Print\_End: Data/hora atual

ODS\_Delete\_At: Data/hora atual + tempo de espera da pilha

**Se o processamento for finalizado corretamente:**

ODS\_PrintState: 2 (OdinPrintState.Ok)

ODS\_State: **NextStackState** do componente

ODS\_StackType: **NextStackState** do componente

**Em erro:**

ODS\_PrintState: 3 (OdinPrintState.Error)

ODS\_State: 3 (OdinStackState.Locked)

ODS\_MessageText: Mensagem de erro

**Sintaxe**

End()

End(string errorMessage)

**Parâmetro**

Parâmetro	Significado
errorMessage	Mensagem de erro

**Valor de retorno**

-

**ExecuteBatchPrint**

Executa todos os trabalhos de lote inicializados e imprime os trabalhos na impressora correspondente.

**Sintaxe**

ExecuteBatchPrint()

**Parâmetro**

-

**Valor de retorno**

True: impressão com êxito

False: erro ao imprimir

**GetCurrentStackRow**

[▶ Mostrar tudo](#)

Após chamar Start() e antes de chamar End(), o método retorna um objeto cuja interface fornece acesso de leitura ao conjunto de dados atual da pilha.

**Sintaxe**

GetCurrentStackRow()

**Parâmetro**

-

**Valor de retorno**

Interface de objeto que fornece acesso de leitura ao conjunto de dados atual da pilha.

**Os seguintes métodos estão disponíveis:**

GetValue

Retorna o valor da coluna dada.

**Sintaxe**

GetValue(string columnName)

Parâmetro	Significado
columnName	Nome da coluna

**Valor de retorno**

Valor da coluna

Se o tipo de dados da coluna do banco de dados for DATE, Datetime ou qualquer tipo de dados válido, a data lida geralmente será retornada no formato MM/DD/AAAA hh:mm:ss.

**Exemplo**

```
GetObject("DateVar", "MLDate")
...
StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT")
;Converter a data no formato desejado com a ajuda do objeto MLDate
IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy hh:mm:ss", StringDateVar )
v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy hh:mm:ss ", IntDateVar)
```

GetFieldNameEnumerator

Retorna um enumerador.

### Sintaxe

GetEnvelopeEnumerator()

### Parâmetro

-

### Valor de retorno

Enumerador com estes métodos:

Método	Função
Current	Retorna o objeto atual.
MoveNext	Movê a posição até o próximo objeto a ser processado.
Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.

### GetNextStack

Determina a próxima pilha a ser convertida, de acordo com o SQL correspondente e retorna sua ID.

### Sintaxe

GetNextStack()

### Parâmetro

-

### Valor de retorno

ID de pilha

### GetPrinterEnumerator

Retorna um enumerador para as impressoras usadas em uma pilha.

No carregamento, as impressoras usadas em uma pilha são classificadas na seguinte ordem:

- Tipo de impressora (impressora online primeiro, depois impressora de email)
- Prioridade de impressora para tipos idênticos de impressora

- Nome da impressora para prioridades idênticas

### Sintaxe

GetPrinterEnumerator()

### Parâmetro

-

### Valor de retorno

Enumerador com estes métodos:

Método	Significado
Current	Retorna o nome da impressora atual.
MoveNext	Retorna TRUE quando é possível navegar para a próxima impressora. Retorna FALSE quando o fim da coleta é passado.
Reset	Redefine o enumerador de volta para o início do primeiro elemento.

### GetPrintJobEnumerator

Retorna um enumerador para os trabalhos de uma impressora.

O enumerador usa **Current** para devolver um trabalho.

A função GetValue() recupera dados de trabalho.

### Sintaxe

GetPrintJobEnumerator(string printerName)

### Parâmetro

Parâmetro	Significado
printerName	Nome da impressora

### Valor de retorno

Enumerador com estes métodos:

Método	Significado
Current	Retorna o nome da impressora atual.

MoveNext	Retorna TRUE quando é possível navegar para a próxima impressora. Retorna FALSE quando o fim da coleta é passado.
Reset	Redefine o enumerador de volta para o início do primeiro elemento.

### GetMailEnumerator

Retorna um enumerador para o endereço de email nos trabalhos de uma impressora.

O enumerador usa 'Current' para recuperar um endereço de email, ou seja, com a ajuda de um iterador, ele pode iterar por vários destinatários de email nos trabalhos.

### Sintaxe

GetMailEnumerator(string printerName)

### Parâmetro

Parâmetro	Significado
printerName	Nome da impressora

### Valor de retorno

Enumerador com estes métodos:

Método	Significado
Current	Retorna o nome da impressora atual.
MoveNext	Retorna TRUE quando é possível navegar para a próxima impressora. Retorna FALSE quando o fim da coleta é passado.
Reset	Redefine o enumerador de volta para o início do primeiro elemento.

### GetMailJobEnumerator

Retorna um enumerador para os trabalhos de uma impressora com um destinatário de email específico.

O enumerador usa **Current** para devolver um trabalho.

A função GetValue() recupera dados de trabalho.

### Sintaxe

GetMailJobEnumerator(string printerName, string email)

### Parâmetro



Parâmetro	Significado
printerName	Nome da impressora
email	Endereço de email

### Valor de retorno

Enumerador com estes métodos:

Método	Significado
Current	Retorna o nome da impressora atual.
MoveNext	Retorna TRUE quando é possível navegar para a próxima impressora. Retorna FALSE quando o fim da coleta é passado.
Reset	Redefine o enumerador de volta para o início do primeiro elemento.

### InitializeBatchPrint

Inicializa impressão em lote em uma impressora especificada.

Somente uma impressão em lote pode ser inicializada por impressora. Uma segunda inicialização irá gerar uma exceção.

### Sintaxe

InitializeBatchPrint(bool useOneSpoolfile)

InitializeBatchPrint(string printerName, bool useOneSpoolfile)

### Parâmetro

Parâmetro	Significado
printerName	Nome da impressora
useOneSpoolfile	TRUE: Cria um único arquivo de spool de impressora para todos os trabalhos inseridos. FALSE: Os trabalhos são impressos em sucessão.

### Valor de retorno

-

### IsEmailPrinter

Verifica se a impressora em questão é uma impressora de e-mail.

**Sintaxe**

IsEmailPrinter(string printerName)

**Parâmetro**

Parâmetro	Significado
printerName	Nome da impressora

**Valor de retorno**

True: A impressora é uma impressora de e-mail.

False: A impressora não é uma impressora de e-mail.

**IsPrinterActive**

Verifica se uma impressora está ativada.

**Sintaxe**

IsPrinterActive(string printerName)

**Parâmetro**

Parâmetro	Significado
printerName	Nome da impressora

**Valor de retorno**

True: a impressora está ativada.

False: impressora desativada devido a um erro.

**PrintJob**

Imprime um trabalho em uma impressora. O nome da impressora é tirado dos dados do trabalho específico (ODJ\_Printer\_Name).

**Sintaxe**

PrintJob(IOdinReadOnlyDataRow job)

**Parâmetro**

Parâmetro	Significado
job	Trabalho a ser impresso

**Valor de retorno**

True: Impressão com êxito

False: Erro na impressão. O erro é inserido automaticamente no registro da pilha correspondente e pode ser acessado por meio da propriedade LastErrorMessage.

### Iniciar

Carrega e inicializa a pilha determinada e executa um script atribuído ao evento "OnBeforeStack".

A pilha deve ser do tipo correspondente ao requisitado pelo componente. O status deve ser definido como "Aguardar".

**Após ser carregada com êxito, a pilha é bloqueada e os status são definidos de acordo:**

ODS\_LockBy: Nome do usuário atual.

ODS\_State: 1 (OdinStackState.InWork)

ODS\_PrintState: 1 (OdinPrintState.InWork)

ODS\_Print\_Start: Data/hora atuais

ODS\_MessageText: String vazia

### Sintaxe

Start(string stackId)

### Parâmetro

Parâmetro	Significado
stackId	ID de pilha

### Valor de retorno

-

### Script de exemplo OdinPrint

```

; define whether print all jobs in 1 spool file or print separately
useOneSpoolfile = true
;
; initialize print components
OdinPrint = ModusProcess.GetComponent("OdinPrint")
;
; determine next stack
stackID = OdinPrint.GetNextStack()
;
; was a stack found ?
if (stackID <> "")
;
    ; Yes, now initialize stack print

```

```

OdinPrint.Start(stackID)
;
; get a printer enumerator and print the jobs for each printer
PrinterEnumerator = OdinPrint.GetPrinterEnumerator()
;
; are there more printers in the stack for which jobs are to be
processed?
nextPrinter = PrinterEnumerator.MoveNext()
;
while (nextPrinter = true)
    ; yes, now print all jobs for this printer in one spool file
    printerName = PrinterEnumerator.Current
    ;
    ; Is the printer active?
    if (OdinPrint.IsPrinterActive(printerName))
        ; yes, is it an Email Printer.
        if (OdinPrint.IsEmailPrinter(printerName))
            ;
            ; yes, now initialize batch print for the Email Printer
            OdinPrint.InitializeBatchPrint(printerName)
            ;
            ; get an email enumerator for this printer
            MailEnumerator = OdinPrint.GetMailEnumerator(printerName)
            ;
            ; any more emails for this printer?
            nextEmail = MailEnumerator.MoveNext()
            ;
            while (nextEmail = true and
OdinPrint.IsPrinterActive(printerName))
                ; get email address from enumerator
                emailAdr = MailEnumerator.Current
                ;
                ; get an mail job enumerator for this email address
                MailJobEnumerator =
OdinPrint.GetMailJobEnumerator(printerName, emailAdr)
                ;
                ; and print all jobs for this Email Printer and for this
email address
                nextMailJob = MailJobEnumerator.MoveNext()
                ;
                while (nextMailJob = true)
                    ; get current job from enumerator
                    job = MailJobEnumerator.Current
                    ; as a test get job ID
                    jobId = job.GetValue("ODJ_ID")
                    ;

```

```

        ; insert job as BatchPrintJob
        OdinPrint.AddBatchPrintJob(job)
        ;
        ; process next Job
        nextMailJob = MailJobEnumerator.MoveNext()
end-while
;
; any BatchPrintJobs found?
; If an empty string passed here, the number of jobs
; for *all* initialized BatchPrints is returned.
printed = true
;
if (OdinPrint.BatchJobCount(printerName) > 0)
    ; yes, now execute print for this printer
    printed = OdinPrint.ExecuteBatchPrint(printerName)
    ;
    if (printed = false)
        Protocol("Error in print to printer '{0}':", 1,
printerName)
    end-if
    ; This method executes all initialized BatchPrints in
the order
    ; they were initialized.
    ;OdinPrint.ExecuteBatchPrint()
end-if
;
; If printed successfully, check if there are more emails
for the printer
; to process.
if (printed = true)
    nextEmail = MailEnumerator.MoveNext()
end-if
end-while
else
    ; Initialize Batchprint for this Online-Printer
    ; and print all Jobs as one spool file!
    OdinPrint.InitializeBatchPrint(printerName, useOneSpoolfile)
    ; This call sets useOneSpoolfile to TRUE !
    ;OdinPrint.InitializeBatchPrint(printerName)
    ;
    ; get a job enumerator for this printer
    JobEnumerator = OdinPrint.GetPrintJobEnumerator(printerName)
    ;
    ; and insert or print all jobs
    nextJob = JobEnumerator.MoveNext()
    ;

```

```

while (nextJob = true)
    ; get current job from enumerator
    job = JobEnumerator.Current
    ; as a test get job ID
    jobId = job.GetValue("ODJ_ID")
    ;
    ; This method enables you to print the individual job
    ;OdinPrint.PrintJob(job)
    ;
    ; insert job as BatchPrintJob
    OdinPrint.AddBatchPrintJob(job)
    ;
    ; process next Job
    nextJob = JobEnumerator.MoveNext()
    ;
end-while
;
; any BatchPrintJobs found?
; If an empty string passed here, the number of jobs
; for *all* initialized BatchPrints is returned.
;
if (OdinPrint.BatchJobCount(printerName) > 0)
    ;
    ; yes, now execute print for this printer
    printed = OdinPrint.ExecuteBatchPrint(printerName)
    ;
    if (printed = false)
        Protocol("Error when printing at printer '{0}':", 1,
printerName)
    end-if
    ; This method executes all initialized BatchPrints in the
order
    ; they were initialized.
    ;OdinPrint.ExecuteBatchPrint()
end-if
end-if
end-if
;
; are there any more printers that need to be processed?
nextPrinter = PrinterEnumerator.MoveNext()
end-while
;
; finish stack print
OdinPrint.End()
;

```

```
end-if
```

## OdinProcessImport

### Informações gerais

Use esse componente para importar dados de processamento de impressão para as tabelas de banco de dados Odin para processamento adicional.

### Propriedades de OdinProcessImport

Propriedade	Significado
Alias de BD	Define o banco de dados Odin no qual o componente irá executar suas ações.
DocumentBaseDestinationDirectory	<p>Define o diretório em que os arquivos de documento são armazenados na importação. Esta pode ser uma instrução de caminho completa ou pode se referir a uma entrada em uma tabela de pesquisa.</p> <p>Se, por exemplo, a entrada <b>ImportPath</b> na tabela de pesquisa <b>Odin</b> for o caminho de importação desejado, ele é especificado como <code>\$Odin.ImportPath\$</code>.</p> <p><b>A seguinte estrutura de pastas é criada abaixo do diretório definido:</b></p> <p>[Carimbo de data/hora de quando o componente de importação foi iniciado]</p> <p style="padding-left: 40px;">[Data de quando o diretório foi criado]</p> <p style="padding-left: 80px;">[Hora do dia em que o diretório foi criado]</p> <p style="padding-left: 120px;">[Pasta com o nome 1000, 2000, 3000]</p> <p>No exemplo a seguir, o <b>DocumentBaseDestinationDirectory</b> é definido como <b>odinImport</b>.</p> <p>A seguinte estrutura de pastas foi criada na importação do primeiro processo:</p>  <p>O componente de importação move os primeiros 1000</p>

	<p>documentos de processos para a pasta nomeada como <b>1000</b>.</p> <p>Posteriormente, uma nova pasta é criada.</p> <p>Se o dia atual e a hora ainda forem os mesmos, a nova pasta é criada abaixo da pasta <b>10</b> com o nome <b>2000</b>.</p> <p>Se não, uma pasta de um novo dia e/ou hora com uma subpasta <b>1000</b> é criada.</p> <p>Os arquivos de documento armazenados na estrutura de pastas são excluídos quando o componente <b>OdinDbTasks</b> exclui conjuntos de dados de pilha finalizados (processados).</p> <p>Se os diretórios vazios resultarem da exclusão do processo, esses também são excluídos.</p>
OdinSystemOID	<p>Especifica o sistema OID do sistema no qual as impressoras e os formulários Odin serão carregados.</p> <p>Se não estiverem definidos, os objetos Odin são carregados do sistema do qual o processo de execução do componente foi carregado.</p>
Synchronize	<p>Se mais de um componente OdinImport usar o mesmo diretório de importação simultaneamente, essa propriedade deverá ser definida para True.</p> <p>Isso garante que um arquivo de processo nunca seja processado por mais de uma instância do OdinImport. Se a propriedade for False, nenhuma sincronização de medidas é feita.</p>

## ***Métodos de OdinProcessImport***

### **ImportProcessFile**

Importa um arquivo de processo.

### **Sintaxe**

ImportProcessFile(ProcessFile, DocumentFileDirectory)

### **Parâmetro**

Parâmetro	Significado
ProcessFile	O arquivo a ser importado.



DocumentFileDirectory	O caminho do diretório em que os arquivos de documento relacionados ao arquivo XML de processo estão armazenados.
-----------------------	---

**Valor de retorno**

ID do processo: ID do processo importado.

**Script de exemplo**

```
OdinImport = ModusProcess.GetComponent("OdinProcessImport")
ImportedProcessId =
OdinImport.ImportProcessFile("c:\OdinImport\TestProcess.xml",
"c:\OdinImport")
```

**SearchAndImportProcessFile**

Pesquisa por arquivos com a extensão \*.xml no diretório especificado e os importa.

**Sintaxe**

```
SearchAndImportProcessFile(ProcessFileDirectory, DocumentBaseDirectory)
```

**Parâmetro**

Parâmetro	Significado
ProcessFileDirectory	O diretório no qual os arquivos de processo devem ser pesquisados.
DocumentBaseDirectory	<p>Caminho do diretório no qual os arquivos de documento de processo serão encontrados nas respectivas subpastas.</p> <p>A associação entre um arquivo de processo e seu diretório de documentos é feita usando o mesmo nome.</p>

**Valor de retorno**

ID de processo do processo importado ou uma string vazia se nenhum arquivo de processo for encontrado.

**Script de exemplo**

```
OdinImport = ModusProcess.GetComponent("OdinProcessImport")
ImportedProcessId = OdinImport.SearchAndImportProcessFile("d:\OdinImport",
"d:\OdinImport")
```

**Script de exemplo OdinProcessImport**

```
OdinImport = ModusProcess.GetComponent("OdinProcessImport")
;
processId = OdinImport.SearchAndImportProcessFile(SYSTEM.ImportDirectory,
SYSTEM.ImportDirectory)
if (processId <> "")
```

```

;Process data imported... Repeat the loop
SYSTEM.ExecuteLoop = true
else
;no data to import... no loop necessary
SYSTEM.ExecuteLoop = false
end-if

```

## OdinRendition

### **Informações gerais**

Componente para conversão de arquivos de documentos dos trabalhos para o formato de arquivo de spool do Odin (formato XPS).

O componente pode gerenciar diferentes formatos de documento, por exemplo, docx, PDF e TIFF.

O formato XPS é usado para criar fluxos de impressão.

Os documentos criados com ModusOne podem ser impressos diretamente para uma impressora, enquanto formatos de terceiros como PDF iniciam o aplicativo correspondente para a operação de impressão. O arquivo de spool criado também é salvo, além do documento original.

#### **O componente suporta os seguintes cenários:**

- Processamento de trabalhos de um processo específico (pilha/impressão online)
- Processamento de trabalhos determinados por uma instrução de seleção pré-configurada

#### **Os formatos de entrada suportados são:**

- arquivos docx criados pelo ModusOne
- arquivos docx estáticos (convertidos em XPS pelo Word, para que o layout original seja 100% garantido)
- PDF
- TIFF
- XPS

### **Propriedades de OdinRendition**

Propriedade	Significado
Alias de BD	Define o banco de dados Odin no qual o componente irá executar suas ações.
JobOrder	Define o campos a serem usados pela classificação atual. Os seguintes campos são definidos como padrão: ODJ_BUNDLE_SEQ ASC, ODJ_SEQUENCE ASC, ODJ_ID ASC
ProcessSelect	Especifica a instrução selecionada executada para detectar o processo a ser manipulado. Essa propriedade não precisa ser definida, se o método

	<p><b>CreateRenditionForProcess</b> for chamado.</p> <p>A instrução selecionada deve especificar um bloqueio de atualização para permitir uma implantação paralela de várias instâncias de apresentação.</p> <p>Para servidores MSSQL, a instrução selecionada deve conter <b>With (Updlock)</b>.</p> <p>Para servidores Oracle, a instrução selecionada deve conter <b>for update</b>.</p>																
<p>StackType</p>	<p>Define o tipo de pilha a ser definido por pilhas de impressão direta se impressões diretas forem encontradas e uma nova pilha for criada.</p> <p>O valor padrão é StackType = 0 (Print).</p> <p><b>Os seguintes tipos de pilhas podem ser definidos:</b></p> <table border="1" data-bbox="402 743 875 1352"> <tr> <td>Print</td> <td>0</td> </tr> <tr> <td>Archive</td> <td>50</td> </tr> <tr> <td>Transfer</td> <td>60</td> </tr> <tr> <td>Finished</td> <td>                     99                      Quando o tipo de pilha é definido como Finished, o                     <ul style="list-style-type: none"> <li>status (ODS_STATE) é definido como <b>Em suspenso</b> e os</li> <li>Dados excluídos (ODS_DELETE_AT) definidos como <b>AGORA + Tempo de espera</b></li> </ul> </td> </tr> </table> <p><b>Os seguintes tipos de pilhas não podem ser definidos e resultarão em um erro se forem definidos em um script:</b></p> <table border="1" data-bbox="402 1495 662 1780"> <tr> <td>PortoOptimization</td> <td>10</td> </tr> <tr> <td>Streaming</td> <td>20</td> </tr> <tr> <td>Convert</td> <td>30</td> </tr> <tr> <td>StreamPrint</td> <td>40</td> </tr> </table>	Print	0	Archive	50	Transfer	60	Finished	99 Quando o tipo de pilha é definido como Finished, o <ul style="list-style-type: none"> <li>status (ODS_STATE) é definido como <b>Em suspenso</b> e os</li> <li>Dados excluídos (ODS_DELETE_AT) definidos como <b>AGORA + Tempo de espera</b></li> </ul>	PortoOptimization	10	Streaming	20	Convert	30	StreamPrint	40
Print	0																
Archive	50																
Transfer	60																
Finished	99 Quando o tipo de pilha é definido como Finished, o <ul style="list-style-type: none"> <li>status (ODS_STATE) é definido como <b>Em suspenso</b> e os</li> <li>Dados excluídos (ODS_DELETE_AT) definidos como <b>AGORA + Tempo de espera</b></li> </ul>																
PortoOptimization	10																
Streaming	20																
Convert	30																
StreamPrint	40																

## Eventos de OdinRendition

Propriedade	Significado
OnAfterJobScript	<p>Esse evento é chamado após uma apresentação de trabalho ter sido criado.</p> <p>O nome do arquivo de apresentação são disponíveis no conjunto de dados do trabalho (ODJ_RENDITION).</p> <p><b>Contexto disponível:</b>  <a href="#">Context OdinRendition</a></p>
OnBeforeJobScript	<p>Esse evento é executado antes de um trabalho ser processado.</p> <p>Usando o atributo <b>CurrentJobRecord</b> e os métodos <code>GetValue()</code> ou <code>SetValue()</code>, um script atribuído aqui tem acesso á leitura e á gravação das colunas do conjunto de dados atual.</p> <p>Qualquer alteração feita para o conjunto de dados de trabalho é levado em consideração durante a criação do arquivo de apresentação e é salvo no banco de dados se a apresentação de trabalhar tiver êxito.</p> <p><b>Contexto disponível:</b>  <a href="#">Context OdinRendition</a></p> <p><b>Script de exemplo</b></p> <p>Os exemplos seguintes verificam se o conjunto de dados de trabalho contém uma ID de objeto de impressora. Se não contiver, uma ID é definida como é necessário durante a criação de uma apresentação.</p> <pre>Printer = CurrentJobRecord.GetValue("ODJ_PRINTER_OID") if (Printer = "")     CurrentJobRecord.SetValue("ODJ_PRINTER_OID", "test-pr-5") end-if</pre>
OnProcessEnd	<p>Esse evento é executado depois que um trabalho tiver sido processado.</p> <p><b>Contexto disponível:</b>            Nenhum</p>
OnProcessStart	<p>Esse evento é executado no início da manipulação do processo.</p> <p><b>Contexto disponível:</b>            Nenhum</p>

## Métodos de OdinRendition

**CreateRenditionForProcess**

Cria arquivos de apresentação (spool) para trabalhos/documentos do processo especificado.

**Sintaxe**

```
CreateRenditionForProcess(processId)
```

**Parâmetro**

Parâmetro	Significado
processId	ID do processo a ser manipulado.

**Valor de retorno**

ID da pilha de impressão direta criada para o processo manipulado.

Uma string vazia é retornada se o processo não tiver uma pilha de impressão direta, pois nenhum documento de impressão direta foi encontrado.

**Script de exemplo**

```
RenditionManager = ModusProcess.GetComponent("OdinRendition")
DirectPrintStackId = RenditionManager.CreateRenditionForProcess("{002A2ACE-29A1-4B3E-A573-5B95ED0EE8B5}")
```

**CreateRenditionForSelectedProcess**

Determina o processo a ser manipulado por meio do SQL de seleção do processo.

Na sequência, o mesmo processamento é executado, como no método **CreateRenditionForProcess**.

A ID do processo a ser manipulado é determinada pela execução da instrução de seleção do processo.

**Sintaxe**

```
CreateRenditionForSelectedProcess()
```

**Parâmetro**

Parâmetro	Significado

**Valor de retorno**

Retorna a ID do processo manipulado ou uma string vazia se nenhum processo foi encontrado.

**Script de exemplo**

```
RenditionManager = ModusProcess.GetComponent("OdinRendition")
;
RenditionManager.ProcessSelect =
```

```
"Select TOP 1 ODP_ID From Odin_Process
where ODP_State = 0 and ODP_Lockby is Null Order By Odp_Create_At Asc"
;
ProcId = RenditionManager.CreateRenditionForSelectedProcess()
```

### Script de exemplo OdinRendition

```
rendition = ModusProcess.GetComponent("OdinRendition")
;rendition.ProcessSelect =
;"select top 1 odp_id from Odin_process with (updlock) where odp_state = 0
and odp_lockby is null " +
;"and odp_bundle_name = 'Invoice' " +
;"order by odp_create_at asc"
;
if (processId <> "")
    ;create XPS files for the imported Process
    directPrintStackId = rendition.CreateRenditionForProcess(processId)
end-if
```

### OdinStream

### Informações gerais

[▶ Mostrar tudo](#)

Este componente só estará disponível quando o Odin tiver sido licenciado corretamente.

Este componente é usado para criar arquivos de fluxo no formato XML, seus arquivos de descrição e os arquivos de log do protocolo correspondente.

### Banco de dados virtual

Cada script tem acesso às tabelas do banco de dados virtual para a pilha (**STACK**), os envelopes (**ENVELOPE**) e os trabalhos do envelope (**JOB**).

Essas tabelas são usadas para estabelecer informações sobre pilha/envelopes/trabalhos. Isso é feito usando a funcionalidade do banco de dados virtual (VirtualDatabase).

### Exemplo de acesso à pilha, envelope e colunas de trabalho.

```
stackID = GetField("STACK", "ODS_ID")
envID = GetField("ENVELOPE", "ODE_ID")
jobID = GetField("JOB", "ODJ_ID")
```

### OMR-Definição

Os códigos OMR são definidos na criação da pilha, por exemplo, em um filtro via SetProperty("OMR\_CODE").

Todas as posições e configurações de tamanho de um código OMR são feitas em mm.

### Os seguintes códigos OMR são suportados atualmente:

- C4\_OMR
- C4EF\_OMR
- K34\_OMR

- K34K\_OMR
- BELL\_OMR
- AS1A\_OMR
- STIELOW\_OMR
- SI68\_OMR
- KERN\_POST\_OMR
- Kern Post 2

As seguintes configurações podem ser definidas para cada OMR:

OMR_POS_X / OMR_POS_Y	Posições iniciais horizontal e vertical para o código OMR no papel.
OMR_LINELENGTH / OMR_LINEHEIGHT	Comprimento e largura das linhas OMR.
OMR_GAP	Espaço entre cada linha
OMR_EDGE	Especifica se o código OMR é impresso na borda longa (=LONG) ou na borda curta (=SHORT) de uma página. Padrão: LONG

Se forem necessárias configurações variáveis para um único tipo OMR, elas podem ser definidas inserindo outro nome de configuração separado por um ponto:

OMR\_TYP.CONFIG

#### Exemplo

`C4_OMR=OMR_POS_X=10,OMR_POS_Y=10,OMR_LineLength=5,OMR_LineHeight=0.5,OMR_Gap=5,OMR_Edge=Long`

`C4_OMR.ShortEdge=OMR_POS_X=10,OMR_POS_Y=10,OMR_LineLength=5,OMR_LineHeight=0,5,OMR_Gap=5,OMR_Edge=Short`

#### Textos adicionais (Extended Text)

Esses textos adicionais são definidos nas tabelas de pesquisa atribuídas à propriedade 'ExtendedText' de um componente de fluxo.

Valores de pesquisas podem ser definidos para páginas de capa de envelopes, pilhas e documentos.

Novamente, essas especificações podem ser ainda mais diferenciadas para layouts de páginas nos modos retrato e paisagem.

ExtendFlyLeafStack_Portrait_Def	Texto adicional para layout no modo retrato para a página de capa da pilha
ExtendFlyLeafStack_Landscape_Def	Texto adicional para layout no modo paisagem para a página de capa da pilha
ExtendFlyLeafEnvelope_Portrait_Def	Texto adicional para layout no modo retrato para a

	página de capa do envelope
ExtendFlyLeafEnvelope_Landscape_Def	Texto adicional para layout no modo paisagem para a página de capa do envelope
ExtendAllPages_Portrait_Def	Texto adicional para layout no modo retrato da página de documento
ExtendAllPages_Landscape_Def	Texto adicional para layout no modo paisagem da página de documento

Até o momento, saída de texto (**Type=Text**), Linhas (**Type=LINE**), Retângulos (**Type=RECTANGLE**) e Imagens (**Type=IMAGE**) são suportadas.

Várias saídas são possíveis enumerando as entradas:

```
ExtendAllPages_Portrait_Def=TYPE=TEXT,XPOS=25.5,YPOS=280.5, FONTSIZE=6,BOLD=N,
Italic=N, FONTNAME=ARIAL, Angle=0, TEXT=" %S. ODS_ID%
```

```
ExtendAllPages_Portrait_Def2=TYPE=TEXT,XPOS=22.5,YPOS=260.5, FONTSIZE=6,BOLD=N
, Italic=Y, FONTNAME=ARIAL, Angle=0, TEXT=2nd text
```

Texto (Type=TEXT)

COLOR	Cor do texto (Padrão: Black)
LAYER	Define a camada de saída. 0 ou primeiro plano: camada de primeiro plano (o desenho é feito por cima do documento original) 1 ou plano de fundo: camada de plano de fundo (o desenho é feito atrás do documento original)
XPOS/YPOS	Especifica a posição do texto X / Y em mm (Padrão: 10 mm)
FONTSIZE	Tamanho da fonte (Padrão: 6)
BOLD	Negrito S/N (Padrão: false)
ITALIC	Itálico S/N (Padrão: false)
FONTNAME	Nome da fonte usada para saída de texto. (Padrão: Arial)
ANGLE	Ângulo do texto. (Padrão: 0)
TEXT	Texto a ser escrito. Se o texto contiver espaços, ele deve ser definido conforme a seguir: "TEXT=Este é um texto com espaços!"



Espaços reservados no texto são incluídos por %  
 Espaços reservados por valores de um banco de dados:  
 %S.Field name%: Campo na tabela da pilha **Odin\_Stack**  
 %E.Field name%: Campo na tabela do envelope **Odin\_Envelope**  
 %J.Field name%: Campo na tabela do trabalho **Odin\_Job**

Além disso, todas as propriedades são acessadas pelo método `GetProperty` do contexto `OdinStream` também podem ser usadas.

### Exemplo

```
ExtendFlyLeafStack_Portrait_Def=TYPE=TEXT,XPOS=25,YPOS=280, FONTSIZE=6, BOLD=N,
Italic=N, FONTNAME=ARIAL, Angle=0, "TEXT=Deckblatt Stapel #S.ODS_ID%"
ExtendFlyLeafEnvelope_Portrait_Def=TYPE=TEXT,XPOS=25,YPOS=280, FONTSIZE=6, BOLD=
=N, Italic=N, FONTNAME=ARIAL, Angle=0, "TEXT=Deckblatt Kuvert #E.ODE_ID%"
ExtendAllPages_Portrait_Def=TYPE=TEXT,XPOS=10,YPOS=260, FONTSIZE=6, BOLD=N, Ital
ic=N, FONTNAME=Arial, Angle=270, TEXT=%S.ODS_ID%/E.ODE_ID%/J.ODJ_ID%
ExtendAllPages_Portrait_Def2=TYPE=TEXT,XPOS=28,YPOS=290, FONTSIZE=6, BOLD=N, Ita
lic=N, FONTNAME=Arial, Angle=0, "TEXT=Envelope##StackEnvelope%/TotalStackEnvelo
pes% - Env_job##EnvelopeJob%/TotalEnvelopeJobs% -
Env_Page##EnvelopePage%/TotalEnvelopePages% Page##JobPage%/TotalJobPages%"
ExtendAllPages_Landscape_Def=TYPE=TEXT,XPOS=39,YPOS=10, FONTSIZE=6, BOLD=N, Ital
ic=N, FONTNAME=Arial, Angle=180, TEXT=%S.ODS_ID%/E.ODE_ID%/J.ODJ_ID%
ExtendAllPages_Landscape_Def2=TYPE=TEXT,XPOS=8,YPOS=28, FONTSIZE=6, BOLD=N, Ital
ic=N, FONTNAME=Arial, Angle=270, "TEXT=Envelope##StackEnvelope%/TotalStackEnvelo
pes% - Env_job##EnvelopeJob%/TotalEnvelopeJobs% -
Env_Page##EnvelopePage%/TotalEnvelopePages% Page##JobPage%/TotalJobPages%"
```

### Linha (Type=LINE)

XPOS/YPOS	Especifica a posição inicial do texto X / Y em mm (Padrão: 10 mm)
XPOS2/YPOS2	Especifica a posição final do texto X / Y em mm (Padrão: 10 mm)
LineWidth	Especifica a espessura da linha em mm (Padrão: 1 mm)
LineStyle	Especifica o estilo da linha (Padrão 0=Solid) 0 ou sólido: Solid 1 ou tracejado: Dashed 2 ou pontilhado: Dotted
LineColor	Cor da linha
LAYER	Define a camada de saída. 0 ou foreground: camada de primeiro plano (o desenho é feito por cima do

documento original)  
 1 ou background: camada de plano de fundo (o desenho é feito atrás do documento original)

**Exemplo:**

```
ExtendAllPages_Portrait_Def3=TYPE=LINE,XPOS=10,YPOS=10,XPOS2=60,YPOS2=60,Line
Width=0,5,LineStyle=2,LineColor=Red,Layer=1
```

```
ExtendAllPages_Portrait_Def3=TYPE=LINE,XPOS=10,YPOS=10,XPOS2=60,YPOS2=60,Line
Width=0,5,LineStyle=dotted,LineColor=Red,Layer=background
```

**Retângulo (Type=RECTANGLE)**

XPOS/YPOS	Especifica a posição X / Y em mm (Padrão: 10 mm)
Espessura/altura	Espessura/altura do retângulo em mm (Padrão: 10 mm)
LineWidth	Especifica a espessura da linha em mm (Padrão: 1 mm)
LineStyle	Especifica o estilo da linha (Padrão: 0=Solid) 0 ou sólido: Solid 1 ou tracejado: Dashed 2 ou pontilhado: Dotted
LineColor	Cor da borda (Padrão: Black)
Preenchimento	True / False Especifica se o retângulo está preenchido com cor (Padrão: false)
FillColor	A cor de preenchimento do retângulo.
LAYER	Define a camada de saída. 0 ou primeiro plano: camada de primeiro plano (o desenho é feito por cima do documento original) 1 ou plano de fundo: camada de plano de fundo (o desenho é feito atrás do documento original)

**Exemplo:**

```
ExtendAllPages_Portrait_Def4=Type=RECTANGLE,XPOS=20,YPOS=20,Width=10,Height=1
0,LineWidth=0,5,LineStyle=1,LineColor=Black,fill=true,fillColor=yellow,Layer=
0
```

```
ExtendAllPages_Portrait_Def4=Type=RECTANGLE,XPOS=20,YPOS=20,Width=10,Height=10,LineWidth=0,5,LineStyle=dashed,LineColor=Black,fill=true,fillColor=yellow,Layer=foreground
```

Imagem (Type=IMAGE)

XPOS/YPOS	Especifica a posição X / Y em mm (Padrão: 10 mm)
Arquivo	Nome do arquivo da imagem a ser pintado.

### Exemplo:

```
ExtendAllPages_Portrait_Def5=Type=IMAGE,XPOS=40,YPOS=40,file=d:\temp\pa_logo.bmp,Layer=1
```

### As cores são definidas como uma cadeia:

#AARRGGBB AA=Alpha-Transparency, RR=Red, GG=Green, BB=Blue

ex.: para Black #00000000, para White#00FFFFFF

Transparência alfa não é suportada durante a conversão.

Se 255 (FF) for especificado para o texto para Transparência alfa, o texto não é transparente, caso contrário a transparência é desenhada na cor correspondente.

Lista de todas as cores que podem ser definidas por nome (essas não são transparentes como FF é definido como Transparência alfa)

AliceBlue: #FFF0F8FF

AntiqueWhite: #FFFAEBD7

Aqua: #FF00FFFF

Aquamarine: #FF7FFFD4

Azure: #FFF0FFFF

Beige: #FFF5F5DC

Bisque: #FFF5E4C4

Black: #FF000000

BlanchedAlmond: #FFF5EBCD

Blue: #FF0000FF

BlueViolet: #FF8A2BE2

Brown: #FFA52A2A

BurlyWood: #FFDEB887

CadetBlue: #FF5F9EA0

Chartreuse: #FF7FFF00

Chocolate: #FFD2691E  
Coral: #FFFF7F50  
CornflowerBlue: #FF6495ED  
Cornsilk: #FFFFFF8DC  
Crimson: #FFDC143C  
Cyan: #FF00FFFF  
DarkBlue: #FF00008B  
Dark Cyan: #FF008B8B  
DarkGoldenrod: #FFB8860B  
DarkGray: #FFA9A9A9  
DarkGreen: #FF006400  
DarkKhaki: #FFBDB76B  
DarkMagenta: #FF8B008B  
DarkOliveGreen: #FF556B2F  
DarkOrange: #FFFF8C00  
DarkOrchid: #FF9932CC  
DarkRed: #FF8B0000  
DarkSalmon: #FFE9967A  
DarkSeaGreen: #FF8FBC8B  
DarkSlateBlue: #FF483D8B  
DarkSlateGray: #FF2F4F4F  
DarkTurquoise: #FF00CED1  
DarkViolet: #FF9400D3  
DeepPink: #FFFF1493  
DeepSkyBlue: #FF00BFFF  
DimGray: #FF696969  
DodgerBlue: #FF1E90FF  
Firebrick: #FFB22222  
FloralWhite: #FFFFFFAF0  
ForestGreen: #FF228B22  
Fuchsia: #FFFF00FF  
Gainsboro: #FFDCDCDC  
GhostWhite: #FFF8F8FF  
Gold: #FFFFD700

Goldenrod: #FFDAA520  
Gray: #FF808080  
Green: #FF008000  
GreenYellow: #FFADFF2F  
Honeydew: #FFF0FFF0  
HotPink: #FFFF69B4  
IndianRed: #FFCD5C5C  
Indigo: #FF4B0082  
Ivory: #FFFFFFF0  
Khaki: #FFF0E68C  
Lavender: #FFE6E6FA  
LavenderBlush: #FFFFFF0F5  
LawnGreen: #FF7CFC00  
LemonChiffon: #FFFFFFACD  
LightBlue: #FFADD8E6  
LightCoral: #FFF08080  
LightCyan: #FFE0FFFF  
LightGoldenrodYellow: #FFF0FAD2  
LightGray: #FFD3D3D3  
LightGreen: #FF90EE90  
LightPink: #FFFB6C1  
LightSalmon: #FFFA07A  
LightSeaGreen: #FF20B2AA  
LightSkyBlue: #FF87CEFA  
LightSlateGray: #FF778899  
LightSteelBlue: #FFB0C4DE  
LightYellow: #FFFFFFE0  
Lime: #FF00FF00  
LimeGreen: #FF32CD32  
Linen: #FFFAF0E6  
Magenta: #FFFF00FF  
Maroon: #FF800000  
MediumAquamarine: #FF66CDAA  
MediumBlue: #FF0000CD

MediumOrchid: #FFBA55D3  
MediumPurple: #FF9370DB  
MediumSeaGreen: #FF3CB371  
MediumSlateBlue: #FF7B68EE  
MediumSpringGreen: #FF00FA9A  
MediumTurquoise: #FF48D1CC  
MediumVioletRed: #FFC71585  
MidnightBlue: #FF191970  
MintCream: #FFF5FFFA  
MistyRose: #FFF5E4E1  
Moccasin: #FFF5E4B5  
NavajoWhite: #FFF5DEAD  
Navy: #FF000080  
OldLace: #FFF5DF5E6  
Olive: #FF808000  
OliveDrab: #FF6B8E23  
Orange: #FFF5FA500  
OrangeRed: #FFF5F4500  
Orchid: #FFDA70D6  
PaleGoldenrod: #FFF5EEE8AA  
PaleGreen: #FF98FB98  
PaleTurquoise: #FFAFEEEE  
PaleVioletRed: #FFDB7093  
PapayaWhip: #FFF5FEFD5  
PeachPuff: #FFF5FDAB9  
Peru: #FFCD853F  
Pink: #FFF5FC0CB  
Plum: #FFB0E0E6  
Purple: #FF800080  
Red: #FFF5F0000  
RosyBrown: #FFBC8F8F  
RoyalBlue: #FF4169E1  
SaddleBrown: #FF8B4513  
Salmon: #FFF5FA8072

SandyBrown: #FFF4A460

SeaGreen: #FF2E8B57

SeaShell: #FFFFFF5EE

Sienna: #FFA0522D

Silver: #FFC0C0C0

SkyBlue: #FF87CEEB

SlateBlue: #FF6A5ACD

SlateGray: #FF708090

Snow: #FFFFFFAFA

SpringGreen: #FF00FF7F

SteelBlue: #FF4682B4

Tan: #FFD2B48C

Teal: #FF008080

Thistle: #FFD8BFD8

Tomato: #FFFF6347

Turquoise: #FF40E0D0

Violet: #FFEE82EE

Wheat: #FFF5DEB3

White: #FFFFFFFF

WhiteSmoke: #FFF5F5F5

Yellow: #FFFFFFF00

YellowGreen: #FF9ACD32

### **Arquivos de saída**

**Os seguintes arquivos XML são criados como um OPC-Package (=zipped):**

- Arquivo de fluxo .str

**O arquivo de fluxo contém os seguintes arquivos XML:**

- /odin/Stream.xml  
Esse XML descreve a estrutura do fluxo e contém as "sobreposições de páginas" em formato XML.
- /Odin/StreamInfo.xml  
Esse XML contém informações em fluxo contínuo.  
Em primeiro lugar, as configurações do código OMR e, em segundo lugar, informações como número de páginas, folhas, envelopes, trabalhos, bem como nos formulários e nas bandejas virtuais usadas pelo fluxo.

### **Arquivo de descrição .dsc**

Contém um XML com os dados e conjuntos de dados da pilha/envelope/documentos.

- /odin/StreamDescription.xml  
Esse arquivo armazena os conjuntos de dados do fluxo.

#### Estrutura do arquivo XML

```

<streamdescription>
  <stack>
    <fields>
      <field name="Fieldname" value="Wert" />
      ...
    </fields>
    <envelopes>
      <envelope>
        <fields>
          <field name="Fieldname" value="Wert" />
          ...
        </fields>
        <jobs>
          <job>
            <fields>
              <field name="Fieldname" value="Wert" />
              ...
            </fields>
          </job>
        </jobs>
        ...
      </envelope>
      ...
    </stack>
  </streamdescription>

```

#### Exemplo

```

<?xml version="1.0" encoding="utf-8" ?>
<streamdescription>
<stack>
<fields>
  <field name="ODS_ID" value="28" />
  <field name="ODS_STATE" value="1" />
  <field name="ODS_PRIORITY" value="200" />
  <field name="ODS_CREATE_AT" value="10/23/2011 12:34:48" />
  <field name="ODS_CREATE_BY" value="user" />
  <field name="ODS_EDIT_AT" value="11/21/2011 16:12:34" />
  <field name="ODS_EDIT_BY" value="user" />
  <field name="ODS_PRINTSTATE" value="1" />
  <field name="ODS_PRINT_START" value="11/21/2011 16:26:13" />
  <field name="ODS_PRINT_END" value="11/21/2011 16:12:34" />

```



```

<field name="ODS_LASTPRINTED" value="" />
<field name="ODS_HOLDTIME" value="10" />
<field name="ODS_DELETE_AT" value="11/22/2011 02:12:34" />
<field name="ODS_STACKTYPE" value="20" />
<field name="ODS_INITIATORATYPE" value="" />
<field name="ODS_FILTERNAME" value="Postage3" />
<field name="ODS_LASTARCHIVED" value="" />
<field name="ODS_MESSAGETEXT" value="" />
<field name="ODS_LOCKBY" value="user" />
<field name="ODS_RETRY_AT" value="" />
<field name="ODS_RETRY_COUNT" value="0" />
<field name="ODS_PRINTSTREAM" value="d:\Import\output\28\28.str" />
<field name="ODS_OMR_TYPE" value="C4_OMR.Test1" />
<field name="ODS_STACKOBJECTSQL" value="" />
<field name="ODS_PAPER_COUNT" value="2077" />
<field name="ODS_PAGE_COUNT" value="2940" />
<field name="ODS_JOB_COUNT" value="179" />
<field name="ODS_ENVELOPE_COUNT" value="10" />
</fields>
<envelopes>
  <envelope>
    <fields>
      <field name="ODE_ID" value="7" />
      <field name="ODE_STACK_ID" value="28" />
      <field name="ODE_CREATE_AT" value="10/23/2011 12:34:34" />
      <field name="ODE_CREATE_BY" value="user" />
      <field name="ODE_STATE" value="0" />
      <field name="ODE_POSTAGE" value="Postage3" />
      <field name="ODE_WEIGHT" value="416,00000" />
      <field name="ODE_NAME" value="Envelopel" />
      <field name="ODE_DEVICE" value="'Device x'" />
      <field name="ODE_DVF" value="" />
      <field name="ODE_DVF_TT" value="" />
      <field name="ODE_DVF_MMJJ" value="" />
      <field name="ODE_DVF_ABRNR" value="" />
      <field name="ODE_DVF_ELIEPLZ" value="" />
      <field name="ODE_DVF_BRUTTOENTGELD" value="" />
      <field name="ODE_DVF_CURRENCY" value="" />
      <field name="ODE_OMR_ANNEX" value="0" />
      <field name="ODE_OMR_REJECTION" value="" />
      <field name="ODE_REC_NAME" value="Locksley Sir Robin of" />
      <field name="ODE_REC_AKZ" value="D" />
      <field name="ODE_REC_ZIP" value="71711" />
      <field name="ODE_REC_CITY" value="Steinheim" />
      <field name="ODE_REC_STREET" value="Castle Rock" />
      <field name="ODE_PAPER_COUNT" value="246" />
    </fields>
  </envelope>
</envelopes>

```

```

    <field name="ODE_PAGE_COUNT" value="350" />
    <field name="ODE_SEQUENCE" value="" />
    <field name="ODE_JOB_COUNT" value="21" />
  </fields>
</jobs>
  <job>
    <fields>
      <field name="ODJ_ID" value="707" />
      <field name="ODJ_STACK_ID" value="" />
      <field name="ODJ_ENV_ID" value="7" />
      <field name="ODJ_SEQUENCE" value="1" />
      <field name="ODJ_CREATE_AT" value="10/23/2011 11:26:45" />
      <field name="ODJ_CREATE_BY" value="PAthena" />
      <field name="ODJ_STATE" value="1" />
      <field name="ODJ_ANNEX" value="" />
      <field name="ODJ_DOCNAME" value="" />
      <field name="ODJ_FILENAME"
value="D:\Import\Bin\071023_112908421\071023\11\1000\707.docx" />
      <field name="ODJ_RENDITION"
value="D:\Import\Bin\071023_112908421\071023\11\1000\707.xps" />
      <field name="ODJ_NUMBER_OF_COPIES" value="0" />
      <field name="ODJ_DUPLEX" value="0" />
      <field name="ODJ_OMR_CHANNEL" value="" />
      <field name="ODJ_ARCHIVED" value="" />
      <field name="ODE_REC_NAME" value="Locksley Sir Robin of" />
      <field name="ODJ_REC_AKZ" value="D" />
      <field name="ODJ_REC_ZIP" value="71711" />
      <field name="ODJ_REC_CITY" value="Stonehome" />
      <field name="ODJ_REC_STREET" value="Castle Rock" />
      <field name="ODJ_PROCESS_ID" value="{683de553-a858-4934-9581-
8c6b9bf0a98d}" />
      <field name="ODJ_BUNDLE_SEQ" value="0" />
      <field name="ODJ_BUNDLE_COUNT" value="0" />
      <field name="ODJ_BUNDLE_NAME" value="" />
      <field name="ODJ_BUNDLE_OID" value="" />
      <field name="ODJ_VSNR" value="" />
      <field name="ODJ_PARNR" value="" />
      <field name="ODJ_IS_COPY" value="N" />
      <field name="ODJ_DOCTITLE" value="" />
      <field name="ODJ_PAPER_COUNT" value="24" />
      <field name="ODJ_PAGE_COUNT" value="24" />
      <field name="ODJ_DIRECTPRINT" value="0" />
      <field name="ODJ_PRINTER_OID" value="odin-opr-14" />
      <field name="ODJ_PRINTER_NAME" value="PrinterOutsourcing" />
      <field name="ODJ_FORM_OID" value="odin-fo-9" />
      <field name="ODJ_FORM_NAME" value="Form1" />
    </fields>
  </job>
</jobs>

```

```

    <field name="ODJ_TO_ARCHIVE" value="" />
    <field name="ODJ_DOCTYPE" value="1" />
    <field name="ODJ_DOC_OID" value="" />
    <field name="ODJ_SORTING" value="" />
    <field name="ODJ_GROUPING" value="" />
    <field name="ODJ_LOCKBY" value="" />
    <field name="ODJ_MESSAGE TEXT" value="" />
    <field name="ODJ_TIFF_LAYOUT" value="" />
  </fields>
</job>
</envelope>- <fields> <field name="ODE_STACK_ID" value="28" />
</envelopes>- <fields> <field name="ODE_STACK_ID" value="28" />
</stack>
</streamdescription>- <fields> <field name="ODE_STACK_ID" value="28" />

```

## Propriedades de OdinStream

[▶ Mostrar tudo](#)

Propriedade	Significado		
CreatedDescriptionFile	True: Os conjuntos de dados são escritos para um arquivo de descrição separado (.dsc). False: Os conjuntos de dados são escritos para um arquivo de descrição separado (.dsc).		
DB Alias	Define o banco de dados Odin, o componente irá executar suas ações dentro.		
ExtendText	<p>Especifica a tabela de pesquisa para os textos adicionais a serem impressos nos documentos.</p> <p>Textos adicionais podem ser definidos para pilhas, envelopes e documentos, e, novamente, para layouts no modo retrato ou paisagem.</p> <p>Textos adicionais (Extended Text)</p> <p>Estes textos adicionais são definidos nas tabelas de pesquisa atribuídas à propriedade 'ExtendedText' de um componente de fluxo.</p> <p>Valores de pesquisas podem ser definidos para pilhas, páginas de capa de envelopes e documentos.</p> <p>Novamente, estas especificações podem ser ainda mais diferenciadas para layouts de páginas nos modos retrato e paisagem.</p> <table border="1" data-bbox="334 1738 1430 1837"> <tbody> <tr> <td>ExtendFlyLeafStack_Portrait_Def</td> <td>Texto adicional para layout no modo retrato para a página de capa da pilha</td> </tr> </tbody> </table>	ExtendFlyLeafStack_Portrait_Def	Texto adicional para layout no modo retrato para a página de capa da pilha
ExtendFlyLeafStack_Portrait_Def	Texto adicional para layout no modo retrato para a página de capa da pilha		

ExtendFlyLeafStack_Landscape_Def	Texto adicional para layout no modo paisagem para a página de capa da pilha
ExtendFlyLeafEnvelope_Portrait_Def	Texto adicional para layout no modo retrato para a página de capa do envelope
ExtendFlyLeafEnvelope_Landscape_Def	Texto adicional para layout no modo paisagem para a página de capa do envelope
ExtendAllPages_Portrait_Def	Texto adicional para layout no modo retrato da página de documento
ExtendAllPages_Landscape_Def	Texto adicional para layout no modo paisagem da página de documento

Até o momento, saída de texto (Type=Text), Linhas (Type=LINE), Retângulos (Type=RECTANGLE) e Imagens (Type=IMAGE) são suportados.

Várias saídas são possíveis, enumerando as entradas:

ExtendAllPages\_Portrait\_Def=TYPE=TEXT,XPOS=25,5,YPOS=280,5,fontsize=6,BOLD=N,Italic=N,fontname=ARIAL,Angle=0,TEXT="%S.ODS\_ID%"

ExtendAllPages\_Portrait\_Def2=TYPE=TEXT,XPOS=22,5,YPOS=260,5,fontsize=6,BOLD=N,Italic=Y,fontname=ARIAL,Angle=0,TEXT=2nd text

### Texto (Type=Text)

**COLOR** Cor do texto (Padrão: Black)

Define a camada de saída.

**LAYER** 0 ou primeiro plano: camada de primeiro plano (o desenho é feito por cima do documento original)

1 ou plano de fundo: camada de plano de fundo (o desenho é feito atrás do documento original)

XPOS/YPOS Especifica a posição do texto X / Y em mm (Padrão: 10 mm)

**fontsize** Tamanho da fonte (Padrão: 6)

**BOLD** Negrito S/N (Padrão: false)

**ITALIC** Itálico S/N (Padrão: false)

**fontname** Nome da fonte usada para a saída de texto. (Padrão: Arial)

**ANGLE** Ângulo do texto. (Padrão: 0)

Texto a ser escrito. Se o texto contiver espaços, ele deve ser definido conforme a seguir:

"TEXT=This is a text with spaces!"

Espaços reservados no texto são incluídos por %

**TEXT** Espaços reservados por valores de um banco de dados:

%S.Field name% - Field in the stack table "Odin\_Stack"

%E.Field name% - Field in the envelope table "Odin\_Envelope"

%J.Field name% - Field in the job table "Odin\_Job"

Além disso, todas as propriedades acessadas pelo método `GetProperty` do contexto `OdinStream` também podem ser usadas.

#### Exemplo

```
ExtendFlyLeafStack_Portrait_Def=TYPE=TEXT,XPOS=25,YPOS=280,fontsize=6,BOLD=N,Italic=N,FontName=ARIAL,Angle=0,"TEXT=Deckblatt Stapel #S.ODS_ID%"
```

```
ExtendFlyLeafEnvelope_Portrait_Def=TYPE=TEXT,XPOS=25,YPOS=280,fontSize=6,BOLD=N,Italic=N,FontName=ARIAL,Angle=0,"TEXT=Deckblatt Kuvert #E.ODE_ID%"
```

```
ExtendAllPages_Portrait_Def=TYPE=TEXT,XPOS=10,YPOS=260,fontSize=6,BOLD=N,Italic=N,FontName=Arial,Angle=270,TEXT=%S.ODS_ID%/E.ODE_ID%/J.OBJ_ID%
```

```
ExtendAllPages_Portrait_Def2=TYPE=TEXT,XPOS=28,YPOS=290,fontSize=6,BOLD=N,Italic=N,FontName=Arial,Angle=0,"TEXT=Envelope##StackEnvelope%/TotalStackEnvelopes% - Env_job##EnvelopeJob%/TotalEnvelopeJobs% - Env_Page##EnvelopePage%/TotalEnvelopePages% Page##JobPage%/TotalJobPages%"
```

```
ExtendAllPages_Landscape_Def=TYPE=TEXT,XPOS=39,YPOS=10,fontSize=6,BOLD=N,Italic=N,FontName=Arial,Angle=180,TEXT=%S.ODS_ID%/E.ODE_ID%/J.OBJ_ID%
```

```
ExtendAllPages_Landscape_Def2=TYPE=TEXT,XPOS=8,YPOS=28,fontSize=6,BOLD=N,Italic=N,FontName=Arial,Angle=270,"TEXT=Envelope##StackEnvelope%/TotalStackEnvelopes% - Env_job##EnvelopeJob%/TotalEnvelopeJobs% - Env_Page##EnvelopePage%/TotalEnvelopePages% Page##JobPage%/TotalJobPages%"
```

#### Linha (Type=LINE)

**XPOS/YPOS** Especifica a posição inicial do texto X / Y em mm (Padrão: 10 mm)

**XPOS2/YPOS2** Especifica a posição final do texto X / Y em mm (Padrão: 10 mm)

LineWidth	Especifica a espessura da linha em mm (Padrão: 1 mm)
	Especifica o estilo da linha (Padrão 0=Solid)
	0 ou sólido:
	1 ou tracejado:
LineStyle	2 ou pontilhado:
	Sólido
	Tracejado
	Pontilhado
LineColor	Cor da linha
	Define a camada de saída.
	0 ou primeiro plano:
Layer	1 ou plano de fundo:
	Camada de primeiro plano (o desenho é feito por cima do documento original)
	Camada de plano de fundo (o desenho é feito atrás do documento original)

**Exemplo:**

```
ExtendAllPages_Portrait_Def3=TYPE=LINE,XPOS=10,YPOS=10,XPOS2=60,YPOS2=60,LineWidth=0,5,LineStyle=2,LineColor=Red,Layer=1
```

```
ExtendAllPages_Portrait_Def3=TYPE=LINE,XPOS=10,YPOS=10,XPOS2=60,YPOS2=60,LineWidth=0,5,LineStyle=dotted,LineColor=Red,Layer=background
```

**Retângulo (Type=RECTANGLE)**

XPOS/YPOS	Especifica a posição X / Y em mm (Padrão: 10 mm)
Width/Height	Largura/altura do retângulo em mm (Padrão: 10 mm)
LineWidth	Especifica a espessura da linha em mm (Padrão: 1 mm)
	Especifica o estilo da linha (Padrão: 0=Solid)
	0 ou sólido: Solid
LineStyle	1 ou tracejado: Dashed
	2 ou pontilhado: Dotted
LineColor	Cor da borda (Padrão: Black)

Preenchimento	True / False Especifica se o retângulo está preenchido com cor (Padrão: false)
FillColor	A cor de preenchimento do retângulo.
Layer	Define a camada de saída. 0 ou primeiro plano: camada de primeiro plano (o desenho é feito por cima do documento original) 1 ou plano de fundo: camada de plano de fundo (o desenho é feito atrás do documento original)

**Exemplo:**

```
ExtendAllPages_Portrait_Def4=Type=RECTANGLE,XPOS=20,YPOS=20,Width=10,Height=10,LineWidth=0,5,LineStyle=1,LineColor=Black,fill=true,fillColor=yellow,Layer=0
ExtendAllPages_Portrait_Def4=Type=RECTANGLE,XPOS=20,YPOS=20,Width=10,Height=10,LineWidth=0,5,LineStyle=dashed,LineColor=Black,fill=true,fillColor=yellow,Layer=foreground
```

**Imagem (Type=IMAGE)**

XPOS/YPOS	Especifica a posição X / Y em mm (Padrão: 10 mm)
Arquivo	Nome do arquivo da imagem a ser pintado.

**Exemplo:**

```
ExtendAllPages_Portrait_Def5=Type=IMAGE,XPOS=40,YPOS=40,file=d:\temp\pa_logo.bmp,Layer=1
```

**As cores são definidas como uma string:**

#AARRGGBB AA=Alpha-Transparency, RR=Red, GG=Green, BB=Blue  
por exemplo, para Black #00000000, para White#00FFFFFF

Transparência alfa não é suportada durante a conversão.

Se 255 (FF) for especificado para o texto para Transparência alfa, o texto não é transparente, caso contrário a transparência é desenhada na cor correspondente.

Lista de todas as cores que podem ser definidas pelo nome (essas não são transparentes como o FF é definido para Transparência alfa):

AliceBlue: #FFF0F8FF  
AntiqueWhite: #FFFAEBD7  
Aqua: #FF00FFFF  
Aquamarine: #FF7FFFD4  
Azure: #FFF0FFFF  
Beige: #FFF5F5DC  
Bisque: #FFFFE4C4  
Black: #FF000000  
BlanchedAlmond: #FFFFEBCD  
Blue: #FF0000FF  
BlueViolet: #FF8A2BE2  
Brown: #FFA52A2A  
BurlyWood: #FFDEB887  
CadetBlue: #FF5F9EA0  
Chartreuse: #FF7FFF00  
Chocolate: #FFD2691E  
Coral: #FFFF7F50  
CornflowerBlue: #FF6495ED  
Cornsilk: #FFFFFF8DC  
Crimson: #FFDC143C  
Cyan: #FF00FFFF  
DarkBlue: #FF00008B  
DarkCyan: #FF008B8B  
DarkGoldenrod: #FFB8860B  
DarkGray: #FFA9A9A9  
DarkGreen: #FF006400  
DarkKhaki: #FFBDB76B  
DarkMagenta: #FF8B008B  
DarkOliveGreen: #FF556B2F  
DarkOrange: #FFFF8C00  
DarkOrchid: #FF9932CC  
DarkRed: #FF8B0000  
DarkSalmon: #FFE9967A  
DarkSeaGreen: #FF8FBC8B



DarkSlateBlue: #FF483D8B  
DarkSlateGray: #FF2F4F4F  
DarkTurquoise: #FF00CED1  
DarkViolet: #FF9400D3  
DeepPink: #FFFF1493  
DeepSkyBlue: #FF00BFFF  
DimGray: #FF696969  
DodgerBlue: #FF1E90FF  
Firebrick: #FFB22222  
FloralWhite: #FFFFFFAF0  
ForestGreen: #FF228B22  
Fuchsia: #FFFF00FF  
Gainsboro: #FFDCDCDC  
GhostWhite: #FFF8F8FF  
Gold: #FFFD700  
Goldenrod: #FFDAA520  
Gray: #FF808080  
Green: #FF008000  
GreenYellow: #FFADFF2F  
Honeydew: #FFF0FFF0  
HotPink: #FFF69B4  
IndianRed: #FFCD5C5C  
Indigo: #FF4B0082  
Ivory: #FFFFFFF0  
Khaki: #FFF0E68C  
Lavender: #FFE6E6FA  
LavenderBlush: #FFF0F5  
LawnGreen: #FF7CFC00  
LemonChiffon: #FFFFACD  
LightBlue: #FFADD8E6  
LightCoral: #FFF08080  
LightCyan: #FFE0FFFF  
LightGoldenrodYellow: #FFFAFAD2  
LightGray: #FFD3D3D3

LightGreen: #FF90EE90  
LightPink: #FFFB6C1  
LightSalmon: #FFFA07A  
LightSeaGreen: #FF20B2AA  
LightSkyBlue: #FF87CEFA  
LightSlateGray: #FF778899  
LightSteelBlue: #FFB0C4DE  
LightYellow: #FFFFFFE0  
Lime: #FF00FF00  
LimeGreen: #FF32CD32  
Linen: #FFFAF0E6  
Magenta: #FFFF00FF  
Maroon: #FF800000  
MediumAquamarine: #FF66CDAA  
MediumBlue: #FF0000CD  
MediumOrchid: #FFBA55D3  
MediumPurple: #FF9370DB  
MediumSeaGreen: #FF3CB371  
MediumSlateBlue: #FF7B68EE  
MediumSpringGreen: #FF00FA9A  
MediumTurquoise: #FF48D1CC  
MediumVioletRed: #FFC71585  
MidnightBlue: #FF191970  
MintCream: #FFF5FFFA  
MistyRose: #FFF4E1  
Moccasin: #FFF4B5  
NavajoWhite: #FFF4DEAD  
Navy: #FF000080  
OldLace: #FFF4F5E6  
Olive: #FF808000  
OliveDrab: #FF6B8E23  
Orange: #FFFA500  
OrangeRed: #FFF4500  
Orchid: #FFDA70D6

PaleGoldenrod: #FFEEE8AA  
PaleGreen: #FF98FB98  
PaleTurquoise: #FFAFEEEE  
PaleVioletRed: #FFDB7093  
PapayaWhip: #FFFEEFD5  
PeachPuff: #FFFFDAB9  
Peru: #FFCD853F  
Pink: #FFFC0CB  
Plum: #FFB0E0E6  
Purple: #FF800080  
Red: #FFF0000  
RosyBrown: #FFBC8F8F  
RoyalBlue: #FF4169E1  
SaddleBrown: #FF8B4513  
Salmon: #FFFA8072  
SandyBrown: #FFF4A460  
SeaGreen: #FF2E8B57  
SeaShell: #FFFFFF5EE  
Sienna: #FFA0522D  
Silver: #FFC0C0C0  
SkyBlue: #FF87CEEB  
SlateBlue: #FF6A5ACD  
SlateGray: #FF708090  
Snow: #FFFFFFAFA  
SpringGreen: #FF00FF7F  
SteelBlue: #FF4682B4  
Tan: #FFD2B48C  
Teal: #FF008080  
Thistle: #FFD8BFD8  
Tomato: #FFF6347  
Turquoise: #FF40E0D0  
Violet: #FFEE82EE  
Wheat: #FFF5DEB3  
White: #FFFFFFFF

	<p>WhiteSmoke: #FFF5F5F5</p> <p>Yellow: #FFFFFF00</p> <p>YellowGreen: #FF9ACD32</p>																					
NextStackType	<p>Especifica o NextStackType que é definido após uma execução bem sucedida e após a chamado do método 'End'.</p> <p>A lista suspensa exibe todos os tipos de pilha possíveis.</p> <p>Qualquer tipo de pilha necessários adicionalmente deve ser definido no arquivo ModusSuite.Common.PropertyEditors.dll.config.</p> <p><b>Tipos de pilha padrão</b></p> <table border="1"> <tr> <td>0</td> <td>Print</td> <td>A pilha está pronta para imprimir em uma impressora online ou impressora de e-mail (impressora online)</td> </tr> <tr> <td>10</td> <td>PortoOptimization</td> <td>A pilha está pronta para IT Franking (não suportado no momento!)</td> </tr> <tr> <td>20</td> <td>Streaming</td> <td>A pilha está pronta para o fluxo.</td> </tr> <tr> <td>30</td> <td>Convert</td> <td>A pilha está pronta para se convertida ao formato de saída apropriado.</td> </tr> <tr> <td>40</td> <td>StreamPrint</td> <td>Pilha/fluxo) está pronto para impressão em massa em uma impressora terceirizada.</td> </tr> <tr> <td>50</td> <td>Archive</td> <td>A pilha está pronta para o arquivamento.</td> </tr> <tr> <td>99</td> <td>Finish</td> <td>O processamento da pilha está concluído, nenhuma ação adicional.</td> </tr> </table>	0	Print	A pilha está pronta para imprimir em uma impressora online ou impressora de e-mail (impressora online)	10	PortoOptimization	A pilha está pronta para IT Franking (não suportado no momento!)	20	Streaming	A pilha está pronta para o fluxo.	30	Convert	A pilha está pronta para se convertida ao formato de saída apropriado.	40	StreamPrint	Pilha/fluxo) está pronto para impressão em massa em uma impressora terceirizada.	50	Archive	A pilha está pronta para o arquivamento.	99	Finish	O processamento da pilha está concluído, nenhuma ação adicional.
0	Print	A pilha está pronta para imprimir em uma impressora online ou impressora de e-mail (impressora online)																				
10	PortoOptimization	A pilha está pronta para IT Franking (não suportado no momento!)																				
20	Streaming	A pilha está pronta para o fluxo.																				
30	Convert	A pilha está pronta para se convertida ao formato de saída apropriado.																				
40	StreamPrint	Pilha/fluxo) está pronto para impressão em massa em uma impressora terceirizada.																				
50	Archive	A pilha está pronta para o arquivamento.																				
99	Finish	O processamento da pilha está concluído, nenhuma ação adicional.																				
OMR	<p>Especifica a tabela de pesquisa para as definições OMR.</p> <p>OMR-Definição</p> <p>Os códigos OMR são definidos na pilha criada, por exemplo, em um filtro por meio do SetProperty("OMR_CODE").</p> <p>Todas as posições e configurações de tamanho de um código OMR são feitas em mm.</p> <p><b>Os seguintes códigos OMR são suportados atualmente:</b></p> <p>C4_OMR</p> <p>C4EF_OMR</p>																					

	<p>K34_OMR  K34K_OMR  BELL_OMR  AS1A_OMR  STIELOW_OMR  SI68_OMR  KERN_POST_OMR  Kern Post 2</p> <p>As seguintes configurações podem ser definidas para cada OMR:</p> <p>OMR_POS_X /  OMR_POS_Y      Posições iniciais horizontal e vertical para o código OMR no papel.</p> <p>OMR_LINELENGTH  /  OMR_LINEHEIGHT      Comprimento e largura das linhas OMR.</p> <p>OMR_GAP      Espaço entre cada linha</p> <p>OMR_EDGE      Especifica se o código OMR é impresso na borda longa (=LONG) ou na borda curta (=SHORT) de uma página.</p> <p>Padrão: LONG</p> <p>Se forem necessárias configurações variáveis para um único tipo OMR, elas podem ser definidas inserindo outro nome de configuração separado por um ponto:</p> <p>OMR_TYP.CONFIG</p> <p><b>Exemplo</b></p> <p>C4_OMR=OMR_POS_X=10,OMR_POS_Y=10,OMR_LineLength=5,OMR_LineHeight=0,5,OMR_Gap=5,OMR_Edge=Long  C4_OMR.ShortEdge=OMR_POS_X=10,OMR_POS_Y=10,OMR_LineLength=5,OMR_LineHeight=0,5,OMR_Gap=5,OMR_Edge=Short</p>
StackSelect	<p>Seleção de SQL para identificar a pilha a ser processada.</p> <p>Ou insira a seleção de instrução diretamente ou consulte uma tabela de pesquisa (por exemplo, \$OdinSQL.PrintStackSelect\$).</p>
StreamOutputPath	<p>Diretório de saída para o arquivo de fluxo-/descrição.</p> <p>O diretório em que uma subpasta com o nome da ID da pilha é criado e no qual o arquivo de fluxo é salvo.</p>

## Eventos de OdinStream

[▶ Mostrar tudo](#)

Evento	Significado
OnDocumentStream	<p>Este evento é exigido para cada documento e oferece a possibilidade de modificar páginas de documentos.</p> <p><b>Contexto disponível:</b></p> <p>Contexto <a href="#">OdinStreamDocument</a></p> <p><b>Evento de script de exemplo 'OnDocumentStream' com Bookmarks</b></p> <pre> ; ; Test access to database tables ods_id = CurrentStack.GetValue("ODS_ID") ode_id = CurrentEnvelope.GetValue("ODE_ID") odj_id = CurrentJob.GetValue("ODJ_ID") ; ; Set Bookmark for the 1. job page.. jobBookmark = FormatStr("Job %s", odj_id) envelopeBookmark = FormatStr("Envelope %s", ode_id) ; x.EditPage(0) streamdoc.AddBookmark(jobBookmark, envelopeBookmark, jobBookmark, 0, 0) streamdoc.Post() ; pagenumber = 0 while (pagenumber &lt; streamdoc.PageCount)     streamdoc.EditPage(pagenumber)     ;     ; sets for each page a Bookmark as child of Job- Bookmark     pageBookmark = FormatStr("Page %d", pagenumber + 1)     pageBookmarkId = FormatStr("%s_%s", jobBookmark, pageBookmark)     streamdoc.AddBookmark(pageBookmarkId, jobBookmark, pageBookmark, 0, 0)     ;     streamdoc.Post()     pagenumber = pagenumber + 1 ; end-while ; </pre>

## OnEnvelopeCoverPage

Este evento é exigido para cada envelope para criar uma capa para um envelope e adicioná-lo ao fluxo.

**Contexto disponível:**

Contexto [OdinStream](#)

Evento de script de exemplo 'OnEnvelopeCoverPage' com Bookmarks

```
; load empty cover page. The trays firstbin/nextbin
must exists
document =
stream.LoadDocument("c:\PAG\Docs\Empty_Page.xps",
OdinDuplexSetting.Simplex, "firstbin","nextbin")
;
; Determine envelope information
envPages = stream.GetProperty("TotalEnvelopePages")
envJobs = stream.GetProperty("TotalEnvelopeJobs")
;
ods_id = CurrentStack.GetValue("ODS_ID")
ode_id = CurrentEnvelope.GetValue("ODE_ID")
;
; as a test, get data from the VirtualDatabase
zip = GetField("ENVELOPE", "ODE_REC_ZIP")
city = GetField("ENVELOPE", "ODE_REC_CITY")
country = GetField("ENVELOPE", "ODE_REC_AKZ")
;
AdrName = GetField("ENVELOPE", "ODE_REC_NAME")
AdrStreet = GetField("ENVELOPE", "ODE_REC_STREET")
AdrCity = formatstr("%s - %s %s", country, zip, city)
EnvId = FormatStr("%s", ode_id)
EnvJobPages = formatstr("%d/%d", envJobs, envPages)
;
stackBookmark = FormatStr("Stapel %s", ods_id)
envelopeBookmark = FormatStr("Envelope %s", ode_id)
coverPageBookmark = formatstr("Envelope_Cover_Page_%s",
EnvId)
;
; Print information on the cover page
document.EditPage(0)
stackBookmark = FormatStr("Stapel %s", ods_id)
envelopeBookmark = FormatStr("Envelope %s", ode_id)
document.AddBookmark(envelopeBookmark, stackBookmark,
envelopeBookmark, 0, 0)
document.AddBookmark(coverPageBookmark,
envelopeBookmark, "Kuvertdeckblatt", 0, 0)
;
document.SetFont("Tahoma", 16)
document.TextOut(20,5, 35, "EnvelopeCoverPage", false,
```

	<pre> false) document.Line(20,5, 37, 70, 37, 0,5, OdinLineStyle.Solid, "black") ; document.SetFont("Tahoma", 12) document.TextOut(20,5, 42, "Envelope-ID:", false, false) document.TextOut(20,5, 46, "Number Jobs/Pages:", false, false) ; document.TextOut(65, 42, EnvId, true, false, "Gray", 0) document.TextOut(65, 46, EnvJobPages, true, false, "Gray", 0) ; document.TextOut(20,5, 55, "Recipient-Address:", true, false) document.TextOut(20,5, 62, AdrName, false, false, "Gray", 0) document.TextOut(20,5, 66, AdrStreet, false, false, "Gray", 0) document.TextOut(20,5, 70, AdrCity, false, false, "Gray", 0) ; document.Image(10, 5, "d:\temp\2_compart-logo- small.gif") document.Post() ; ; Add cover page to stream stream.AppendDocument(document, true, false) ; ; Navigate to the next envelope for VirtualDB Test/GetField Next("Envelope") ; </pre>
OnStackCoverPage	<p>Este evento é exigido no início da pilha para criar uma capa para a pilha e adicioná-la ao fluxo.</p> <p><b>Contexto disponível:</b></p> <p>Contexto <a href="#">OdinStream</a></p> <p>Evento de script de exemplo 'OnStackCoverPage' com Bookmarks</p> <pre> ; load empty cover page. The trays firstbin/nextbin must exists document = stream.LoadDocument("c:\PAG\Docs\Empty_Page.xps", OdinDuplexSetting.Simplex, "firstbin","nextbin") </pre>



```

;
; Determine stack information
TotalPages = stream.GetProperty("TotalStackPages")
TotalPapers = stream.GetProperty("TotalStackPapers")
TotalEnvelopes =
stream.GetProperty("TotalStackEnvelopes")
TotalJobs = stream.GetProperty("TotalStackJobs")
;
ods_id = CurrentStack.GetValue("ODS_ID")
;
StackId = FormatStr("%s", ods_id)
EnvJobCount = FormatStr("%d/%d", TotalEnvelopes,
TotalJobs)
PageCount = FormatStr("%d/%d", TotalPages, TotalPapers)
;
; Print information on the cover page
document.EditPage(0)
;
; set Bookmark for stack
stackBookmark = FormatStr("Stapel %s", ods_id)
document.AddBookmark(stackBookmark, "", stackBookmark,
0, 0)
document.AddBookmark("StackCoverPage", stackBookmark,
"StackCoverPage", 0, 0)
;
document.SetFont("Tahoma", 16)
document.TextOut(20,5, 35, "StackCoverPage", false,
false)
document.Line(20,5, 37, 60, 37, 0,5,
OdinLineStyle.Solid, "black")
;
document.SetFont("Tahoma", 12)
document.TextOut(20,5, 42, "stack-ID:", false, false)
document.TextOut(20,5, 46, "Number Envelopes/Jobs:",
false, false)
document.TextOut(20,5, 50, "Number Pages/Sheets:",
false, false)
;
document.TextOut(65, 42, StackId, true, false, "Gray",
0)
document.TextOut(65, 46, EnvJobCount, true, false,
"Gray", 0)
document.TextOut(65, 50, PageCount, true, false,
"Gray", 0)
document.Image(10, 5, "d:\temp\2_compart-logo-
small.gif")
;

```

	<pre>document.Post() ; ; Add cover page to stream stream.AppendDocument(document, true, false)</pre>
OnStackEnd	<p>Este evento é exigido após o processamento / fluxo de pilha de acordo com o caso em questão.</p> <p><b>Contexto disponível:</b> Contexto <a href="#">OdinStream</a></p>
OnStackStart	<p>Este evento é exigido após o processamento / fluxo de pilha de acordo com o caso em questão.</p> <p><b>Contexto disponível:</b> Contexto <a href="#">OdinStream</a></p>

## ***Métodos de OdinStream***

### **AddBookmark**

Adiciona um bookmark a um documento. Se o formato de saída for PDF, um bookmark é criado para cada bookmark definido.

As IDs deve ser exclusivas para cada bookmark.

### **Sintaxe**

AddBookmark(string id, string parentId, string title)

AddBookmark(string id, string parentId, string title, double x, double y)

### **Parâmetro**

<b>Parâmetro</b>	<b>Significado</b>
id	ID do Bookmark
parentId	Uma string vazia aqui significa que o bookmark será inserido no nível mais elevado. Caso contrário, o bookmark é inserido como um filho do bookmark especificado.
title	Texto para o bookmark
x	Posição X na página

y	Posição Y na página
---	---------------------

**Valor de retorno**

-

**Exemplo**

```
document.AddBookmark("doc1_page1", "doc1", "Page 1")
document.AddBookmark("doc1_page2", "doc1", "Page 2", 0, 0)
```

**End**

Finaliza o processamento do componente, conjuntos a pilha para o próximo tipo de pilha e, quando relevante, executa o script atribuído para o evento 'OnStackEnd'.

**Os seguintes valores são definidos:**

ODS_Print_End:	Data e hora atuais
ODS_Delete_At:	Data e hora atuais + tempo de espera da pilha

**Se o processamento for finalizado corretamente:**

ODS_PrintState:	2 (OdinPrintState.Ok)
ODS_State:	'NextStackState' do componente
ODS_StackType:	'NextStackState' do componente

**No erro:**

ODS_PrintState:	3 (OdinPrintState.Error)
ODS_State:	3 (OdinStackState.Locked)
ODS_MessageText:	Mensagem de erro

**Sintaxe**

End()

End(string errorMessage)

**Parâmetro**

Parâmetro	Significado
errorMessage	Mensagem de erro

**Valor de retorno**

-

**GetCurrentStackRow**[▶ Mostrar tudo](#)

Após chamar Start() e antes de chamar End(), o método retorna um objeto cuja interface fornece acesso de leitura ao conjunto de dados atual da pilha.

**Sintaxe**

GetCurrentStackRow()

**Parâmetro**

-

**Valor de retorno**

Interface de objeto que fornece acesso de leitura ao conjunto de dados atual da pilha.

**Os seguintes métodos estão disponíveis:**

GetValue

Retorna o valor da coluna dada.

**Sintaxe**

GetValue(string columnName)

Parâmetro	Significado
columnName	Nome da coluna

**Valor de retorno**

Valor da coluna

Se o tipo de dados da coluna do banco de dados for DATE, Datetime ou qualquer tipo de dados válido, a data lida geralmente será retornada no formato MM/DD/AAAA hh:mm:ss.

**Exemplo**

```
GetObject("DateVar", "MLDate")
```

...

```
StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT")
;Converter a data no formato desejado com a ajuda do objeto MLDate
IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy hh:mm:ss", StringDateVar )
v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy hh:mm:ss ", IntDateVar)
```

GetFieldNameEnumerator

Retorna um enumerador.

**Sintaxe**

GetEnvelopeEnumerator()

#### Parâmetro

-

#### Valor de retorno

Enumerador com estes métodos:

Método	Função
Current	Retorna o objeto atual.
MoveNext	Move a posição até o próximo objeto a ser processado.
Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.

#### GetEnvelopeEnumerator

Retorna um enumerador de envelope. A função GetValue() recupera dados de envelope.

#### Sintaxe

GetEnvelopeEnumerator()

#### Parâmetro

-

#### Valor de retorno

Enumerador com estes métodos:

Método	Significado
Current	Retorna o envelope atual.
MoveNext()	<p>Move a posição para o próximo envelope a ser processado.</p> <p><b>Valor de retorno:</b></p> <p>True: êxito</p> <p>False: o fim da coleta foi passado.</p> <p><b>Exemplo</b></p> <pre>... envEnumerator = OdinPrint.GetEnvelopeEnumerator() nextElement = envEnumerator.MoveNext() while (nextElement)     envelope = envEnumerator.Current;</pre>

	<pre>... nextElement = envEnumerator.MoveNext();</pre>
Reset	Redefine o enumerador de volta para o início do primeiro elemento.

### GetJobEnumerator

Retorna um enumerador de trabalho.

A função GetValue() recupera dados de trabalho.

### Sintaxe

GetJobEnumerator()

GetJobEnumerator(envelope)

### Parâmetro

Parâmetro	Significado
envelope	O envelope cujos trabalhos serão acessados.

### Valor de retorno

Enumerador com estes métodos:

Método	Significado
Current	Retorna o trabalho atual.
MoveNext()	<p>Move a posição para o próximo trabalho a ser processado.</p> <p><b>Valor de retorno:</b></p> <p>True: êxito</p> <p>False: o fim da coleta foi passado.</p> <p><b>Exemplo</b></p> <pre>... jobEnumerator = OdinPrint.GetJobEnumerator() nextElement = jobEnumerator.MoveNext() while (nextElement)     job = jobEnumerator.Current; ... nextElement = jobEnumerator.MoveNext()</pre>
Reset	Redefine o enumerador de volta para o início do primeiro elemento.

**GetNextStack**

Detecta a próxima pilha a ser processada de acordo com a pilha selecionada.

**Sintaxe**

GetNextStack()

**Parâmetro**

-

**Valor de retorno**

ID da pilha encontrada ou uma string vazia, se nenhuma for encontrada.

**SetStackError**

Define o status de uma pilha para erro.

**Sintaxe**

SetStackError(errorMessage)

**Parâmetro**

Parâmetro	Significado
errorMessage	Mensagem de erro

**Valor de retorno**

-

**Start**

Carrega e inicializa a pilha dada e executa um script atribuído para o evento **OnStackCoverPage**.

A pilha deve ser do tipo correspondente ao requisitado pelo componente. O status deve ser definido como **Aguardar**.

Após ser carregada com êxito, a pilha é bloqueada e os status são definidos de acordo:

ODS_LockBy:	Nome do usuário atual.
ODS_State:	1 (OdinStackState.InWork)
ODS_PrintState:	1 (OdinPrintState.InWork)
ODS_Print_Start:	Data e hora atuais
ODS_MessageText:	String vazia

**Sintaxe**

Start(string stackId)

**Parâmetro**

Parâmetro	Significado
stackId	ID de pilha

**Valor de retorno**

-

**StreamDocument**

Coloca o documento de determinado trabalho em fluxo.

**Sintaxe**

StreamDocument(IOdinReadOnlyDataRow job)

**Parâmetro**

Parâmetro	Significado
job	Objeto de trabalho

**Valor de retorno**

-

**Script de exemplo OdinStream**

```

;
; get component for streaming
OdinStream = ModusProcess.GetComponent("OdinStream")
;
; determine the next stack for streaming
stackId = OdinStream.GetNextStack()
;
; was a stack found ?
if (stackID <> "")
;
    ; yes, now initialize stack and stream.
    ; call the StackCoverPage event.
    OdinStream.Start(stackId)
        ; get an envelope enumerator.
        envEnumerator = OdinStream.GetEnvelopeEnumerator()
        ;

```



```

; run through all envelopes and stream jobs
nextEnv = envEnumerator.MoveNext()
while (nextEnv)
    ;
    ; get current envelope from enumerator
    envelope = envEnumerator.Current
    ; as a test get envelope ID
    envelopId = envelope.GetValue("ODE_ID")
    ;
    ; all jobs in the envelope could be streamed with the method
StreamDocuments
    ;OdinStream.StreamDocuments(envelope)
    ;
    ; get an job enumerator for the envelope.
    jobEnumerator = OdinStream.GetJobEnumerator(envelope)
    ; run through all envelope jobs and stream
    nextJob = jobEnumerator.MoveNext()
    while (nextJob)
        ;
        ; get current job from enumerator
        job = jobEnumerator.Current
        ; as a test get job ID
        jobId = job.GetValue("ODJ_ID")
        ;
        ; and now archive, call the OnArchive script
        OdinStream.StreamDocument(job)
        ;
        ; process next Job
        nextJob = jobEnumerator.MoveNext()
        ;
    end-while
    ;
    ; process next envelope
    nextEnv = envEnumerator.MoveNext()
end-while
;
; Streaming finished, call EndStack
OdinStream.End()
end-if

```

## OdinStreamPrint

### **Informações gerais**

Este componente fica disponível somente quando o Odin tiver sido licenciado corretamente.

Este componente permite a saída de um fluxo de impressão Odin para um driver de impressora.

Em vez de converter um fluxo para o formato de saída obrigatório por meio do componente **OdinConvert**, o mesmo pode ser realizado por meio do componente **OdinStreamPrint** e direcionado para qualquer driver de impressora.

A pré-condição para isso é que as bandejas usadas no fluxo também sejam definidas para a impressora terceirizada correspondente.

Se uma impressora não é atribuída ao componente, ela tentará carregar a impressora definida no arquivo de fluxo.

Os nomes de bandeja na impressora terceirizada são usadas para identificar as IDs da bandeja física para impressão.

As configurações para duplex etc são tiradas daquelas definidas no arquivo de fluxo.

Os eventos da impressora terceirizada são executados junto aos eventos **OnBeforeStack/OnAfterStack**.

### Consulte também

Contexto [OdinStreamPrint](#)

### ***Propriedades de OdinStreamPrint***

Propriedade	Significado																			
DB Alias	Define o banco de dados Odin no qual o componente irá executar suas ações.																			
NextStackType	<p>Especifica o próximo tipo de pilha definido após uma execução com êxito e após a chamada do método End.</p> <p>A lista seguinte exibe todos os possíveis tipos de pilhas.</p> <p>Qualquer tipo de pilha necessário adicionalmente deve ser definido no arquivo <b>ModusSuite.Common.PropertyEditors.dll.config</b>.</p> <p><b>Tipos de pilha padrão</b></p> <table border="1"> <thead> <tr> <th>Valor</th> <th>Nome</th> <th>Descrição</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Print</td> <td>A pilha está pronta para a impressão em uma impressora online ou impressora de e-mail (impressão online)</td> </tr> <tr> <td>10</td> <td>PortoOptimization</td> <td>A pilha está pronta para IT Franking (não suportado no momento!)</td> </tr> <tr> <td>20</td> <td>Streaming</td> <td>A pilha está pronta para o fluxo.</td> </tr> <tr> <td>30</td> <td>Convert</td> <td>A pilha está pronta para ser convertida ao formato de saída apropriado.</td> </tr> <tr> <td>40</td> <td>StreamPrint</td> <td>A pilha (fluxo) está pronta para impressão em lote em uma impressora terceirizada.</td> </tr> </tbody> </table>		Valor	Nome	Descrição	0	Print	A pilha está pronta para a impressão em uma impressora online ou impressora de e-mail (impressão online)	10	PortoOptimization	A pilha está pronta para IT Franking (não suportado no momento!)	20	Streaming	A pilha está pronta para o fluxo.	30	Convert	A pilha está pronta para ser convertida ao formato de saída apropriado.	40	StreamPrint	A pilha (fluxo) está pronta para impressão em lote em uma impressora terceirizada.
Valor	Nome	Descrição																		
0	Print	A pilha está pronta para a impressão em uma impressora online ou impressora de e-mail (impressão online)																		
10	PortoOptimization	A pilha está pronta para IT Franking (não suportado no momento!)																		
20	Streaming	A pilha está pronta para o fluxo.																		
30	Convert	A pilha está pronta para ser convertida ao formato de saída apropriado.																		
40	StreamPrint	A pilha (fluxo) está pronta para impressão em lote em uma impressora terceirizada.																		

	50	Archive	A pilha está pronta para o arquivamento.
	99	Finish	O processamento da pilha está concluído, nenhuma ação adicional
Printer	Impressora terceirizada para impressão do fluxo de impressão.		
StackSelect	Define a instrução selecionada para detectar a próxima pilha a ser processada. Ou insira a seleção de instrução diretamente ou consulte uma tabela de pesquisa (por exemplo, \$OdinSQL.PrintStackSelect\$).		

### ***Eventos de OdinStreamPrint***

<b>Evento</b>	<b>Significado</b>
OnAfterStack	<p>O script especificado é executado após o processamento da pilha, enquanto chama o método <b>End</b>.</p> <p>O evento corresponde ao evento <b>AfterStack</b> de uma impressora no modus 4.</p> <p><b>Contexto disponível:</b> <b>Contexto OdinCustomProcess</b></p>
OnBeforeStack	<p>O script especificado é ao chamar o método <b>Start()</b>, antes de executar o processamento de pilha.</p> <p>O evento corresponde ao evento <b>BeforeStack</b> de uma impressora no modus 4.</p> <p><b>Contexto disponível:</b> <b>Contexto OdinCustomProcess</b></p>

### ***Métodos de OdinStreamPrint***

#### **DefinePrinterByName**

Define uma impressora por seu nome.

#### **Sintaxe**

DefinePrinterByName(string printerName)

#### **Parâmetro**

Parâmetro	Significado
printerName	Nome da impressora

**Valor de retorno**

-

**DefinePrinterByOld**

Define a impressora por uso de suas ID de objeto.

**Sintaxe**

DefinePrinterByOld(printerOld)

**Parâmetro**

Parâmetro	Significado
printerOld	ID do objeto da impressora.

**Valor de retorno**

-

**Finalizar**

Finaliza o processamento do componente, define a pilha para o tipo da próxima pilha e, se for relevante, executa um script atribuído ao evento 'OnAfterStack'.

**Os seguintes valores são definidos:**

ODS_Print_End:	Data e hora atuais
ODS_Delete_At:	Data e hora atuais + tempo de espera da pilha

**Se o processamento for finalizado corretamente:**

ODS_PrintState:	2 (OdinPrintState.Ok)
ODS_State:	'NextStackState' do componente
ODS_StackType:	'NextStackState' do componente

**No erro:**

ODS_PrintState:	3 (OdinPrintState.Error)
ODS_State:	3 (OdinStackState.Locked)

ODS_MessageText:	Mensagem de erro
------------------	------------------

**Sintaxe**

End()

End(string errorMessage)

**Parâmetro**

Parâmetro	Significado
errorMessage	Mensagem de erro

**Valor de retorno**

-

**GetCurrentStackRow**[▶ Mostrar tudo](#)

Após chamar Start() e antes de chamar End(), o método retorna um objeto cuja interface fornece acesso de leitura ao conjunto de dados atual da pilha.

**Sintaxe**

GetCurrentStackRow()

**Parâmetro**

-

**Valor de retorno**

Interface de objeto que fornece acesso de leitura ao conjunto de dados atual da pilha.

**Os seguintes métodos estão disponíveis:**

GetValue

Retorna o valor da coluna dada.

**Sintaxe**

GetValue(string columnName)

Parâmetro	Significado
columnName	Nome da coluna

**Valor de retorno**

Valor da coluna

Se o tipo de dados da coluna do banco de dados for DATE, Datetime ou qualquer tipo de dados válido, a data lida geralmente será retornada no formato MM/DD/AAAA hh:mm:ss.

### Exemplo

```
GetObject("DateVar", "MLDate")
...
StringDateVar = CurrentEnvelope.GetValue("ODE_CREATE_AT")
;Converter a data no formato desejado com a ajuda do objeto MLDate
IntDateVar = DateVar.DateStringToInt64("MM/dd/yyyy hh:mm:ss", StringDateVar )
v_LETTERDATE= DateVar.Int64ToDateString("MM.dd.yyyy hh:mm:ss ", IntDateVar)
GetFieldNameEnumerator
```

Retorna um enumerador.

### Sintaxe

GetEnvelopeEnumerator()

### Parâmetro

-

### Valor de retorno

Enumerador com estes métodos:

Método	Função
Current	Retorna o objeto atual.
MoveNext	Movê a posição até o próximo objeto a ser processado.
Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.

### GetDocumentEnumerator

Retorna um enumerador de documento.

### Sintaxe

GetDocumentEnumerator()

### Parâmetro

-

### Valor de retorno

Enumerador com estes métodos:

Parâmetro	Significado
Current	Retorna o documento atual.

MoveNext	Move a posição até o próximo documento a ser processado.
Redefinir	Redefine o enumerador de volta ao início do primeiro elemento.

**GetNextStack**

Detecta a próxima pilha a ser processada de acordo com a pilha selecionada.

**Sintaxe**

GetNextStack()

**Parâmetro**

-

**Valor de retorno**

ID da pilha encontrada ou uma string vazia, se nenhuma for encontrada.

**PrintDocument**

Imprime o documento especificado.

**Sintaxe**

PrintDocument(document)

**Parâmetro**

Parâmetro	Significado
document	O documento a ser impresso.

**Valor de retorno**

-

**SetStackError**

Define o status de uma pilha para erro.

**Sintaxe**

SetStackError(errorMessage)

**Parâmetro**

Parâmetro	Significado
-----------	-------------

errorMessage	Mensagem de erro
--------------	------------------

### Valor de retorno

-

### Start

Carrega e inicializa a pilha determinada e executa o script atribuído ao evento **OnBeforeStack**, se encontrado.

A pilha deve ser do tipo correspondente ao requisitado pelo componente. O status deve ser definido como **Aguardar**.

Após ser carregada com êxito, a pilha é bloqueada e os status são definidos de acordo:

ODS_LockBy:	Nome do usuário atual
ODS_State:	1 (OdinStackState.InWork)
ODS_PrintState:	1 (OdinPrintState.InWork)
ODS_Print_Start:	Data e hora atuais
ODS_MessageText:	String vazia

### Sintaxe

Start(string stackId)

### Parâmetro

Parâmetro	Significado
stackId	ID de pilha

### Valor de retorno

-

### Script de exemplo OdinStreamPrint

```
; initialize StreamPrint components
OdinStreamPrint = ModusProcess.GetComponent("OdinStreamPrint")
; determine next stack
stackID = OdinStreamPrint.GetNextStack()
;
; was a stack found ?
if (stackID <> "")
;
```



```

; Yes, now initialize stack print
OdinStreamPrint.Start(stackID)
;
; get a DocumentEnumerator and print each document in the stream
DocumentEnumerator = OdinStreamPrint.GetDocumentEnumerator()
;
nextDocument = DocumentEnumerator.MoveNext()
;
while(nextDocument = true)
; print next document in the stream
document = DocumentEnumerator.Current
OdinStreamPrint.PrintDocument(document)
;
; navigate the stream to the next document
nextDocument = DocumentEnumerator.MoveNext()
end-while
; end stream print
OdinStreamPrint.End()
end-if

```

## Componentes de XData

### DataSelectionManager

#### **Informações gerais**

Este componente fica disponível somente quando o Xdata tiver sido licenciado corretamente.

Use esse componente em um processo para executar a recuperação de dados definida por uma seleção Xdata.

#### **Propriedades do DataSelectionManager**

Propriedade	Significado
DebugMode	Executa uma seleção no depurador: True / False
LastError	Use isso para recuperar informações de um erro ocorrido durante a execução da recuperação de dados.
NavigatorXml	Se o método GetDataExtended foi chamado com o parâmetro 'writeNavigatorXml = True', o Navigationsxml pode ser lido.
SelectionReference	Define a seleção que o componente executará. Parâmetros de seleção são passados com a ajuda de variáveis de pool do sistema (consulte Script de exemplo).
SystemOld	Se um sistema OID for definido, a seleção é retirada do sistema

	correspondente. Caso contrário, ela será retirada do mesmo sistema do próprio processo.
XmlData	Lê o XML da última recuperação de dados como uma string.

### ***Métodos de DataSelectionManager***

#### **GetData**

Executa recuperação de dados.

#### **Sintaxe**

GetData(bool raiseInSelvariable)

#### **Parâmetro**

Parâmetro	Significado
raiseInSelvariable	Define se um erro será lançado se um script de leitura com SelVariable tentar ler um objeto de dados que não foi encontrado.

#### **Valor de retorno**

<> 0 se um erro ocorrer.

#### **GetDataExtended**

Executa recuperação de dados.

#### **Sintaxe**

GetDataExtended(bool raiseInSelvariable, bool writeNavigatorXml)

#### **Parâmetro**

Parâmetro	Significado
raiseInSelvariable	Define se um erro será lançado se um script de leitura com SelVariable tentar ler um objeto de dados que não foi encontrado.
writeNavigatorXml	Define se um esquema de navegação será gravado quando a recuperação de dados é executada. Depois, está disponível na propriedade NavigatorXml.

#### **Valor de retorno**

<> 0 se um erro ocorrer.

**LoadSelection**

Define a referência de seleção a ser usada na próxima recuperação de dados.

**Sintaxe**

LoadSelection(selectionReference)

**Parâmetro**

Parâmetro	Significado
selectionReference	Referência de seleção

**Valor de retorno**

-

**LoadSelectionByName**

Carrega uma seleção por nome.

**Sintaxe**

LoadSelectionByName(string selectionName)

**Parâmetro**

Parâmetro	Significado
selectionName	Nome da seleção

**Valor de retorno**

-

**LoadSelectionByOld**

Carrega uma seleção pela ID de objeto.

**Sintaxe**

LoadSelectionByOld(string selectionOld)

**Parâmetro**

Parâmetro	Significado
selectionOld	ID do objeto da seleção

**Valor de retorno**

-

**WriteXml**

Salva o XML da última coleta de dados em um arquivo.

**Sintaxe**

WriteXml(string fileName, bool prettyXml)

**Parâmetro**

Parâmetro	Significado
Filename	Nome do arquivo no qual o XML será salvo.
Prettyxml	Define se o XML será salvo em um formato legível (recuado).

**Valor de retorno**

<> 0 se um erro ocorreu.

**Exemplo**

```
Selection = ModusProcess.GetComponent("DataSelectionManager")
Selection.GetData(true)
Selection.WriteXml("c:\sel.txt", true)
```

**Script de exemplo DataSelectionManager**

```
;
SelManager = ModusProcess.GetComponent("DataSelectionManager")
;
;-----
;Insert Selection Parameter in the Systempool
System.Partnernumber = "0815"
;
;-----
;Run Selection
Res = SelManager.GetData(False)
;
;-----
;Save result in a file
OutputFileName = "c:\temp\" + System.Partnernumber + ".txt"
SelManager.WriteXml(OutputFileName, true)
;
```

XMLProcessingManager

**Informações gerais**

O componente XMLProcessingManager é necessário se os arquivos XML forem manipulados por um processo.

Os arquivos a serem processados podem conter um ou mais trabalhos de impressão.

### ***Propriedades de XmlProcessingManager***

Propriedade	Significado
FileName	Nome do arquivo XML aberto atualmente: processId + ".wrk"
JobElementPath	Caminho para os conjuntos de dados dentro do arquivo MultiXML, por exemplo, /JOBS/JOB.
LastXmlContent	Dados XML mais recentes
LastOuterXmlContent	<p>Durante o processamento em um único XML, o conteúdo é o mesmo de LastXmlContent.</p> <p>Durante o processamento dos arquivos MultiXML, a propriedade também contém o nó de trabalho.</p> <p><b>Exemplo</b></p> <pre>&lt;JOBS&gt;   &lt;JOB&gt;     &lt;Partner ...&gt;       ...     &lt;/Partner&gt;   &lt;/JOB&gt; &lt;/JOBS )</pre> <p>LastXmlContent:</p> <pre>&lt;Partner .....&gt;   ... &lt;/Partner&gt;</pre> <p>LastOuterXmlContent:</p> <pre>&lt;JOB&gt;   &lt;Partner ...&gt;     ...   &lt;/Partner&gt; &lt;/JOB&gt;</pre>
OriginalFileName	Nome original do arquivo XML

SearchDirectory	Caminho no qual os arquivos XML a serem processados são armazenados, por exemplo D:\Input\Multi\*.xml.
-----------------	--

## ***Métodos de XMLProcessingManager***

### **Cancel**

Ordena que o XmlParser cancele a próxima chamada de NextJob().

### **Sintaxe**

Cancel()

### **Parâmetro**

-

### **Valor de retorno**

-

### **Exemplo**

```
processor.Cancel()
```

### **FileExists**

Pesquisa no diretório especificado como **SearchDirectory** arquivos que contenham dados para processamento em lote.

Se um arquivo for encontrado, ele será aberto e a extensão de arquivo será alterada para **.wrk**.

### **Sintaxe**

FileExists(string processID)

### **Parâmetro**

Parâmetro	Significado
processId	ID do processo

### **Valor de retorno**

True	Arquivo encontrado
False	Arquivo não encontrado

### **Exemplo**

```

UniqueId = UniqueGuid.ToString()
ExecuteTimer = processor.FileExists(UniqueId)

```

### FileExistsEx

Pesquisa o diretório especificado como **SearchDirectory** por arquivos contendo dados para processamento em lote.

Se um arquivo for encontrado, ele é aberto e a extensão do arquivo é mudada para **.wrk**.

### Sintaxe

```
FileExistsEx(string processID)
```

### Parâmetro

Parâmetro	Significado
processId	ID do processo

### Valor de retorno

Se um arquivo for encontrado, todo o conteúdo do arquivo MultiXml é retornado como uma string XML. Se nenhum arquivo for encontrado, uma string vazia é retornada.

### GetJobAttribute

Permite acesso aos atributos do elemento JOB no arquivo em lote XML.

### Sintaxe

```
GetJobAttribute(string attributeName)
```

### Parâmetro

Parâmetro	Significado
attributeName	Nome do atributo

### Valor de retorno

Valor do atributo ou uma string vazia, se o atributo não for encontrado.

### Exemplo

```
System.DocumentCollection = processor.GetJobAttribute("name")
```

### NextJob

Lê o XML de dados do próximo trabalho do arquivo de lote XML. O XML de dados pode, então, acessar o atributo **LastXmlContent**.

### Sintaxe

NextJob()

#### Parâmetro

-

#### Valor de retorno

Valor	Significado
True	O próximo trabalho foi lido.
False	Nenhum próximo trabalho encontrado.

#### Exemplo

```
SYSTEM.DoWork = processor.NextJob()
```

#### StartProcessing

Inicializa o processamento de um arquivo de lote XML.

#### Sintaxe

StartProcessing()

#### Parâmetro

-

#### Valor de retorno

-

#### Exemplo

```
processor = ModusProcess.GetComponent("XmlProcessingManager")
processor.StartProcessing()
```

#### Terminate

Encerra o processamento de um arquivo de lote XML.

#### Sintaxe

Terminate()

#### Parâmetro

-

#### Valor de retorno

-



**Exemplo**

```
processor = ModusProcess.GetComponent("XmlProcessingManager")
processor.StartProcessing()
...
processor.Terminate()
```

**XwsDataSelectionManager****Informações gerais**

Este componente fica disponível somente quando o Xdata tiver sido licenciado corretamente.

Ele fornece recuperação de dados por meio de serviços Web do Xdata e cria um arquivo XML usado para geração de documentos.

**Propriedades de XwsDataSelectionManager**

Parâmetro	Significado
SelectionReference	Seleção a ser executada pelo componente.
WebServiceUrl	URL dos serviços Web Xdata.
LastError	Use isso para recuperar informações de um erro ocorrido durante a execução da coleta de dados.
NavigatorXml	Se o método GetDataExtended for chamado com o parâmetro <b>retrieveNavigateXml = True</b> , o Navigationsxml pode ser acessado aqui.
XmlData	Lê o XML da última coleta de dados como uma string.

**Métodos de XwsDataSelectionManagers****GetData**

Executa recuperação de dados.

**Sintaxe**

GetData(bool raiseInSelVariable)

GetData(string selectionParameters, bool raiseInSelVariable)

**Parâmetro**

Parâmetro	Significado
selectionParameters	Parâmetro de seleção no formato XML

	<p><b>Exemplo:</b></p> <pre>selectionParameters = "&lt;parameters&gt;&lt;parameter name=" "Partnernumber" " value=" "0815" " /&gt;&lt;/parameters&gt;"</pre> <p>Se esse parâmetro não for especificado, os parâmetros de seleção devem ser armazenados no pool do sistema.</p> <p><b>Exemplo:</b></p> <pre>system.Partnernumber = "0815"</pre>
raiseInSelVariable	Define se um erro será lançado se um script de leitura com SelVariable tentar ler um objeto de dados que não foi encontrado.

### Valor de retorno

<> 0 se um erro ocorrer.

### GetDataExtended

Executa a recuperação de dados.

### Sintaxe

GetDataExtended(bool raiseInSelVariable, bool retrieveNavigateXml)

GetDataExtended(string selectionParameters, bool raiseInSelVariable, bool retrieveNavigateXml)

### Parâmetro

Parâmetro	Significado
selectionParameters	<p>Parâmetro de seleção</p> <p><b>Exemplo:</b></p> <pre>selectionParameters = "&lt;parameters&gt;&lt;parameter name=" "Partnernumber" " value=" "0815" " /&gt;&lt;/parameters&gt;"</pre> <p>Se este parâmetro não estiver especificado, os parâmetros de seleção devem ser armazenados no pool do sistema.</p> <p><b>Exemplo:</b></p> <pre>system.Partnernumber = "0815"</pre>
raiseInSelVariable	Define se um erro será lançado se um script de leitura com SelVariable tentar

	ler um objeto de dados não encontrado.
retriveNavigateXml	Grava NavigationsXml na propriedade NavigatorXml True / False

**Valor de retorno**

<> 0 se um erro ocorreu.

**Ping**

Verifica o alcance do serviço Web definido na propriedade WebServiceUrl.

**Sintaxe**

Ping()

**Parâmetro**

-

**Valor de retorno**

"Xdata Runtime Webservice" (nome o usuário que executa o serviço Web)".

**Script de exemplo XwsDataSelectionManager**

```
xws = ModusProcess.GetComponent("XWSDataSelectionManager")
;
pingResult = xws.Ping()
;
; The parameter raiseInSelVar (set to false in all examples) throws an
exception if true,
; i.e. if the selection tries to read a non-existent variable.
;
;-----
;Alternative with Selection Parameter in an XML string
;-----
selectionParameters = "<parameters><parameter name=" "Partnernumber" "
value=" "0815" " /></parameters>"
;no NavigationsXml
rescode = xws.GetData(selectionParameters, false)
data = xws.XmlData
error = xws.LastError
;
;-----
;with NavigationsXML (here 3. Parameter)
;-----
rescode = xws.GetDataExtended(selectionParameters, false, true)
data = xws.XmlData
navigData = xws.NavigatorXml
```

```

error = xws.LastError
;
;-----
;Alternative with Selection Parameter stored in system pool
;no NavigationsXml
;-----
system.Partnernumber = "0815"
rescode = xws.GetData(false)
data = xws.XmlData
error = xws.LastError
;
;-----
;with NavigationsXML (2. Parameter)
;-----
rescode = xws.GetDataExtended(false, true)
data = xws.XmlData
navigData = xws.NavigatorXml
error = xws.LastError
;

```

## Processos de hospedagem

### Instalação de um serviço Windows

[▶ Mostrar tudo](#)

**Siga as etapas abaixo para instalar um serviço Windows para hospedar um processo ModusOne:**

1. Vá para o diretório do programa e copie o arquivo: **ModusSuite.Runtime.WindowsServiceHost.exe** e nomeie o novo arquivo apropriadamente para o processo.
2. Vá para o diretório do programa e copie o arquivo: **ModusSuite.Runtime.WindowsServiceHost.exe.config** e dê à cópia o mesmo nome, com exceção da extensão **.config** como o nome na etapa 1.

#### Exemplo

Copiar ModusSuite.Runtime.WindowsServiceHost.exe	para BatchServiceHost.exe
Copiar ModusSuite.Runtime.WindowsServiceHost.exe.config	para BatchServiceHost.exe.config

**Faça as seguintes alterações no novo arquivo exe.config:**

**Defina o nome do arquivo de log**

Localize a linha começando com

```
'<add name="tracefile_listener" type=...'
```

e especifique o nome do arquivo de log desejado em

```
initializeData="WindowsServiceHostModus_log.txt"
```

.

### Exemplo

```
<add name="tracefile_listener"
type="ModusSuite.Common.SystemFramework.DailyTraceListener,
ModusSuite.Common.SystemFramework" delimiter=";"
initializeData="BatchServiceHost_log.txt" traceOutputOptions="DateTime"/>
```

### Defina o modelo do Word

Procure pelo nó <appSettings>.

Adicione a seguinte linha:

```
<add key="wordtemplate" value="PathToModusOneVerzeichnis\normal.dotm" />
```

A especificação do modelo do Word só é necessária se o processo criar documentos Modus.

### Dica

Quando o ModusOne é instalado, uma variável de ambiente chamada %modusdir% é criada. O valor dessa variável é o caminho para o diretório de instalação do ModusOne.

Se o .dotm normal estiver localizado no diretório de instalação do ModusOne, é possível definir o caminho conforme a seguir:

```
%modusdir%Normal.dotm
```

### Especifique os processos desejados na configuração

Há duas maneiras possíveis de fazer isso, que também podem ser combinadas.

Se as duas alternativas estiverem configuradas, os processos únicos são iniciados e, em seguida, aqueles processos que foram mencionados por meio de uma tabela de alias.

- **Defina os processos únicos**

name	ID de instância (nome do processo)
reference	Referência de repositório Selecione o processo no Navigator, clique nele com o botão direito do mouse para obter o item do menu de contexto <b>Show Object Information</b> e, em seguida, copie a entrada do resultado.

- 
- <modusruntime>
- <runtimeservices>
- ...
- List of run time services

- ...
- </runtimeservices>
- <processes>
- <process name="ModBatch\_MultiXmlFile" reference="ModBatch\_MultiXmlFile|28012011-1-prc-modb|644|Process|None|dm" />
- <process name="ModBatch\_XmlFile" reference="ModBatch\_XmlFile|29012011-13-prc-modb|17|Process|None|modb" />
- <process name="ModBatch\_TableJob" reference="ModBatch\_TableJob|30012011-15-prc-modb|19|Process|None|modb" />
- <process name="MWS\_Assistant" reference="MWS\_Assistant|07022011-14-prc-mws|18|Process|None|mws" />
- <process name="ODIN\_Rendition 1" reference="Odin\_Rendition|19022011-10-prc-odv|2233|Process|None|modstd\_megatron" />
- <process name="ODIN\_Rendition 2" reference="Odin\_Rendition|19022011-10-prc-odv|2233|Process|None|modstd\_megatron" />
- </processes>
- <aliastables>
- ...
- </aliastables>
- </modusruntime>
- Se a definição individual dos processos únicos envolve a execução dos mesmos processos várias vezes, as entradas correspondentes só difere no que diz respeito à ID de instância ("nome" do atributo).

- 
- **Defina as tabelas de alias que contêm processos**

name	Nome da tabela de alias
systemoid	ID do sistema onde a tabela de alias reside

- 
- <modusruntime >
- <runtimeservices>
- ...
- List of run time services
- ...
- </runtimeservices>
- <processes>
- ...

- `</processes>`
- `<aliastables>`
- `<aliastable name="BatchProcesses" systemid="mod5processes"/>`
- `<aliastable name="OdinProcesses" systemid="mod5processes"/>`
- `</aliastables>`
- `</modusruntime>`
- Se mais de uma tabela for configurada, um usuário deve garantir que os aliases não resultem, uma vez que podem levar a uma exceção.
- Como um alias pode conter um número de referências para diferentes objetos, as seguintes regras se aplicam:
  - Apenas os processos normais serão manipulados (por exemplo: nenhum processo MWS ou de outros tipos)
  - Se uma chave contém vários processos, primeiro é dado o nome da chave conforme a ID de instância e todos os outros dão o nome da chave, mais um contador incremental adicional começando com um, por exemplo: 'OdinOnline 1'.
- **Registre o serviço no console de gerenciamento do serviço Windows**

Execute o arquivo em lotes `install_modus_service.cmd` especificando o nome do serviço:

ServiceName	O nome do serviço na administração do serviço
DisplayName	Nome de exibição do serviço no console de gerenciamento do serviço
ImageName	Nome do arquivo .exe, por exemplo: BatchServiceHost.exe
Description	Descrição do serviço na administração do serviço

#### Observação:

Os direitos de administrador local são necessários para executar o arquivo CMD.

#### Execução no Windows XP

O usuário conectado requer os direitos de administrador local.

Se o usuário conectado não tiver direitos de administrador, um prompt de comando pode ser aberto usando os seguintes comandos:

```
runas /User:[AdminUserName] cmd.exe
```

Após inserir a senha para a conta de administrador especificada, uma caixa de comandos, na qual o arquivo CMD pode ser executado, é aberta.

#### Execução no Windows Vista

Devido ao UAC (Controle de acesso do usuário), mesmo um administrador conectado pode não possuir todos os direitos necessários.

O arquivo CMD deve, portanto, ser executado de um prompt da caixa do DOS **Executar como um administrador**.

A caixa DOS pode ser iniciada no menu de contexto do Explorer, ou entrando **cmd**, ou indo a Start/Run.. e pressionando Ctrl+Shift+Enter.

Após confirmar o diálogo UAC, uma caixa DOS, na qual o arquivo CMD pode ser executado com os direitos necessários, abre.

#### Exemplo

```
Install_modus_service "BatchService" "BatchServiceModus"  
"BatchServiceHost.exe" "Modus Tools Service"
```

## Desinstalação de um serviço Windows

Execute o arquivo em lotes **uninstall\_modus\_service.cmd** especificando o nome do serviço.

O nome do serviço é o nome definido pelo parâmetro **ServiceName** durante a instalação.

#### Exemplo

```
uninstall_modus_service.cmd BatchService
```



# Definições de consulta

## Introdução

Definições de consulta permitem que você compile análises de tabelas de bancos de dados que podem ser executadas no [Assistente de consulta](#).

## Criar definições de consulta

**Para criar uma nova definição de consulta, execute estas etapas:**

1. Selecione **Novo -> Definição de consulta** do menu de contexto do Navigator.
2. Digite o nome da definição de consulta na seguinte caixa de diálogo.
3. Selecione um alias de DB na janela do lado direito do editor.  
Após selecionar um alias de DB, uma lista de tabelas que ele contém é exibida.
4. Clique no sinal de adição da tabela da qual os dados devem ser lidos.  
Isso exibe uma lista de campos da tabela.
5. Arraste os campos da tabela que precisa para a janela **Colunas de resultado**.  
Se você precisar de todos os campos em uma tabela, simplesmente arraste a tabela para a janela **Colunas de resultado** na barra inferior da janela do editor.
6. Edite as **propriedades da definição de consulta**.
7. Edite as **propriedades das colunas de resultado**.

**Para adicionar condições à definição de consulta, execute estas etapas:**

1. No menu de contexto da janela do editor, selecione o item do menu **Adicionar nova condição**.
2. Digite o nome da condição na caixa de diálogo seguinte.
3. Edite as **propriedades da condição**.

A condição corresponde à cláusula **WHERE** da instrução SQL.

Você pode inserir quantas condições desejar.

**Para adicionar variáveis em uma condição, execute estas etapas:**

1. No menu de contexto da janela do editor, selecione o item do menu **Adicionar nova variável**.
2. Digite o nome da condição na caixa de diálogo seguinte.
3. Edite as **propriedades das variáveis**.

Cada condição pode conter um número ilimitado de variáveis e todas podem ser usadas na cláusula **WHERE** e inseridas como parâmetros no assistente de consulta.

## Propriedades de uma definição de consulta

### Descrição

Descrição em texto livre.

**Alias de BD**

Alias de BD da consulta executada.

**Tabela de BD**

Corresponde à instrução FROM na caixa de diálogo Selecionar.

**Consulta detalhada**

É possível configurar uma definição de consulta aqui para que ela forneça resultados mais detalhados.

**Categoria**

Categoria da consulta (texto livre)

**Máximo de linhas**

Número de linhas em um resultado de consulta

## Propriedades de colunas de resultados

**Descrição**

Descrição em texto livre.

**Coluna de BD**

Nome do campo de banco de dados.

**Visível**

True significa que a coluna de resultados é exibida no resultado, False significa que ela é oculta.

**Tamanho da coluna**

Largura da coluna no resultado da consulta. É possível inserir qualquer valor aqui.

Se você especificar -1, a largura da coluna será determinada automaticamente, dependendo do conteúdo e dos cabeçalhos de coluna.

**Título**

Título da coluna no resultado da consulta.

## Propriedades de condições

**Descrição**

Descrição em texto livre.

**GroupBy**

Coluna de banco de dados em que o resultado é agrupado.

**OrderBy**

Coluna de banco de dados em que o resultado é classificado.

**Title**

Título de parâmetro

**Where**

Corresponde à instrução WHERE na caixa de diálogo Selecionar.

Se a condição contiver variáveis, elas podem ser usadas da seguinte maneira:

- `PR_TYPE = $VariableName$` ou
- `PR_TYPE = :VariableName`

## Propriedades de variáveis

**Prompt**

Propriedade	Significado
True	A variável é visível na caixa de diálogo de consulta e pode ser editada.
False	A variável não é visível na caixa de diálogo de consulta.

**Title**

Título da variável como exibida no assistente de consulta.

**TypeName**

Tipo da variável (texto livre)

**DefaultValue**

Valor padrão de entrada do parâmetro.

Também é possível usar os valores no formato `[ModusUser.AttributeName]` aqui.

**ValueHelp**

Nome da ajuda de valor atribuída à variável.

# XData

## Introdução

O Xdata é o módulo de recuperação de dados do ModusOne e realiza todas as atividades de recuperação de dados.

Os dados podem ser extraídos de vários bancos de dados ou lidos do SAP e disponibilizados para outros componentes como XML

O tempo de execução do Xdata fornece uma framework para leitura dos metadados dos bancos de dados e dos sistemas SAP configurados. Esses metadados são usados para criar seleções e objetos de dados.

Um compilador integrado converte essa seleção de dados em um assembly .NET. Esse assembly contém o fluxo de processo completo de uma seleção, ou seja, toda a sequência de operações individuais de leitura de dados.

Os seguintes fornecedores de acesso a bancos de dados são suportados:

- ODP
- OLEDB
- ORAMS
- MSSQL

As operações de recuperação de dados para seleções SAP usam o SAP .NET Connector 2.0.

## Objetos de dados

### Objetos de dados

Os objetos de dados são visualizações lógicas dos dados físicos e contêm atributos 1-n (variáveis de objetos de dados).

Os objetos de dados são autônomos em relação aos demais.

### Criar objetos de dados

Objetos de dados podem ser criados apenas quando o Xdata foi licenciado corretamente.

**Para criar um novo objeto de dados, execute estas etapas:**

1. Selecione **Novo -> Objeto de dados** do menu de contexto do Navigator.  
O assistente de objeto de dados o guiará pelas etapas para a criação de um objeto de dados.
2. Na primeira janela, selecione **Criar um novo objeto de dados**.
3. Na janela seguinte, selecione o alias de banco de dados necessário do menu suspenso **Conexão de alias**.  
Uma lista de todas as tabelas associadas a esse alias de DB é exibida.  
Para conexões SAP, a lista exibe todos os BAPIs disponíveis.
4. Selecione as tabelas das quais deseja criar os novos objetos de dados.  
Use o item do menu de contexto **Renomear** para renomear os objetos de dados.

5. Clique no sinal de adição na frente de uma das tabelas.  
Selecione os campos de dados que o objeto de dados deve conter.
6. A próxima janela resume a seleção feita.  
Pressione o botão **Criar objetos de dados** para iniciar a criação de objetos de dados.
7. Caso mais objetos de dados precisem ser criados, pressione o botão **Voltar**.
8. Ao terminar, feche o assistente.  
Todos os novos objetos de dados agora estão visíveis no Navigator.

## Editar objetos de dados

### Editar objetos de dados

**Para editar um objeto de dados, execute estas etapas:**

1. Abra o objeto de dados.
  - A estrutura do objeto de dados é exibida no painel à esquerda. Quando aplicável, todos os objetos subordinados de primeiro nível também são exibidos.
  - As variáveis do objeto de dados selecionado são exibidas no painel superior direito.
  - O painel inferior direito exibe uma lista de scripts de leitura atribuídos a um objeto de dados selecionado.
2. Crie os **scripts de leitura** obrigatórios.
3. Modifique a **consulta SQL** do script de leitura.
4. Se necessário, crie quaisquer **SQL** adicionais opcionais.
5. Atribua um **script de leitura** ao objeto de dados.

### Criar scripts de leitura

Todos os objetos de dados necessitam de um script de leitura que seja executado no tempo de execução.

**Para criar um novo script de leitura, execute estas etapas:**

1. Abra o objeto de dados.
2. Selecione o objeto de dados superior.
 

**Observação:**  
para criar scripts de leitura para objetos de dado filhos, abra o objetos de dado filho respectivo em sua própria janela do editor.
3. Vá para a guia **Ferramentas do editor de Xdata** e clique no botão **Novo script de leitura**.
4. Digite o nome do script de leitura na caixa de diálogo seguinte.
5. Abra o script de leitura criado recentemente com um clique duplo.  
O script de leitura já contém duas instruções padrão e já é totalmente funcional:

<code>db.WorkWithQuery("Q1")</code>	Executa a consulta de SQL especificada
-------------------------------------	--

	("Q1")
db.ApplyDataSet( )	Aplica o conjunto de dados de leitura ao objeto de dados

Você pode encontrar uma visão geral das funções adicionais do script de leitura no capítulo sobre Funções de Xdata.

Na criação do script de leitura, uma consulta de SQL **Q1** foi gerada, mas necessita de modificação. Essa consulta é usada para especificar a instrução de SQL definida para ler os campos de dados.

### Funções no menu de contexto dos scripts de leitura

#### Excluir script

Exclui o script.

#### Novo script

Cria um novo script e abre-o em uma janela do editor.

#### Modificar uma consulta SQL

#### Conclua as seguintes etapas para modificar uma consulta SQL:

1. Clique na janela **Ativar scripts** na **C1**.  
A instrução de consulta é exibida na janela abaixo do script de leitura.
2. Conclua a condição Where.  
A cláusula Where normalmente contém **variáveis** que são preenchidas com valores no tempo de execução.  
Uma variável deve ser definida por uma anterior por dois pontos.

#### Exemplo

```
select * from BANKS
where BNK_BLZ=:KTO_BLZ
```

A variável é substituída no tempo de execução de acordo com a seguinte regra:

- Se o objeto de dados principal contiver um campo que corresponde ao nome da variável, o valor é usado.
- Se nada for encontrado, o **pool de sistema** é verificado para ver se há variáveis desse nome e o valor correspondente usado.

Consultas SQL adicionais podem ser criadas por meio do menu de contexto da janela.

#### Criar consultas SQL adicionais

#### Para criar consultas SQL adicionais, siga estas etapas:

1. Na janela **Scripts ativos**, selecione o item do menu de contexto **Criar novo SQL**
2. Digite o nome da consulta SQL na caixa de diálogo seguinte.

3. Complete a condição **WHERE**.
4. Edite a instrução no script de leitura:

```
db.WorkWithQuery("Q1")
```

Substitua o parâmetro **Q1** com o nome da nova consulta SQL.

### Atribuir scripts de leitura

**Para atribuir um script de leitura, execute estas etapas:**

1. Vá para a janela do editor do objeto de dados.
2. Arraste o script de leitura da janela inferior direita e solte-o no objeto de dados. Como alternativa, atribua o script de leitura pela propriedade **Active Reading Script** no object inspector.

### Inserir variáveis

**Conclua as seguintes etapas para inserir uma variável em um objeto de dados:**

1. Abra o objeto de dados.
2. Clique em **Ferramentas do editor de Xdata** e clique no botão **Nova variável**.
3. Digite o nome da condição na caixa de diálogo seguinte.

### Funções no menu de contexto das variáveis de objeto de dados

#### Renomear

Abre uma janela de diálogo para renomear as variáveis de objeto de dados.

#### Excluir variável

Remove a variável do objeto de dados.

#### Nova variável

Insera uma nova variável no objeto de dados.

### Inserir objetos filhos

Um objeto filho pode conter objetos filhos com os quais ele mantém um relacionamento de detalhes mestre.

**Conclua as seguintes etapas para inserir objetos filhos em um objeto de dados:**

1. Abra o objeto de dados em que um objeto de dados está para ser inserido.
2. Arraste o objeto de dados necessário que funcionará como um objeto filho do Navigator no painel à esquerda do editor.
3. No Object Inspector, use a propriedade **RelationType** para definir o tipo de relacionamento.
  - Um relacionamento **One2One** é aquele que existe entre parceiro e título, ou seja, cada parceiro tem apenas um título.
  - Um relacionamento **One2Many** é aquele que existe entre parceiro e endereço, ou seja, cada parceiro pode ter mais de um endereço.
4. Um campo de chave para os dois objetos é necessário. Essas chaves são usadas para definir o relacionamento entre os dois objetos.
 

Atribua os campos de chave da seguinte maneira:

  - Selecione o campo de chave de um objeto principal e arraste-o até a visualização da estrutura à esquerda.
  - Faça o mesmo com os objetos filhos.

Agora, os dois campos são definidos na visualização da estrutura do objeto filho (a chave do primeiro objeto principal, a

chave do segundo filho).

Esses campos são registrados como **Chave** e **Chave-mestra** nas propriedades do objeto filho.

As chaves de relacionamento também podem conter mais de um par de campo.

5. Insira objetos filhos adicionais necessários, como opção.

### Aninhamento da profundidade de objetos filhos

Os objetos podem ser aninhados dentro de uns dos outros em qualquer profundidade.

Para um desempenho melhor, o editor apenas exibe o próximo nível de aninhamento para cada objeto.

#### Exemplo:

O objeto de dados **Parceiro** tem o objeto filho **Contas**.

O objeto de dados **Contas** tem o objeto de dados Bancos atribuído como um objeto filho.

Abra o objeto de dados **Contas** e atribua o objeto filho **Bancos** como descrito acima.

No objeto de dados **Parceiro**, o objeto filho **Bancos** não está visível.

### Funções no menu de contexto dos objetos filhos

#### Exclusão do link

Remova o objeto filho do objeto de dados.

#### Abertura de objeto de dado

Abra o objeto filho em uma nova guia.

### Nós de resumo

Os nós de resumo são uma função especial dos objetos de dados filhos.

Eles permitem você agrupe conjuntos de dados, por exemplo, de objetos de dados filhos, que devem ser disponibilizados imediatamente na seleção.

#### Exemplo

O objeto de dados **Partner** contém o objeto filho **Accounts**, que por sua vez contém o objeto filho **Banks**.

Todos os dados de Banks lidos devem ser disponibilizados diretamente na seleção.

#### Execute as seguintes etapas para inserir o objeto Banks como um nó de resumo no objeto Partner:





1. Abra o objeto de dados Partner.
2. Arraste o objeto de dados Banks do Navigator e solte-o na janela do editor.
3. Na propriedade **Masterkey** do objeto filho **Banks**, insira o caractere **'\*'**.  
O objeto filho Banks agora é definido como um nó de resumo.
4. Na propriedade **Key** do objeto filho **Banks**, insira o campo de tabela pelo qual as entradas podem ser identificadas com exatidão, por exemplo, o campo que contém o número de roteamento.
5. Os nós de resumo não precisam de um processo de leitura.

A seleção lê todas as Accounts de Partner e, para cada Account, os dados de Bank correspondentes.

O nó de resumo **Banks** contém todos os Banks lidos pela seleção.



## Ferramentas do editor de Xdata

	<b>Atualizar</b>
	Atualiza o objeto de dados. Essa função pode ser executada apenas quando nenhuma alteração foi feita na seleção ou quando qualquer alteração feita foi salva.
	<b>Nova variável</b>
	Cria uma nova variável de objeto de dados.
	<b>Novo script de leitura</b>
	Cria um novo script de leitura
	<b>Abrir objeto de dados</b>
	Abre o objeto de dados selecionado

## Objetos de dados vazios

### Criar objetos de dados vazios

Objetos de dados vazios podem ser criados apenas quando o Xdata foi licenciado corretamente.

Objetos de dados vazios referem aos objetos de dados que não são associados diretamente a uma fonte de dados.

#### Para criar um objeto de dados vazio, execute estas etapas:

1. Selecione **Novo -> Objeto de dados** do menu de contexto do Navigator.  
O assistente de objeto de dados o guiará pelas etapas para a criação de um objeto de dados.
2. Na primeira janela, selecione **Criar um novo objeto de dados vazio**.
3. Digite o nome do objeto de dados na caixa de diálogo seguinte.
4. No Navigator, clique no botão **Criar objeto de dados**.

### Editar ou excluir conjuntos de dados

Objetos de dados vazios podem ser usados para editar ou excluir conjuntos de dados.

**Execute as seguintes etapas:**

1. Abra um objeto de dados vazio.
2. Crie um novo **script de leitura**.
3. Altere o script de leitura da seguinte maneira:
 

```
db.StartTransaction()
db.ExecuteQuery("Q1")
db.Commit()
```
4. Altere a **consulta SQL** para uma instrução de atualização ou exclusão.

**Exemplo**

```
Update Partner set LastSaved = sysdate where par_id = :PAR_ID
```

**Executar um procedimento armazenado em um objeto de dados****Para executar um procedimento armazenado em um objeto de dados, execute estas etapas:**

1. Crie um objeto de dados vazio.
2. Crie o mesmo número de variáveis no objeto de dados do número de parâmetros para o procedimento armazenado.
3. Digite o nome do parâmetro respectivo na propriedade **DataLink**, na caixa de diálogo seguinte.
4. Crie um novo **script de leitura**.  
O nome do script deve corresponder ao nome do procedimento armazenado.
5. Altere o script de leitura da seguinte maneira:
 

```
System.Inputparam = System.Partnernummer; Inputparam of the
StoredProcedure
db.StartProcedure("ProcedureName", 0)
db.ApplyDataSet()
```
6. Exclua a instrução de seleção da consulta de SQL e, por último, defina o tipo de parâmetro.

**Exemplo:**

```
Inputparam=IS
Outputparam1=OS
Outputparam2=OI
Outputparam3=O@
```

O primeiro caractere define o tipo de parâmetro:

Parâmetro	Definição
I	Parâmetro de entrada
O	Parâmetro de saída
B	Parâmetro de entrada/saída
R	Valor de retorno

O segundo caractere define o tipo de campo de dados:

Parâmetro	Definição
S	String
I	Número inteiro
F	Flutuante
D	Data
T	Hora
@	Data/hora

## Atualizar objetos de dados

Se a estrutura de um tabela for alterada, os objetos de dados correspondentes que acessam a tabela devem ser atualizados.

### Execute as seguintes etapas para atualizar os objetos de dados:

1. Selecione um ou mais objetos de dados no Navigator.
2. Selecione **Novo -> Objeto de dados** no menu de contexto.
3. No assistente de objeto de dados, selecione a opção **Atualizar objetos de dados**.
4. Você pode verificar ou alterar sua seleção na página seguinte.
5. Sua seleção final é resumida na última página.  
Inicie o processo de atualização com o botão **Atualizar objetos de dados**.

## Propriedades de objeto de dados

### Script de leitura ativo

Nome do script de leitura ativo

### Description

Texto livre.

### DataLink

Nome da tabela do banco de dados

### DataSource

Tipo de DataSource (banco de dados/SAP).

**DBAlias**

Alias de DB associado.

**FieldRange**

O texto **[FIELD RANGE]** (deve ser inserido em letras maiúsculas) na instrução de SQL será substituído com esse texto antes da execução.

**Exemplo**

O SQL do objeto de dados é:

```
select [FIELD RANGE] from Contract where Number = :ContractNumber
```

"ContractId, Number" foi inserido para a propriedade **FieldRange** .

O SQL executado no tempo de execução é:

```
select ContractId, Number from Contract where Number = :ContractNumber
```

Esse recurso foi projetado especificamente para especificar as colunas (campos) de uma seleção, mas também pode ser usado para substituir outras partes do SQL, como o nome de uma tabela, por exemplo:

```
select * from [FIELD RANGE] where Number = :ContractNumber
```

**IsActive**

Define se o objeto de dados está ativo.

Essa propriedade está disponível apenas para objetos de dados filhos.

**Coluna(s) Masterkey**

Chave(s) mestre(s) do relacionamento mestre-detalle dos objetos de dados principal.

**RelationType**

Tipo de relacionamento:

**One2One**: um exemplo de relacionamento One2One é o existente entre parceiro e título, ou seja, cada parceiro tem apenas um título.

**One2Many**: um exemplo de relacionamento One2Many é o existente entre cada parceiro e endereço, ou seja, cada parceiro tem mais de um endereço.

Essa propriedade está disponível apenas para objetos de dados filhos.

**Coluna(s) Key**

Chave(s) do relacionamento mestre-detalle.

**Propriedade da variável do objeto de dados****DataLink**

Nome do campo de dados no banco de dados

**Fieldlength**

Comprimento do campo de dados.

### IsContainerColumn

Defina essa propriedade como **True** quando as variáveis do objeto de dados serão usadas para armazenar valores de outra coluna de banco de dados que contenha um ou mais pares de valores separados um do outro por um caractere fixo.

### Exemplo

A coluna do banco de dados **CustomerInfo** será usada para armazenar o nome e sobrenome de uma pessoa. O conteúdo tem a estrutura geral de

```
FirstName=John,LastName=Doe
```

As 2 entradas são separadas por uma vírgula. Se você deseja atribuir automaticamente **FirstName** e **LastName** a uma variável de objeto de dados, siga as etapas abaixo

1. Crie as variáveis de objeto de dados **FirstName** e **LastName**.

#### Observação:

valor do atributo **DataLink** deve ser idêntico ao nome da chave na coluna do banco de dados (nesse exemplo, **FirstName** ou **LastName**)

2. Defina a propriedade **IsContainerColumn** para as duas variáveis como **True**.
3. No script de leitura, use o método **DefineContainer** para definir a coluna e o separador:
 

```
db.WorkWithQuery("Q1")
db.DefineContainer(0, " CustomerInfo ", "44"); 44 = ASCII-Code for the comma
db.ApplyDataSet()
```

### Consulte também:

[DefineContainer](#)

## Seleções

### Seleções

As seleções contêm o objeto de dados raiz e qualquer quantidade de objetos filhos. Sua principal função é a recuperação de dados.

As [Seleções ativas](#) leem dados de um banco de dados ou SAP.

As [Seleções XSD](#) leem dados de um arquivo XML.

### Criar seleções ativas





Criar uma seleçãoSeleções ativas podem ser criadas quando o Xdata foi licenciado corretamente.



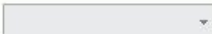
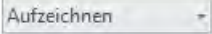

#### Para criar uma nova seleção, execute estas etapas:

1. Selecione **Novo -> Seleção** do menu de contexto do Navigator.
2. Digite o nome da seleção na caixa de diálogo seguinte.
3. Insira o objeto de dados obrigatório.

4. Ative o objeto de dados e os objetos filho obrigatórios.
5. Para usar a seleção, você deve compilá-la primeiramente.  
 Compile a seleção com o botão 'Compilar seleção' na aba Ferramentas do editor de Xdata.  
 A seleção também é compilada ao salvá-la.  
 Mensagens de êxito ou falha são registradas na janela de saída.

### Funções na aba Ferramentas do editor de Xdata

	<p><b>Atualizar</b></p> <p>Atualiza os objetos de dados na seleção.</p> <p>Essa função pode ser executada apenas quando nenhuma alteração foi feita na seleção ou quando qualquer alteração feita foi salva.</p>
	<p><b>Compilar seleção</b></p> <p>Compila a seleção.</p>
	<p><b>Abrir objeto de dados</b></p> <p>Abre o objeto de dados selecionado.</p>
	<p><b>Exportar esquema XML</b></p> <p>Consulte Salvar seleção como um</p>

	esquema
	<p><b>Testar uma seleção</b></p> <p>Consulte Testar uma seleção</p>
	<p><b>Depurar seleção</b></p> <p>Consulte Depurar uma seleção</p>
	<p><b>Conjuntos de teste</b></p> <p>Consulte Conjuntos de teste</p>
	<p><b>Modo decorador</b></p> <p>Consulte Gravação de recuperação de dados</p>
<input checked="" type="checkbox"/> Raise in SelVariable	<p><b>SelVariable</b></p> <p>Consulte Objetos de dados ou atributos ausentes</p>
	<p><b>Mostrar dados</b></p> <p>Use o botão 'Salvar XML' na caixa de diálogo seguinte para salvar os resultados em um arquivo</p>

	XML.
--	------

## Inserir objetos de dados

**Conclua as seguintes etapas para inserir um objeto de dados em um seleção:**

1. Abra a seleção.
2. Arraste o objeto de dados obrigatório do Navigator e arraste-o na seleção.  
Todos os objetos filhos do objeto de dados principal são automaticamente transferidos para a seleção.

### **Observação:**

A inserção ou exclusão de objetos filhos de um objeto de dados pode ser feita somente na janela do próprio editor do objeto de dados.

## **Menu de contexto do objeto de dados**

### **Abrir objeto de dados**

Abre o objeto de dados em uma nova guia.

### **Remover os objetos de dados**

Remove o objeto de dados da seleção.

### **Ativo**

Ativa / desativa o objeto de dados.

## **Menu de contexto dos objetos filhos**

### **Abrir objeto de dados**

Abre o objeto filho em uma nova guia.

### **Ativo**

Ativa / desativa o objeto de dados filho.

### **Atualizar**

Atualiza o objeto de dados.

## **Ativar objetos de dados**

Depois que um objeto de dados é inserido em uma seleção, ele é primeiramente desativado.

Nenhum dado é lido para objetos de dados desativados.

Todos os objetos filhos também são desativados. Um objeto de dados desativado é identificado por uma cruz vermelha.

Para desativar um objeto de dados, use o botão direito do mouse para acessar o menu de contexto e alterne o item de menu **Ativo** ou defina a propriedade **IsActive** como **true**.

A ativação ou a desativação de objetos de dados também pode ser feita usando o **botão esquerdo do mouse** e pressionando a **tecla CTRL**.



Cada um dos objetos filhos pode ser ativado individualmente. É possível desativar um objeto usando o mesmo procedimento.

Os objetos ativos são identificados pelo símbolo de objeto de dados.

### Atualizar objetos de dados

Após a alteração de um objeto de dados, as seguintes etapas devem ser executadas:

1. Salve o objeto de dados modificado.
2. Feche a janela do editor de seleção, se estiver aberta.
3. Reabra a janela do editor de seleção
4. Compile a seleção para assegurar que todas as alterações feitas no objetos de dados estão registradas na seleção.  
Se a seleção for executada antes de ser compilada, o código compilado já armazenado no repositório, e ainda não atualizado com as alterações do objeto de dados, é executado.
5. Execute a seleção.

Com a ajuda do item do menu de contexto **Atualizar**, você pode atualizar as propriedades de **Key** e **Masterkey** do objeto de dados selecionado.

### Editar interface de seleção

A interface de seleção é usada para especificar os parâmetros iniciais da seleção.

**Para editar uma seleção de interface, execute estas etapas:**

1. Abra a seleção.
2. No Object Inspector, clique na propriedade **Parâmetro inicial**.
3. Agora, clique no botão com reticências ao lado da propriedade.
4. Insira os parâmetros iniciais na caixa de diálogo seguinte.
  - O **primeiro** parâmetro contém o nome da chave.
  - O **segundo** parâmetro define o texto do prompt.
  - Defina um valor padrão para a entrada no **terceiro** parâmetro. Esse parâmetro é opcional.
  - Use o **quarto** parâmetro para especificar uma máscara de entrada. Esse parâmetro também é opcional.

#### Exemplo:



```
Partnernumber="Partner-Nr. ", "0815", "####"
```

### Testar seleções

#### **Executar uma seleção**


**Para executar uma seleção, execute estas etapas:**

1. Abra a seleção.

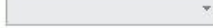
2. Vá para a guia **Ferramentas do editor de Xdata** e clique no botão **Testar seleções** .
3. Digite o(s) parâmetro(s) de seleção na caixa de diálogo seguinte.
4. Vá para a guia **Ferramentas do editor de Xdata** e clique no botão **Mostrar dados** .
5. Clique nos objetos de dados individuais na caixa de diálogo seguinte para mostrar os dados lidos deles.  
Ao pressionar o botão **Salvar XML**, você pode salvar os resultados em um arquivo XML local.

### ***Depurar uma seleção***

**Para depurar uma seleção, execute estas etapas:**

1. Abra a seleção.
2. Vá para a guia **Ferramentas do editor de Xdata** e clique no botão **Depurar seleção** .
3. Digite o(s) parâmetro(s) de seleção na caixa de diálogo seguinte.
4. Depure a seleção usando os botões na guia **Ferramentas do editor de Xdata**.

### ***Conjuntos de teste***

No menu suspenso **Conjuntos de teste**  é possível selecionar um dos conjuntos de teste disponíveis (conjuntos de teste contêm parâmetros de seleções usadas com frequência).

Para usar essas funcionalidade, o arquivo **xdata.config** deve conter o atributo **testsetsfile=**.

```
<xdata prettyxml="true" raise_on_dead_link="true" decoratorprotocolpath=""
keyfile="" writelastcode="true" testsetsfile="xdatatestsets.xml">
```

O arquivo **xdatatestsets.xml** contém parâmetros de seleções usadas com frequência.

Se nenhum caminho for especificado para o arquivo, ele deve estar no mesmo diretório de **xdata.config** que, via de regra, é o Bin-Directory.

#### **Estrutura do arquivo xdatatestset.xml**

```
<xdatatests>
  <selections>
    <selection name="Partner_Letter">
      <testsets>
        <testset systemoid="dm" name="Partner-0815" desc="Selection
parameter für den Partner 0815">
          <parameters>
            <parameter name="Partner Number" value="0815" />
          </parameters>
        </testset>
      </testsets>
    </selection>
  </selections>
</xdatatests>
```

```

    </selections>
</xdatatests>

```


Atributos do elemento <selection>	Definição
name	Nome da seleção à qual as definições se aplicam. Se name="*" for definido, as definições seguintes se aplicam a todas as seleções.

Atributos do elemento <testset>	Definição
Systemold	Nome do sistema no qual o teste estará disponível. Se este atributo estiver ausente, o conjunto de testes estará disponível em todos os sistemas.
name	Nome do conjunto de testes como mostrado na máscara de entrada.
desc	Descrição

Atributos do elemento <parameter>	Definição
name	Nome dos parâmetros da seleção.
value	Valor passado ao parâmetro.



### ***Gravar e reproduzir recuperação de dados***

**Execute as etapas a seguir para gravar uma recuperação de dados:**

1. Abra a seleção.
2. Vá para a guia **Ferramentas do editor de Xdata** e selecione no menu suspenso o item **Gravar**.
3. Vá para a guia **Ferramentas do editor de Xdata** no botão **Testar seleções** .
4. Insira o nome do arquivo gravado na caixa de diálogo seguinte e clique no botão **Salvar**.  
Os dados lidos são salvos em um arquivo XML.

Esse arquivo pode ser usado para testes em outro computador usando a opção **Reproduzir**.

**Execute as etapas a seguir para reproduzir uma recuperação de dados:**

1. Abra a seleção.
2. Vá para a guia **Ferramentas do editor de Xdata** e selecione no menu suspenso o item **Reproduzir**.
3. Vá para a guia **Ferramentas do editor de Xdata** no botão Testar seleções .
4. Selecione um arquivo XML gravado previamente na caixa de diálogo seguinte.
5. Vá para a guia **Ferramentas do editor de Xdata** no botão **Mostrar dados** .
6. Clique nos objetos de dados individuais na caixa de diálogo seguinte para exibir os dados lidos a partir deles.  
Ao pressionar o botão **Salvar XML**, você pode salvar os resultados em um arquivo XML local.

### **Objetos de dados ou atributos ausentes**

Ative a opção  **Raise in SelVariable** para acionar uma mensagem de erro quando a seleção contiver um objeto de dados ou atributo que não foi encontrado no banco de dados.

### Salvar seleção como um esquema

Ao pressionar o botão **Exportar esquema XML**, você pode salvar a seleção como um arquivo XSD.

### Criar seleções XSD

#### Criar seleções XSD

Essa função cria seleções a partir dos arquivos XSD disponíveis.

**Para criar uma seleção XSD, execute estas etapas:**

- Selecione **Novo -> Seleção XSD** do menu de contexto do Navigator.
- Digite o nome da seleção XSD na caixa de diálogo seguinte.
- Vá para a guia **Ferramentas do editor de Xdata** e clique no botão **Importar seleção XSD**.
- Selecione o arquivo XSD necessário na janela de diálogo seguinte.

O painel superior mostra todos os objetos de dados contidos na seleção.

O painel inferior mostra os atributos do objeto de dados selecionado.

### Mostrar ComplexType

Ative a opção **Mostrar ComplexType** na guia **Ferramentas do editor de Xdata** para exibir os tipos complexos de primeiro nível.

### Construções de esquema XSD suportadas

### **Regras para objetos de dados**

[▶ Mostrar tudo](#)

Um esquema XSD construído é reconhecido como um objeto de dados quando uma das seguintes regras se aplicam:

Se for um tipo de nó

- xs:sequence
- xs:all or
- xs:choice

as regras são aplicadas aos nós filhos.

### Regra 1: Nó xs:complexType

Um nó xs:complexType é reconhecido como um objeto de dados.

```
<xs:schema ... ..
...
<xs:complexType name="MType">
  <xs:attribute name="MAttribute" type="xs:string"/>
</xs:complexType>
...
</xs:schema>
```

### Regra 2: Nó xs:element com subnó xs:complexType

Um nó xs:element é reconhecido como um objeto de dados se tiver um nó filho xs:complexType .

O nome do objeto de dados é especificado em um dos dois nós no nome do atributo.

Se o nó xs:element já tiver o nome do atributo definido, o atributo em um nó xs:complexType definido como nome é ignorado.

```
<xs:schema ... ..
...
<xs:element name="X">
  <xs:annotation>
    <xs:documentation>This is X Element</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:attribute name="XAttribute" type="xs:string"/>
  </xs:complexType>
</xs:element>
...
</xs:schema>
```

### Regra 3: Nó xs:element com xs:complexType como tipo

Um nó xs:element é reconhecido como um objeto de dados se tiver como tipo um nó xs:complexType.

```
<xs:schema ... ..
...
<xs:element name="X" type="Xtype" >
  <xs:annotation>
    <xs:documentation>This is X Element</xs:documentation>
  </xs:annotation>
```

```

</xs:element>
...
<xs:complexType name="XType">
  <xs:attribute name="XAttribute" type="xs:string"/>
</xs:complexType>
...
</xs:schema>

```

#### Regra 4: Nó `xs:element` se refere a um elemento de objeto de dados

Um nó `xs:element` é reconhecido como um objeto de dados, se a referência é relativa ao nó `xs:element` que na aplicação de uma das regras em si seja um objeto de dados.

```

<xs:schema ... ..
...
  <xs:element ref="X" >
    <xs:annotation>
      <xs:documentation>This is X Element</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element ref="Y" />
...
<xs:element name="X">
  <xs:annotation>
    <xs:documentation>This is X Element</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:attribute name="XAttribute" type="xs:string"/>
  </xs:complexType>
</xs:element>
...
<xs:element name="Y">
  <xs:complexType>
    <xs:attribute name="YAttribute" type="xs:string"/>
  </xs:complexType>
</xs:element>
...
</xs:schema>

```

Se um objeto de dados foi reconhecido por uma dessas aplicações de regras, o atributo opcional `maxOccurs` é usado para detectar o tipo de relacionamento:

Valor	Significado
<code>maxOccurs</code>	Tipo de relacionamento
<code>&gt;1</code> or <code>"Unbounded"</code>	1:N

otherwise

1:1

## Regras para atributos

[▶ Mostrar tudo](#)

Uma construções de esquema XSD é reconhecida como um objeto de dados quando uma das seguintes regras se aplica:

Se o tipo de atributo é especificado no nó a ser verificado, a primeira verificação é para determinar se o nó é `xs:complexType`.

Se for, as regras são aplicadas a esse nó. Se não for, as regras são aplicadas ao nó originalmente especificado.

### Regra 1: O nó `xs:attribute` é um subnó de um nó `xs:complexType`

Um nó `xs:attribute` é reconhecido como um atributo, quando é um nó filho de um nó `xs:complexType`.

```
<xs:schema ... ..
...
<xs:complexType name="MType">
  <xs:attribute name="MAttribute" type="xs:string"/>
</xs:complexType>
...
</xs:schema>
```

### Regra 2: O nó `xs:element` tem uma das ref. de atributos, tipo ou nome especificado

Um nó `xs:element` é reconhecido como um atributo, quando um dos seguintes critérios na ordem especificada é atendido

#### Regra 2.1

Se no nó `xs:element` a ref. de atributo for especificada, ele é reconhecido como um atributo, quando o nó mencionado é reconhecido como um atributo depois da aplicação das regra 1 ou 2.

```
<xs:schema ... ..
...
  <xs:element ref="Y" />
.....
<xs:element name="Y">
  <xs:complexType>
    <xs:attribute name="YAttribute" type="xs:string"/>
  </xs:complexType>
</xs:element>
...
</xs:schema>
```

Se esse não for o caso, a regra 2.2 é verificada.

### Regra 2.2

Se no nó `xs:element` o tipo de atributo já estiver especificado, ele é reconhecido como um atributo, quando o tipo especificado for `xs:simpleType` ou `xs:complexType` com `xs:simpleContent`.

```
<xs:schema ... ..
  <xs:element name="X" type="Xtype" />
  <xs:element name="Y" type="Ytype" />
  ...
<xs:simpleType name="Xtype">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="11"/>
    <xs:enumeration value="29"/>
  </xs:restriction>
</xs:simpleType>
...
<xs:complexType name="Ytype">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="prompt" type="xs:string" />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
...
</xs:schema>
```

Se esse não for o caso, a regra 2.3 é verificada.

### Regra 2.3

Se no nó `xs:element` o nome do atributo for especificado, ele é reconhecido como um atributo, se ele contiver um nó filho `xs:simpleType` ou um nó filho `xs:complexType` com `xs:simpleContent`.

```
<xs:schema ... ..
  ...
  <xs:element name="X" >
    <xs:simpleType name="Xtype">
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="11"/>
        <xs:enumeration value="29"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element >
  ...
  <xs:element name="Y" >
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="prompt" type="xs:string" />
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element >
  ...
</xs:schema>
```



```
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element >
  ...
</xs:schema>
```

Se nenhuma das regras se aplicar, o nó não é reconhecido como um atributo.

Se nenhum atributos for reconhecido e o nó for

- xs:sequence
- xs:all, xs:choice
- xs:simpleContent or
- xs:extension

as regras são aplicadas aos nós filhos.

Se uma das regras detectar um atributo, o **nome** do atributo do nó ou a **ref.** do atributo de um nó mencionado é usado para determinar o nome do atributo.

## Propriedades de seleção

### Parâmetro inicial

Parâmetro inicial da seleção

Para obter mais informações, consulte o capítulo sobre [Editar interface de seleção](#).

# Odin

## Introdução

O Odin (interface de documento aberto) é uma solução para toda a empresa para processos de impressão complexos.

Documentos podem ser impressos em vários tipos de mídia.

Portanto, por exemplo, seus documentos podem ser impressos, enviados como e-mail ou enviados por fax.

O Odin também pode desempenhar as operações mais complexas de impressões centrais, como enchimento de envelopes e pós-processamento.

O Odin é uma parte integral do fluxo de produção de documentos completos ModusOne, desempenhando todas as tarefas relacionadas à impressão e ao gerenciamento de saída.

## Administração

Use o item **Administração** da barra de ferramentas para exibir visualizações do Odin para processos impressos, pilhas, trabalhos e envelopes abertos.

## Envelopes

Os envelopes só podem ser criados quando o Odin foi licenciado corretamente.

Permite que um ou mais documentos do ModusOne sejam empacotados juntos após a impressão.

**Para criar um envelope, execute estas etapas:**

1. Selecione **Novo -> Envelope** do menu de contexto do Navigator.
2. Digite o nome do envelope na caixa de diálogo seguinte.
3. No campo **Peso em gramas**, insira o peso do envelope.
4. No campo **Descrição**, insira uma descrição de texto livre opcional.

Os envelopes disponíveis são exibidos no painel inferior da janela do editor.

## Documentos anexos

Os documentos anexos só podem ser criados quando o Odin foi licenciado corretamente.

Selecione **Novo -> Documento anexo** do menu de contexto do Navigator e insira o nome do documento anexo na caixa de diálogo seguinte.

Um documento anexo pode ser alimentado em um envelope por uma regra controlada por dados ou especificado em uma definição de postagem.

**Para criar um documento anexo, execute estas etapas:**

1. Selecione **Novo -> Documento anexo** do menu de contexto do Navigator.
2. Digite o nome do documento anexo na caixa de diálogo seguinte.
3. No campo **Peso em gramas**, insira o peso do anexo.

4. No campo **Descrição**, insira uma descrição de texto livre opcional.
  5. No campo **Bandeja**, selecione a bandeja da qual o documento anexo será inserido.
- Os documentos anexos disponíveis são exibidos no painel inferior da janela do editor.

## Definições de postagem

[▶ Mostrar tudo](#)

A definição de postagem pode ser criada somente quando o Odin tiver sido licenciado corretamente.

Use uma definição de postagem para vários grupos de classes de postagem.

### Complete as seguintes etapas para criar uma definição de postagem:

1. Selecione **Novo -> Definição de postagem** do menu de contexto do Navigator.
2. Insira o nome da definição de postagem na caixa de diálogo seguinte.
3. Selecione o item **Adicionar classe de postagem nova** do menu de contexto.
4. Insira o nome da definição da classe de postagem na caixa de diálogo seguinte.
5. Edite as propriedades da definição de postagem no object inspector:

Propriedade	Definição
MaxWeight	Define o peso máximo do envelope adequado para essa classe de postagem. A especificação é mostrada em gramas.
Envelope	Define qual envelope é atribuído à definição de postagem. O peso do envelope é adicionado ao peso total.
FixedAnnex	Este documento anexo será sempre adicionado. Observação: o termo 'Annex' é usado internamente para se referir aos documentos anexos.  Se, como resultado, o peso exceder o da classe de postagem, o envelope automaticamente segue para a próxima classe de postagem.
NeutralAnnex	Este documento anexo será adicionado somente se, ao fazer isso, o peso da classe de postagem não for excedido.  Observação: o termo 'Annex' é usado internamente para se referir aos documentos anexos.

### Menu de contexto da janela de definição de postagem

#### Adicionar nova classe de postagem

Adiciona uma nova classe de postagem.

#### Remover classe de postagem

Remove a classe de postagem selecionada.

#### Renomear classe de postagem

Permite que você renomeie uma classe de postagem.

**Atribuir de envelope selecionado**

Atribui o envelope selecionado da visão geral do envelope à classe de postagem.  
Como alternativa, o envelope pode ser atribuído pelo recurso arraste e solte do Object Inspector.

**Remover atribuição de envelope**

Remove o envelope atribuído da classe de postagem.

**Atribuir documentos anexos selecionados**

Atribui o documento anexo selecionado da Visão geral do documento anexo ao grupo de postagem.

**Remover documento anexo**

Remove o documento anexo atribuído do grupo de postagem.

**Janela de visão geral do envelope**

Exibe uma lista de todos os envelopes disponíveis.

**Menu de contexto da janela de visão geral do envelope****Atribuir envelope**

Atribui o envelope selecionado ao grupo de postagem. O item de menu somente é ativado quando a propriedade Envelope da classe de postagem é selecionada.

**Janela de visão geral do documento anexo**

Exibe uma lista de todos os documentos anexos.

**Menu de contexto da janela de visão geral do documento anexo****Atribuir documento anexo**

Atribui o documento anexo selecionado ao grupo de postagem. O item de menu somente é ativado quando a propriedade FixedAnnex ou FixedEnclosure da classe de postagem é selecionada.

**Remover documento anexo**

Remove os documentos anexos atribuídos.

**Atribuir classe de postagem**

Atribui o grupo de postagem selecionado ao documento anexo. O item de menu somente é ativado quando a classe de postagem é selecionada.

**Remover classe de postagem**

Remove a classe de postagem atribuída.

## Formulários

Cria um formulário para cada tipo de papel que você precisar usar.

**Para criar um formulário, execute estas etapas:**

1. Selecione **Novo -> Formulário** do menu de contexto do Navigator.
2. Digite o nome do formulário na caixa de diálogo seguinte.

3. No campo **Peso em onças**, insira o peso do formulário.  
A especificação de peso é decisiva no cálculo do peso do envelope. Portanto, é crucial que ela seja a mais exata possível.
  4. No campo **Descrição**, insira uma descrição de texto livre opcional.
- Os formulários disponíveis são exibidos no painel inferior da janela do editor.

## Impressoras

### Impressora terceirizada

Impressoras terceirizadas podem ser criadas somente quando o Odin tiver sido licenciado corretamente.

Impressoras terceirizadas são usadas para imprimir um fluxo para um driver de impressora física por meio do componente **OdinStreamPrint** (Impressora terceirizada) e também para definir nomes de bandeja 'virtuais' necessários para converter fluxos com o componente **OdinConvert**.

#### Conclua as etapas seguintes para criar uma impressora terceirizada:

1. Selecione **Novo -> Impressora terceirizada** no menu de contexto do Navigator.
2. Insira o nome da impressora terceirizada na caixa de diálogo seguinte.
3. Use o botão **Seleção de impressora** na guia **Odin Editor Tools** para selecionar um driver de impressora.
4. Crie as bandejas obrigatórias usando o botão **Bandeja** na guia **Odin Editor Tools**.
5. Atribua os formulários obrigatórios usando a propriedade **Formulário(s)** no Object Inspector.

### Bandeja de impressora terceirizada

Durante o fluxo, os nomes de bandejas 'virtuais' no arquivo de fluxo são definidos no nível de documento.

Esses são mapeados como bandejas reais da respectiva impressora de formato / destino de saída no Perfil (Compart-) correspondente, quando o fluxo é convertido.

### Menu de contexto da janela da Impressora

#### Adicionar nova bandeja

Adiciona uma nova bandeja.

#### Excluir bandeja

Remove a bandeja selecionada.

#### Renomear bandeja

Permite que você renomeie a bandeja selecionada.

#### Atribuir formulário selecionado

Atribui o formulário escolhido da bandeja selecionada.

#### Remover atribuições de formulário

Remove os formulários atribuídos da bandeja.

**Selecionar driver de impressora**

Permite que você selecione um driver de impressora.

**Visão geral de formulários**

Exibe uma lista de todos os formulários disponíveis.

**Menu de contexto da janela Visão geral dos formulários****Atribuir formulário**

Atribui o formulário escolhido da bandeja selecionada.

**Remover atribuições de formulário**

Remove os formulários atribuídos da bandeja.

**Atualizar**

Atualiza a visão geral dos formulários.

**Impressora de email**

Uma impressora de email é usada para enviar mensagens de email por SMTP.

Os documentos são convertidos de acordo com as respectivas propriedades de bandeja (Email-Subject, Email-Body ou Attachment/AttachmentCollect).

**Para criar uma impressora de email, execute estas etapas:**

1. Selecione **Novo -> Impressora de email** do menu de contexto do Navigator.
2. Insira o nome da impressora de email.
3. Crie as bandejas obrigatórias usando o botão **Bandeja** na guia **Odin Editor Tools**.
4. Atribua os formulários obrigatórios usando a propriedade **Formulário(s)** no Object Inspector.
5. Use a propriedade **Outputformat** para selecionar o formato necessário.

**Bandejas da impressora de email**

Elas podem ser usadas para definir como um documento deve ser usado/convertido.

Se um anexo for criado, seu nome é inserido na tabela de trabalho Odin, no campo **ODJ\_EMAIL\_ATTACHMENT** .

**Menu de contexto da janela da Impressora****Adicionar nova bandeja**

Adiciona uma nova bandeja.

**Excluir bandeja**

Remove a bandeja selecionada.

**Renomear bandeja**

Permite que você renomeie a bandeja selecionada.

**Atribuir formulário selecionado**

Atribui o formulário escolhido da bandeja selecionada.

#### **Remover atribuições de formulário**

Remove os formulários atribuídos da bandeja.

#### **Visão geral de formulários**

Exibe uma lista de todos os formulários disponíveis.

#### **Menu de contexto da janela Visão geral dos formulários**

##### **Atribuir formulário**

Atribui o formulário escolhido da bandeja selecionada.

##### **Remover atribuições de formulário**

Remove os formulários atribuídos da bandeja.

##### **Atualizar**

Atualiza a visão geral dos formulários.

## Impressora online

Impressoras online são usadas para imprimir documentos de um driver de impressora física (impressão online / impressão direta).

#### **Conclua as seguintes etapas para criar uma impressora online:**

1. Selecione **Novo -> Impressora Online** no menu de contexto do Navigator.
2. Insira o nome da impressora online na caixa de diálogo seguinte.
3. Use o botão **Seleção de impressora** na guia chamada **Odin Editor Tools**.
4. Crie as bandejas obrigatórias usando o botão **Bandeja** na guia **Odin Editor Tools**.
5. Atribua os formulários obrigatórios usando a propriedade **Formulário(s)** no Object Inspector.  
Os formulários que são selecionados para os documentos no Modus e armazenados nos dados de trabalho do Odin são usados para mapear um documento para uma bandeja especificada.

#### **Menu de contexto da janela da Impressora**

##### **Adicionar nova bandeja**

Adiciona uma nova bandeja.

##### **Excluir bandeja**

Remove a bandeja selecionada.

##### **Renomear bandeja**

Permite que você renomeie a bandeja selecionada.

##### **Atribuir formulário selecionado**

Atribui o formulário escolhido da bandeja selecionada.

##### **Remover atribuições de formulário**

Remove os formulários atribuídos da bandeja.

**Selecionar driver de impressora**

Permite que você selecione um driver de impressora.

**Visão geral de formulários**

Exibe uma lista de todos os formulários disponíveis.

**Menu de contexto da janela Visão geral dos formulários****Atribuir formulário**

Atribui o formulário escolhido da bandeja selecionada.

**Remover atribuições de formulário**

Remove os formulários atribuídos da bandeja.

**Atualizar**

Atualiza a visão geral dos formulários.

**Propriedades da impressora****Geral****Ação no caso de um erro**

Ação tomada no caso de um erro

Especifica, no caso de um erro, se a impressão em todas as impressoras na pilha está desativada, ou somente a impressora em que o erro ocorreu.

Valores possíveis	Significado
OdinPrintErrorAction.StopStackPrint	Todas a impressoras serão desativadas.
OdinPrintErrorAction.StopStackPrintForPrinter	A impressora que resultaram em um erro serão desativadas.

Essa propriedade fica disponível para impressoras online e terceirizadas.

**Descrição**

Texto livre to opcional para adicionar uma descrição da impressora.

**Impressora**

Nome físico da impressora. A lista suspensa mostra todas as impressoras disponíveis para o sistema.

Essa propriedade fica disponível para impressoras online e terceirizadas.

**Opções de impressora**

String separada por vírgula com opções.

A opção **Duplex\_AddPage=true/false** é suportada no momento.



A opção especifica se uma página vazia deve ser adicionada ao documento duplex com um número ímpar de páginas, (para impressão online, somente quando UseOneSpoolFile=true).

Por padrão (ou whenDuplex\_AddPage=false) a última página é definida como modo simplex.

Essa propriedade fica disponível para impressoras online e terceirizadas.

### **Prioridade**

Prioridade de uma impressora.

### **Visível para seleção**

Valores possíveis	Significado
True	A impressora está visível e pode ser selecionada por um usuário.
False	A impressora não está visível e não pode ser selecionada por um usuário.

### **Conversão**

Estas propriedades estão disponíveis para impressoras de email.

### **LicenseFile**

Caminho e nome do arquivo de licença para o Compart .NET-Toolkit.

### **ProfilePath**

Caminho para os perfis Compart definidos nos filtros de entrada e saída respectivos.

### **TraceFile**

Caminho e nome do arquivo de trace para o Compart .NET-Toolkit.

### **TraceLevel**

Nível de trace.

### **Eventos de impressora**

#### **OnAfterPrint**

O script especificado é executado após a impressão do documento.

Este evento fica disponível para impressoras online e terceirizadas.

#### **Contexto disponível:**

#### **OdinPrint**

#### **OnBeforePrint**

O script especificado é executado antes da impressão do documento.

Este evento fica disponível para impressoras online e terceirizadas.

**Contexto disponível:**

**OdinPrint**

**OnConvert**

O script especificado é executado durante a conversão do documento para um anexo.

Pode ser usado para definir o nome de um arquivo anexado.

Este evento fica disponível para impressoras de e-mail.

**Contexto disponível:**

**OdinEmailConvert**

**OnEmail**

Este script será executado durante o envio de um e-mail.

As configurações do servidor SMTP são armazenadas na impressora de e-mail.

Configurações de e-mail como o remetente ou o assunto são definidos no script.

Este evento fica disponível para impressoras de e-mail.

**Contexto disponível:**

**OdinEmail**

**OnError**

O script especificado é executado no caso de um erro.

**Contexto disponível:**

**OnError**

**OnPrint**

Se um script é especificado, ele é responsável pela impressão e é chamado para cada documento.

Este evento fica disponível para impressoras online e terceirizadas.

**Configurações de SMTP**

Estas propriedades estão disponíveis para impressoras de email.

**SmtppickupDirectory**

Diretório no qual os emails são criados, se **SmtpdeliveryMethod** estiver definido como **SpecifiedPickupDirectory**.

**Smtpp Username**

Nome de usuário para a autenticação do servidor SMTP.

**SmtpdeliveryMethod**

Especifica como o email será enviado.

**Valores possíveis:**

Valor	Significado
OdinSmtpDeliveryMethod.Network	Os emails são enviados por meio do servidor SMTP configurado (SmptHost / Smtpport).
OdinSmtpDeliveryMethod.SpecifiedPickupDirectory	Os emails são copiados para o diretório especificado na propriedade SmtppickupDirectory e devem ser enviados por um aplicativo externo.
OdinSmtpDeliveryMethod.PickupDirectoryFromIIS	Os emails são copiados para o diretório configurado no Internet Information Server (IIS).

**Smtp-Domain**

Domínio/computador no qual as informações de login serão verificadas.

**Smtp-Host**

Host no qual o servidor SMTP está em execução.

**Smtp-Password**

Nome de usuário para a autenticação do servidor SMTP.

A senha deve ser codificada.

Para codificar a senha, use a ferramenta **Encoder.exe** no diretório bin.

**Smtp-Port**

Porta do servidor SMTP.

**UseDefaultCredentials**

Valor	Significado
True	Autenticar com SMTP-Server usando o usuário conectado.
False	Autenticar com SMTP-Server usando SmptUserName / Smtppassword.

**Propriedades de bandeja****BinType**

Define como o documento Odin é usado e convertido.

**Valores possíveis:**

Valor	Significado
OdinEmailPrinterBinType.Attachment	O documento Odin é convertido para o formato de saída correspondente (por exemplo, PDF) e adicionado ao email como um anexo separado.  O nome do anexo pode ser definido no evento OnConvert.
OdinEmailPrinterBinType.AttachmentCollect	Todos os documentos Odin são convertidos em um arquivo de anexo no formato de saída correspondente, ou seja, vários documentos são combinados em um único arquivo.
OdinEmailPrinterBinType.Body	O documento Odin é convertido em formato de texto ou HTML e usado como corpo do email.
OdinEmailPrinterBinType.Subject	O documento Odin é convertido em formato de texto e usado como corpo do email.  Isso também permite que um documento Odin de linha única seja convertido para um texto de referência individual do email.

### Duplex

Especifica o modo duplex para a bandeja usada para imprimir o documento Odin.

Valores possíveis:

Valor	Significado
Simplex	OdinDuplexSetting.Simplex Sem duplex
Duplex_Horizontal	OdinDuplexSetting.Horizontal Duplex na borda menor
Duplex_Vertical	OdinDuplexSetting.Vertical Duplex na borda maior

### Forms

Especifica 1 para n formulários atribuídos à bandeja. Cada formulário pode ser atribuído apenas a uma bandeja da impressora.

A lista suspensa exibe todos os formulários disponíveis para o sistema.

### NextInputBin

Especifica o nome da bandeja (virtual) para as seguintes páginas.

### Outputformat

Formato de saída do anexo.

A lista suspensa mostra todos os formatos de saída possíveis como definidos no arquivo ModusSuite.Common.PropertyEditors.dll.config. Os valores possíveis para o filtro de saída dependem da licença do Compart .NET-Toolkit correspondente.

**PhysicalInputBin**

A bandeja física usada para impressão.

A lista suspensa exibe todas as bandejas do driver de impressora física selecionados.

## Parâmetro inicial

O ModusStudio pode ser iniciado com parâmetros específicos para controlar o comportamento inicial.

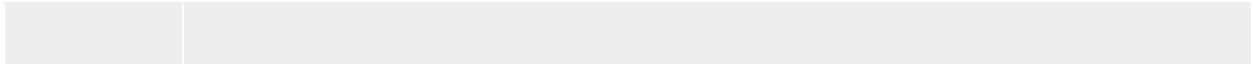
Os parâmetros iniciais têm a seguinte estrutura:

**/parametername:Paramvalue**

Se **Paramvalue** contiver um espaço, o parâmetro deverá estar entre aspas.

Os seguintes parâmetros são suportados:

Parâmetro	Significado								
startsystem	<p>OID do sistema a ser carregado automaticamente após a inicialização do aplicativo.</p> <p>Se o parâmetro não for definido, o Studio iniciará com o último sistema usado.</p> <p><b>Exemplo</b></p> <pre>/startsystem:MySystemOID</pre>								
navipath	<p>Esse elemento define qual pasta será selecionada no Navigator.</p> <p>Se nenhum sistema de início for definido, o parâmetro recorrerá ao último sistema usado.</p> <p>O nome do sistema não deve ser incluído no caminho.</p> <p><b>Exemplo</b></p> <pre>/navipath:/Test</pre>								
credentials	<table border="1"><tr><td>Input</td><td>O nome do usuário e a senha devem ser inseridos.</td></tr><tr><td>Silent</td><td>O nome do usuário e a senha devem ser inseridos uma vez. Se a opção de login "Salvar senha" estiver ativada, os dados inseridos serão usados para logins futuros.</td></tr><tr><td>SilentUI</td><td>Este parâmetro será necessário se for preciso redefinir uma senha <b>Silenciosa</b>. O diálogo do login é exibido novamente.</td></tr><tr><td>Windows</td><td>O login usa o nome do usuário e a senha do Windows. Esta configuração também é usada quando as <b>credenciais</b> do parâmetro não são especificadas.</td></tr></table>	Input	O nome do usuário e a senha devem ser inseridos.	Silent	O nome do usuário e a senha devem ser inseridos uma vez. Se a opção de login "Salvar senha" estiver ativada, os dados inseridos serão usados para logins futuros.	SilentUI	Este parâmetro será necessário se for preciso redefinir uma senha <b>Silenciosa</b> . O diálogo do login é exibido novamente.	Windows	O login usa o nome do usuário e a senha do Windows. Esta configuração também é usada quando as <b>credenciais</b> do parâmetro não são especificadas.
	Input	O nome do usuário e a senha devem ser inseridos.							
	Silent	O nome do usuário e a senha devem ser inseridos uma vez. Se a opção de login "Salvar senha" estiver ativada, os dados inseridos serão usados para logins futuros.							
	SilentUI	Este parâmetro será necessário se for preciso redefinir uma senha <b>Silenciosa</b> . O diálogo do login é exibido novamente.							
Windows	O login usa o nome do usuário e a senha do Windows. Esta configuração também é usada quando as <b>credenciais</b> do parâmetro não são especificadas.								
<p><b>Exemplo</b></p> <pre>/credentials:Silent</pre>									



# Controle de versão

## Informações gerais

O controle de versão é um componente opcional do ModusOne e está disponível apenas quando seu uso foi licenciado.

Quase todos os tipos de objeto administrados no ModusOne Studio podem ter versão.

### Os tipos de objeto a seguir não podem ter versão:

- Pastas
- Perfis
- Funções
- Mapeamentos de função
- Aliases de DB
- Trabalhos de transferência
- Sistemas

### Novos objetos

Um novo objeto é sempre criado como uma versão de trabalho com o número 0.1.

### Migração

Ao migrar do modus 4 para o ModusOne, todos os objetos migrados são criados como releases com o número 1.0.

### Repositórios som controle de versão

Se um repositório, editado como o ModusOne Versão 5.0.4, for transferido para um repositório ModusOne Versão 5.1., todos os objetos transferidos recebem o número de versão 1.0.

Se o controle de versão não estiver ativo para o ModusOne Studio, todos os objetos recebem o número de versão 1.0.

### Renomeando objetos com versão

Renomear um objeto não afeta o seu número de versão. Todas as versões e referências usam o novo nome do objeto.

## Tipos de versão

É feita uma distinção entre dois tipos de versões de objetos:

Tipo de versão	Exemplo
Releases	1.0, 2.0, 3.0



Versões de trabalho	0.1, 1.1, 1.2, 2.1, 3.1, 3.2
---------------------	------------------------------

Apenas a versão mais alta do objeto é a versão ativa, não importa se é uma versão de trabalho ou um release. Somente essa versão é usada pelos objetos.

O número de versão aumenta automaticamente, não é possível atribuir manualmente números de versão.

## Criar uma nova versão

É possível criar novas versões apenas para objetos editáveis.

### Releases

**Não** é possível criar diretamente um release com o número maior mais próximo.

Para salvar as alterações de um release, você deve criar primeiro uma versão de trabalho.

**Execute as seguintes etapas para criar uma nova versão de trabalho a partir de um release:**

1. Abra o objeto.
2. Faça as alterações necessárias.
3. Selecione a função **Salvar** ou a função **Criar nova versão** no menu principal.
4. Insira uma descrição opcional na caixa de diálogo seguinte.
5. Confirme a caixa de diálogo com OK para criar uma nova versão de trabalho.

### Exemplo:

Você editou a versão **2.0** de um objeto. Depois de salvar, o objeto terá a versão número **2.1**.

### Versões de trabalho

Se uma versão de trabalho de um objeto for editada e salva com o botão **Salvar** comum, o número de versão permanece o mesmo.

**Execute as seguintes etapas para criar uma versão de trabalho sem criar uma nova versão:**

1. Abra o objeto.
2. Faça as alterações necessárias.
3. Selecione a função **Salvar** no menu principal.

### Exemplo

A versão de trabalho **2.1** de um objeto é editada.

- Se as alterações forem salvas com o botão **Salvar**, uma nova versão será criada. O número da versão permanecerá **2.1**. O último status do objeto será substituído.

**Execute as seguintes etapas para criar uma nova versão a partir de uma versão de trabalho:**

1. Abra o objeto.

2. Faça as alterações necessárias.
3. Selecione a função **Criar nova versão** ou no menu principal.
4. Insira uma descrição opcional na caixa de diálogo seguinte.
5. Na caixa de diálogo, selecione o tipo de versão **Versão de trabalho** ou **Release**.  
A opção Release só pode ser selecionada se o usuário tiver a **Criar release** correta e as alterações no objeto forem salvas de maneira válida.  
Em ambos os casos, o número de versão é automaticamente aumentado.
6. Confirme a caixa de diálogo com **OK**.

### Exemplo

A versão de trabalho **2.1** de um objeto é editada.

Se Versão de trabalho for selecionada, a versão **2.2** será criada.

Essa versão **não pode** ser transferida para outro sistema.

Se Release for selecionada, a versão **3.0** será criada.

Essa versão pode ser transferida para outro sistema.

### Consulte também:

[Versões de trabalho](#)

## Alterações a objetos com versão

A criação de uma nova versão de um objeto com versão controlada não terá qualquer efeito no número de versão de objetos pai ou filho.

### Exemplo

O bloco de texto **Account\_Overview** está no modelo de documento **Partner\_Application** e contém o bloco de texto filho **Partner\_Accounts**.

O bloco de texto **Partner\_Accounts** contém o bloco de texto filho **Partner\_Account\_InterestRate**.

Modelo de documento **Partner\_Application** Bloco de texto **Account\_Overview**

Bloco de texto **Partner\_Accounts**

Bloco de texto **Partner\_Account\_InterestRate**


As modificações no bloco de texto **Partner\_Accounts** resultam apenas em uma nova versão desse bloco de texto.

Nem o bloco de texto filho **Partner\_Account\_InterestRate** nem o bloco de texto filho **Account\_Overview** recebem um novo número de versão.

O modo de documento **Partner\_Application** também não recebe um novo número de versão.



## Mostrar histórico de objeto

Execute as etapas a seguir para exibir o histórico de um objeto:

1. Selecione o objeto no Navigator.
2. Selecione a função **Mostrar histórico de objeto** no menu de contexto do Navigator.  
As janelas seguintes exibem o histórico completo do objeto.
3. Selecione uma entrada na lista e clique no botão **Mostrar objeto** .
  - A versão mais recente do objeto será aberta para edição, ou seja, o objeto com o maior número de versão.
  - As versões mais antigas só podem ser abertas no modo de exibição.  
Qualquer alteração em versões anteriores não podem ser salvas até que a versão **tenha sido** reativada.
  - As versões mais antigas de modelos de documento, pacotes e processos não podem ser geradas ou executadas, nessa ordem

## Reativar uma versão mais antiga

**Execute as etapas a seguir para reativar uma versão mais antiga de um objeto:**

1. Selecione o objeto no Navigator.
2. Selecione a função **Mostrar histórico de objeto** no menu de contexto do Navigator.  
As janelas seguintes exibem o histórico completo do objeto.
3. Selecione uma entrada na lista e clique no botão **Mostrar objeto** .
4. Clique no botão **Reativar versão**  na **Barra de Ferramentas de Acesso Rápido**.  
Essa função cria uma cópia da versão aberta como uma nova versão de trabalho.

### Exemplo

A versão mais recente de um objeto é a **2.2**.

Se a versão **1.1** do objeto for aberta e reativada, uma nova versão de trabalho, com o número **2.3**, é criada como uma cópia da versão **1.1**.

A reativação de uma versão de objeto só está disponível para objetos editáveis.

Objetos protegidos contra gravação, como é normal em sistemas de produção, não podem ser reativados.

## Transferir objetos com versão para outros sistemas

Para obter mais informações, consulte o capítulo sobre

[Trabalhos de transferência](#)

## Atributos de usuário StudioStartSystemOld e StudioStartFolder

Ao inserir um ID de objeto do sistema no atributo do usuário **StudioStartSystemOld**, você pode definir um sistema de início específico. Isso funciona de maneira similar para configurar o sistema de início nos parâmetros iniciais.

Ao inserir um ID de objeto do sistema no atributo do usuário **StudioStartFolder**, você pode definir um sistema de início específico.

Diferente do parâmetro navipath, o valor aqui não é o caminho para a pasta, mas o próprio nome da pasta. Portanto, esse atributo, apenas tem efeito na configuração de uma pasta inicial quando a permissão MwsChangeFolder está ausente.

Os valores definidos nos parâmetros iniciais têm prioridade sobre aqueles especificados nos atributos do usuário.

- Se os parâmetros iniciais especificarem um sistema de início, esse sistema será predefinido na inicialização do aplicativo.
- Se o parâmetro inicial não for definido, o aplicativo verificará se o atributo do usuário **StudioStartSystemOld** está definido e, em seguida, aplicará o valor na inicialização do aplicativo.
- Se nenhuma das opções for definida, o último sistema usado será definido na inicialização do aplicativo.
- Se o parâmetro inicial navipath for definido, as seguintes verificações serão feitas:
  - Se o usuário tiver o direito **Pasta de alterações (7)**, o Navigator alternará para a pasta. Se a pasta não for localizada, o Navigator alternará para a pasta raiz do sistema.
  - Se o usuário não tiver o direito **Pasta de alterações (7)**, a pasta raiz do sistema se tornará a pasta padrão para o usuário. Se a pasta especificada não for encontrada no sistema, um aviso será exibido e os dados do Navigator serão apagados. Os usuários poderão alterar para outro sistema se eles tiverem o direito Alterar sistema.

Mas isso só será possível se o sistema de destino tiver uma pasta com o mesmo nome.

- Se o parâmetro inicial navipath não for definido, as seguintes verificações serão feitas:
  - Se o atributo **StudioStartFolder** for definido e o usuário tiver o direito **Pasta de alterações (7)**, nenhuma ação adicional será executada.
  - Se o usuário não tiver o direito **Pasta de alterações (7)**, a pasta raiz do sistema se tornará a pasta padrão para o usuário. Se a pasta especificada não for encontrada no sistema, um aviso será exibido e os dados do Navigator serão apagados. Os usuários poderão alterar para outro sistema se eles tiverem o direito Alterar sistema. Mas isso só será possível se o sistema de destino tiver uma pasta com o mesmo nome.

## Índice

### 1

1-n arquivos .....525

### A

Abrir.....15, 230, 251

Acesso ao banco de dados .....166

Ações .....67, 77

Acrescentar .....97, 213

AddAttachment .....322, 485

AddBatchPrintJob.....531

AddBodyHtmlImage.....322, 485

AddBookmark .....334, 376, 578

AddComment.....377

AddDataRecord .....169

AddDocument .....497

AddDocumentByName .....403

AddDocumentById .....406

AddDocumentCopyByName .....408

AddDocumentCopyById .....410

AddFile .....497

AddPJLCommand.....377

AddToHeader .....323, 486

Administração .....634

Administração de alias de BD.....19

Alias de BD.....19, 20

Alterações .....650

Analisar.....121

Anterior.....124, 174, 355

Aplicar bloco de texto .....51

AppendDocument .....367

AppendVJob .....348

ApplyDataRecord.....169

ApplyDataSet .....169

ArchiveJob .....454

Áreas.....434

Arquivo XML .....247

Arredondar .....130

AssignData.....285, 444

Assistente de consulta .....19

Ativar objetos de dados.....624

Atribuir .....50, 64, 76, 88, 615

Scripts .....88

Atribuir scripts de leitura.....615

Atributos .....628, 631

Atributos de arquivamento idênticos .....80

Atributos de usuário de StudioStartSystemOld

.....651

Atualizar .....6, 165, 619, 625

Atualizar objetos de dados.....619, 625

Aumentar .....126

Ausente .....628

Objetos de dados.....628

### B

Barra de ferramentas de acesso rápido .....3

BatchJobCount.....531

Beep.....98

BigInteger .....98

Bloco de processo.....434

Bloco de texto .....28, 58, 64

Bloco de texto filho.....50

Bof .....99

### C

Caixa de ferramentas.....9

Caixa de texto.....426, 427

CalcEnvelopePages.....348

CalcEnvelopePapers .....349

CalcEnvelopePostage.....	349	Contagem .....	103, 350
CalcEnvelopeWeight.....	349	Contains_Exactly .....	103
Calendário pop-up .....	428	Contém .....	102
CallSystem .....	99	Conteúdo de variável especial .....	34
Campo de entrada numérico .....	429	Context MwsProcess .....	287
Cancelar .....	598	Contexto ModusBatch .....	285
Caso especial.....	80	Contexto OdinArchive .....	292
<b>Ch</b>		Contexto OdinConvert.....	296
ChangeExtension .....	182	Contexto OdinCustomProcess.....	303
CheckEvery .....	93	Contexto OdinDbTask .....	307
CheckFirst .....	94	Contexto OdinEmail.....	312
<b>C</b>		Contexto OdinEmailConvert.....	328
ClearAttachments .....	323, 486	Contexto OdinPrint .....	361
ClearBodyHtmlImages .....	324, 487	Contexto OdinRendition .....	365
ClearCache .....	256	Contexto OdinStream .....	367
ClearCertificates.....	324, 487	Contexto OdinStreamPrint .....	385
CloseDatabase.....	101	Contexto OnError.....	389
CloseSQL .....	101	Contexto OnNavigateObject .....	389
Colunas de resultado .....	610	Contexto OnProcessError .....	390
Combine .....	182	Contexto Process .....	393
Comentários.....	87, 442	Contexto ProcessTimer .....	395
Conclusão do código.....	85	Contexto SelectionQueryScript.....	396
Condição da variável .....	45	Contexto TextComponent .....	397
Condição If.....	94	Contexto TextDocument .....	400
Condição IfElse .....	95	Contexto TextDocumentCollection .....	402
Condições .....	58, 70, 83, 88, 89, 440, 610	Contexto TextDocumentReference.....	413
Conectar blocos de processo.....	435	Contexto TextVariable .....	417
Configuração.....	284	Controle de conteúdo.....	49, 64
Configurações de conexão.....	19	Controle de conteúdo EmptyLine.....	49
Configurações SMTP.....	642	Controle remoto.....	20
Confirmar .....	102, 168, 350	Conversão.....	641
Conjunto de caracteres.....	9	ConvertDocToDocX.....	527
Conjuntos de teste.....	626	ConvertDocument.....	467
Consulta SQL.....	614	ConvertSingleFile.....	504

ConvertXDocToHtml .....	509	DefinePostageDefinitionByOld .....	341
ConvertXDocToString .....	510	DefinePrinterByName .....	341, 415, 587
ConvertXDocToText.....	510	DefinePrinterByOld .....	342, 416, 588
ConvertXDocToXps .....	518	DefinePrinterForBundleByName .....	445
Cópias de documento .....	76	Definição .....	25, 35
CopyFile .....	183	Regras de variáveis.....	35
CopyLabel.....	64	Definição de consulta .....	609
CopySection .....	249	Definições de objetos .....	58, 73
CreateDirectory .....	184	Definições de postagem.....	635
CreateDocuments .....	444	Definir bookmarks.....	86
CreateEnv.....	350	DeleteDirectory .....	184, 286
CreatePdfSigner.....	497	DeleteFile .....	184
CreatePool .....	103	DeleteFiles .....	185
CreateRenditionForProcess.....	549	DeleteFinishedProcesses .....	452
CreateRenditionForSelectedProcess .....	549	DeleteKey.....	249
CreateTextFile.....	183	DeletePool .....	105
Criar.. 19, 23, 26, 27, 28, 60, 62, 64, 75, 83, 222, 424, 431, 609, 612, 613, 614, 617, 621, 628, 649		DeleteProcessRecords .....	482
Criar consultas SQL adicionais.....	614	DeleteStack.....	481
Criar definições de consulta .....	609	Depurar .....	626
Criar objetos de dados .....	612	Desinstalação .....	608
Criar objetos de dados vazios .....	617	DirectoryExists.....	185
Criar scripts de leitura.....	613	Dividido .....	356
Criar seleções XSD .....	628	Doc_GetFile_Mime .....	257
CurrentDate.....	177	Doc_SetFile_Mime.....	257
CurrentDateToString .....	177	Documento dinâmico .....	60
CurrentPrinter.....	532	Documento estático .....	62
<b>D</b>		Documentos .....	65, 71, 76, 80
DateStringToInt64 .....	178	Documentos anexos.....	634
DayOfWeek .....	178	Documentos de teste .....	68
DaysInMonth.....	179	Documentos TIFF estáticos.....	63
Decimal .....	104	DoWork .....	451
DefineContainer .....	170	<b>E</b>	
DefinePostageDefinitionByName.....	341	Editar.....	24, 613, 617, 625
		Editar conexões de banco de dados .....	24

Editar interface de seleção .....	625	Exemplos .....	240, 269, 390
Editar objetos de dados .....	613	Exibições .....	10
EditPage .....	334, 378	Exportar .....	20, 425
Encaminhar .....	76	Exportar objetos .....	425
EndDocConvert .....	527	<b>F</b>	
EndTimer .....	351	Fechar .....	100, 221, 248
EnumEnvelopes .....	342	Ferramentas do designer de processo .....	431
Envelopes .....	634	Ferramentas do editor .....	61, 63, 87
Envelopes abertos .....	15	Ferramentas do editor de Xdata .....	617
Enviar .....	326, 491	FileAppendText .....	186
Eof .....	106, 170, 229, 351	FileExists .....	194, 598
EraseSection .....	250	FileExistsEx .....	599
Esquema .....	628	FileName .....	240
Eventos da variável .....	45	FileReadAllText .....	195
Eventos de bloco de texto .....	58	FileSize .....	203
Eventos de documento .....	71	Filtro simples .....	346
Eventos de impressora .....	641	Filtro simples de envelopagem .....	357
Eventos de OdinArchive .....	454	Filtro simples incluindo .....	346
Eventos de OdinConvert .....	466	FindValue .....	107
Eventos de OdinCustomProcess .....	472	FinishWork .....	451
Eventos de OdinPrint .....	530	Flush .....	230
Eventos de OdinRendition .....	548	Flutuante .....	110
Eventos de OdinStream .....	574	FormatDateTimeString .....	179
Eventos de OdinStreamPrint .....	587	Formato .....	110
Eventos de pacote .....	81	FormatStr .....	112
Excluir .....	8, 19, 24, 105, 617	Formulários .....	636
Excluir conexões de banco de dados .....	24	FreeObject .....	112
Excluir conjuntos de dados .....	617	Função .....	26, 27
Execução de script .....	442	Funções da ferramenta do editor .....	61, 63
Executar .....	106, 528, 618, 625	<b>G</b>	
ExecuteBatchPrint .....	533	Gen_Stack .....	343
ExecuteQuery .....	171	Geração de pilha .....	346, 357
ExecuteSynchron .....	186	Geral .....	38, 54, 69, 81, 93, 167, 176, 243, 256, 274, 285, 287, 292, 296, 303, 307, 311, 341, 348, 364, 367, 374, 393, 394, 397, 399, 402, 413, 417, 424, 443, 450, 452, 453, 461, 471, 480,
Exemplo de script .....	166		



483, 496, 528, 529, 543, 546, 550, 585, 593, 596, 601, 640, 648	
GetActivityStatus .....	289
GetAttachments .....	324, 487
GetAttribute.....	243, 275
GetAttributeOfNode .....	243
GetBlobField .....	113
GetCommandLine.....	203
GetCommandLineArgByName .....	203
GetCommandLineArgs .....	204
GetConnection .....	113
GetCurrentDirectory.....	204
GetCurrentStackRow ...	455, 467, 473, 533, 580, 589
GetCurrentUser .....	274
GetData .....	286, 594, 601
GetDataExtended.....	594, 602
GetDataReferenceEnumerator.....	446
GetDeliveryNotificationOptions.....	488
GetDirectories .....	204
GetDirectoryName .....	205
GetDirectoryRoot .....	205
GetDocumentEnumerator .....	469, 590
GetEnvelopeEnumerator.....	456, 475, 581
GetEnvironmentVariable .....	205
GetExtension.....	206
GetField.....	114, 171, 352
GetFieldNameEnumerator .....	365
GetFileName .....	206
GetFileNameWithoutExtension.....	207
GetFiles .....	207
GetFileSystemEntries.....	208
GetForwardInformation.....	290
GetFullPath .....	208
GetInputBin .....	335, 378
GetJobAttribute .....	599
GetJobEnumerator .....	457, 476, 582
GetList.....	258
GetLogicalDrives .....	209
GetLookupValue .....	114
GetMailEnumerator.....	536
GetMailJobEnumerator.....	536
GetMailPriority .....	488
GetManualVariables.....	446
GetNextStack .....	469, 475, 534, 583, 591
GetObject.....	115
GetObjectSelectionParameters .....	291
GetOptions.....	291
GetParam.....	344
GetParentDirectory .....	209
GetPathRoot .....	209
GetPrinterEnumerator .....	534
GetPrintJobEnumerator.....	535
GetProperty .....	368
GetRandomFileName .....	210
GetSectionNames.....	250
GetSelectionParameters.....	447
GetSmtpDeliveryMethod .....	489
GetSystemOld .....	116
GetTempFileName.....	210
GetTempPath.....	211
GetValue .....	365
GetVariable .....	115
GetXpathValue .....	244
GotoJobTillWeight .....	352
GotoRecNum .....	353
Gravar .....	238
Grupos .....	67, 77
Guia Visualização de Odin.....	16

## H

HasData	295, 303, 307, 311, 364, 366, 374, 388
HasEnvelopes	458, 477
HasExtension	211
HasObjectPermission	280
HasPermission	275
Herdar atributos do bloco de texto	53
Histórico de transferência	7

## I

IF-Check	441
IF-ELSE-Check	441
Imagem	335, 379
Importação de repositório	22
Importar objetos	6
ImportProcessFile	544
Impressora	637, 641
Terceirização	637
Impressora de email	638
Impressora online	639
IncDate	180
Index	116
Informações de contexto	86
Informações gerais	167, 424, 443, 450, 452, 453, 461, 471, 480, 483, 496, 528, 529, 543, 546, 550, 585, 593, 596, 601, 648
Iniciar	458, 470, 478, 520, 539, 583, 592, 646
Init	353
InitDataBase	117
InitDataSet	117
InitializeBatchPrint	537
Inserir	29, 50, 64, 118, 615, 624
Inserir objetos de dados	624
Inserir objetos filhos	615
Inserir variáveis	615
Inserir variáveis de bloco de texto	29

InsertVJob	354
Int64ToDateString	180
Inteiro	118
Interface	375
Interface de usuário do Word	430
Introdução	23, 28, 29, 60, 67, 75, 77, 83, 95, 426, 431, 609, 612, 634
IOdinStreamDocument	375
IsDuplex	336, 379
IsEmailPrinter	537
IsInformationRequested	291
IsInRole	283
IsLeapYear	181
IsPathRooted	211
IsPortrait	336, 380
IsPrinterActive	538
<b>J</b>	
Janela de saída	9
Janela do editor	1, 60
Janela do navegador	3
<b>L</b>	
LastBase64DocumentString	268
LastProcessId	268
LastXmlResult	269
Ler	126
Linha	336, 380
LoadBase64StringFromBinFile	267
LoadBodyFromFile	324, 489
LoadByAlias	447
LoadById	447
LoadByName	448
LoadCertificateFromSignedFile	325, 490
LoadCertificateFromCertFile	325, 489
LoadDocument	371, 519
LoadEnvelopes	372

LoadExternalDocument .....	398, 400	NewLineCRLF .....	213
LoadJobs .....	372, 373	NewLineLF .....	213
LoadSelection .....	595	NextJob .....	599
LoadSelectionByName .....	595	Nós de resumo .....	616
LoadSelectionByOld .....	595	Nova conexão de banco de dados .....	23
Localizar .....	107	Nova versão .....	649
Loop .....	442	Novo alias de BD .....	19
Loops While .....	95	Novo sistema .....	23
<b>M</b>		NumberOfFilesConverted .....	520
Mapeamento de função .....	27	<b>O</b>	
Máscara de entrada .....	36, 426	Obj_GetStructure .....	258
Máx .....	119	Obj_SetStructure .....	259
Mensagem .....	172, 173	Obj_Toggle .....	260
Menu de contexto .....	3, 60	Object Inspector .....	8
Menu principal .....	1	Objetos bloqueados .....	7
Métodos .....	345	Objetos com versão .....	650
Métodos de GetParam .....	345	Objetos de dados .....	612, 618, 619, 620, 628
Migração .....	91	Ausente .....	628
Scripts .....	91	Objetos de variável .....	52, 65
Mín .....	120	Objetos excluídos .....	8
Modelo de dados .....	50, 58, 64, 71, 76	OdinDbTasks .....	480
Modificar .....	245, 614	Opções .....	427
Modificar atributos .....	245	OpenSQL .....	121
Modus .....	91	Ordem do script .....	419
Mostrar histórico de objeto .....	650	Outros sistemas .....	651
Mostrar lista de filtros .....	6	<b>P</b>	
MoveDirectory .....	212	Pacote .....	75, 76, 80, 81
MoveFile .....	212	Pacotes de teste .....	78
Mover .....	120	Pacotes dinâmicos .....	78
MwsActivityStatus .....	453	PageCount .....	337, 381
MwsUrl .....	269	Página anterior .....	5
<b>N</b>		Parâmetro inicial .....	646
Navegar .....	399, 401, 411, 418	Parâmetros do script .....	86
NewLine .....	213, 240	Perfil .....	26

Pesquisar .....	45, 130	Propriedades de DataSelectionManager .....	593
Pilha.....	11, 346, 357	Propriedades de DocumentCollectionManager .....	444
Ping .....	260, 603	Propriedades de ModusBatchManager .....	450
Pools de variável.....	52	Propriedades de MwsAssistantManager .....	452
Pós .....	338, 381	Propriedades de objeto de dados.....	619
Posição .....	435	Propriedades de OdinArchive.....	453
Possíveis parâmetros.....	345	Propriedades de OdinConvert.....	461
PrepareParameter .....	173	Propriedades de OdinCustomProcess.....	472
PrepareSQL.....	123	Propriedades de OdinEmail.....	484
Primeira .....	109, 171, 351	Propriedades de OdinFileConvert .....	496
PrintAndArchive .....	448	Propriedades de OdinFilterManager .....	528
PrintDocument .....	591	Propriedades de OdinPrint .....	529
PrintJob.....	538	Propriedades de OdinProcessImport .....	543
Procedimento armazenado.....	618	Propriedades de OdinRendition .....	546
Process_Close .....	261	Propriedades de OdinStream .....	563
Process_Create.....	261	Propriedades de OdinStreamPrint .....	586
Process_Delete .....	262	Propriedades de seleção.....	633
Process_Forward .....	263	Propriedades de variável de objeto de dados .....	620
Process_GetInfo.....	263	Propriedades de XmlProcessingManager .....	597
Process_GetLastError.....	264	Propriedades de XwsDataSelectionManager	601
Process_Open.....	265	Propriedades gerais da variável .....	38
Process_SetData .....	265	Protocolo .....	125
Process_Start.....	266	Próxima página.....	5
Processar .....	10, 274, 431, 434, 477	Próximo.....	5, 121, 173, 354
ProcessData.....	449	PutField.....	126
ProcessJob .....	477	<b>R</b>	
ProcessJobs .....	478	ReadBool .....	251
Processo hospedado.....	20	ReadInteger .....	252
Processo MWS .....	433	ReadLine.....	238
Processos padrão.....	433	ReadNextEnvelope .....	355
Propriedades.. 66, 285, 287, 292, 296, 303, 307, 312, 328, 361, 365, 367, 375, 385, 389, 390, 393, 395, 396, 397, 400, 402, 413, 417, 433, 480, 563, 609, 610, 611		ReadSectionToCommatext .....	252
Propriedades da bandeja.....	643	ReadString .....	253
Propriedades da variável .....	51	ReadToEnd .....	238

Reativar .....	651	Script de exemplo OdinEmail .....	494
Recnum .....	355	Script de exemplo OdinFileConvert .....	524, 525, 526
Redefinir .....	491	Script de exemplo OdinFilterManager.....	528
Região .....	95	Script de exemplo OdinPrint .....	539
RegisterLib .....	129	Script de exemplo OdinProcessImport .....	545
Registrar .....	627	Script de exemplo OdinRendition .....	550
Regras .....	628, 631	Script de exemplo OdinStream .....	584
Regras de variáveis.....	35	Script de exemplo OdinStreamPrint.....	592
Definição.....	35	Script de exemplo XwsDataSelectionManager .....	603
RemoveAttachment .....	326, 490	Scripts.....	83, 86, 88, 91, 419, 442
RemoveFromHeader.....	326, 491	Atribuição .....	88
ReplaceFile.....	212	Migração .....	91
Reproduzir recuperação de dados .....	627	SearchAndImportProcessFile.....	545
ReRaise .....	129	Seleção .....	621, 625, 626, 633
ResetDocuments.....	449	SelFirstRecord .....	131
Restaurar sistemas .....	25	SelNextRecord.....	131
Retângulo .....	338, 381	SelRecordCount .....	131
Rollback .....	129, 168, 356	SelRecordNumber .....	132
RowCount .....	175	SelVariable .....	132
RunFilter .....	344	SelVariableEx .....	133
<b>S</b>		Serviço Windows .....	604, 608
Salvar seleção.....	628	SetAfpCopyGroup .....	382
SaveBase64StringToBinFile .....	267	SetArchiveAttribute.....	412, 417
Script de exemplo 245, 247, 274, 283, 328, 346, 357, 373, 450, 452, 459, 470, 478, 483, 494, 524, 525, 526, 528, 539, 545, 550, 584, 592, 596, 603		SetAttribute.....	244
Script de exemplo Criar .....	247, 274	SetAttributeOfNode .....	245
Script de exemplo DataSelectionManager .....	596	SetBackStackState.....	482
Script de exemplo Ler.....	245	SetCredentials .....	327, 492
Script de exemplo LoadEnvelopes.....	373	SetCurrentDirectory .....	213
Script de exemplo MWSAssistantManager ...	452	SetDataContext .....	287, 412
Script de exemplo OdinArchive .....	459	SetDeliveryNotificationOptions.....	492
Script de exemplo OdinConvert .....	470	SetExecSQL.....	134
Script de exemplo OdinCustomProcess .....	478	SetField .....	356
Script de exemplo OdinDbTasks.....	483	SetFont .....	339, 383

SetInputBin .....	339, 383	StringListIndexOf .....	145
SetMailPriority .....	493	StringListItemValues .....	146
SetManualVariables .....	449	StringListLine .....	147
SetObjectSelectionParameters .....	292	StringListLoad .....	147
SetParam .....	345	StringListParams .....	148
SetPdfSigner .....	333	StringListSave .....	148
SetPoolVariable .....	138	StringListSetCommaText .....	156
SetProperty .....	372	StringListSetValue .....	157
SetSelectionParameters .....	450	StringListSort .....	157
SetSmtpDeliveryMethod .....	493	StrLeft .....	158
SetSmtpServer .....	327, 493	StrLen .....	158
SetSQL .....	139	StrLower .....	158
SetStackError .....	469, 583, 591	StrPos .....	159
SetValue .....	366	StrReplace .....	159
Sincronização de sistema .....	23	StrReverse .....	160
Sistema de transferência .....	25	StrRight .....	161
Sistemas .....	23, 24	StrTok .....	161
Sleep .....	140	StrToLen .....	162
SplitStr .....	140	StrUpper .....	163
StartDocConvert .....	527	StrWord .....	163
StartProcedure .....	175	StrWordCount .....	164
StartProcessing .....	600	StudioStartFolder .....	651
StartTimer .....	357	<b>T</b>	
StartTransaction .....	141, 168, 357	Tabelas de alias .....	422
Status de Odin .....	17	Tabelas de pesquisa .....	423
StoredProcedure .....	141	Temporizador universal .....	443
StrDelChar .....	142	Terceirização .....	637
StreamDocument .....	584	Impressora .....	637
StringListAdd .....	143	Terminate .....	600
StringListClear .....	143	Término .....	455, 467, 473, 519, 532, 579, 588
StringListCount .....	144	Testar scripts .....	89
StringListGetCommaText .....	144	TextOut .....	340, 384
StringListGetText .....	145	Tipos de dados .....	92
StringListGetValue .....	145	Tipos de versão .....	648

Trabalho de transferência .....	424	Versão mais antiga .....	651
Trabalhos abertos .....	15	Versões de trabalho .....	7
Transferir objetos com versão .....	651	Visão geral .....	1, 17, 345
Transferir propriedades.....	54, 68, 81	Visualização de documento .....	65
Trunc .....	164	<b>W</b>	
TryFinally .....	95	WeekOfYear.....	181
TryOnError.....	96	WorkWithQuery.....	176
TryOnErrorFinally .....	96	WriteBool .....	253
<b>U</b>		WriteFormat .....	239
Última .....	6, 119, 172, 354	WriteInteger .....	254
Última visualização .....	6	WriteLine.....	239
UnRegisterLib .....	164	WriteLineFormat .....	239
<b>V</b>		WriteSectionFromCommatext .....	255
Variáveis .....	35, 45, 51, 52, 65, 611	WriteString .....	255
Variáveis globais .....	66	WriteXml.....	596
Variáveis manuais .....	38	<b>X</b>	
Vários blocos de texto .....	51	XDoc .....	526
Verificar .....	439	XPS .....	16, 526
VerifyPool.....	165	XPS Viewer .....	16
Versão .....	166, 648, 650		