

HiperTools[®] v3.0

Guia do Usuário

Brasília, DF 14 de Março de 2000
Versão 3.0.14

ATENÇÃO: ISTO É UMA LICENÇA E NÃO UM INFORMATIVO, ESTE SOFTWARE É ENTREGUE MEDIANTE A ACEITAÇÃO DOS TERMOS DESTA LICENÇA, QUE PODE E O QUE PODE SER FEITO COMO O SOFTWARE, ALÉM DAS LIMITAÇÕES REFERENTES À GARANTIA E/OU RESSARCIMENTOS

LICENÇA:

IMPORTANTE: LEIA CUIDADOSAMENTE ESSA LICENÇA ANTES DE UTILIZAR ESTE SOFTWARE. A UTILIZAÇÃO DESTE SOFTWARE IMPLICA EM SEU RECONHECIMENTO DE QUE VOCÊ ESSA LICENÇA E DE QUE CONCORDA COM SEUS TERMOS. CASO NÃO CONCORDE, DEVOLVA O SOFTWARE COMPLETO À ZEVALLOS TECNOLOGIA EM INFORMAÇÃO NO PRAZO DE 10 (DEZ) DIAS A CONTAR DE SUA AQUISIÇÃO, PARA OBTER REEMBOLSO INTEGRAL. ESTE ACORDO DE UTILIZAÇÃO É A PROVA DE SUA LICENÇA. QUEIRA CONSIDERÁ-LA COMO UM BEM DE SUA PROPRIEDADE.

Este PROGRAMA de computador é protegido pelas leis de copyright e tratados internacionais, bem como por outras legislações e tratados sobre propriedade intelectual. O PROGRAMA é licenciado e não vendido.

A – **LICENÇA:** A ZEVALLOS TECNOLOGIA EM INFORMAÇÃO (“nós”), entrega meios de armazenamento contendo um programa para computadores (o “Programa”), o manual, a Licença de Utilização (a “Licença”), o cartão de registro e a documentação correspondente (denominados em conjunto com o “SOFTWARE”), e garante a sua autorização para que você utilize o SOFTWARE de acordo com os termos deste Licença. **Os direitos autorais e todos os demais direitos relativos aos SOFTWARE são de nossa propriedade ou de nossos fornecedores. Você deverá reproduzir qualquer notação de direitos autorais ou de outras que constem do SOFTWARE em todas as cópias que venha a fazer.**

B – É PERMITIDO:

- 1 – Usar o Programa em apenas um computador ou rede, e somente por um usuário por vez, independentemente do número de cópias originais do Programa incluídas no SOFTWARE. Se quiser que mais usuários utilizem o Programa, deverá providenciar uma Licença para cada um deles;
- 2 – Fazer uma cópia do SOFTWARE exclusivamente por motivo de segurança e arquivamento, e
- 3 – Transferir o Programa para outra pessoa, desde que você deixe de usar completamente o Programa, apague ou destrua qualquer cópia (incluindo a que está em disco rígido) feita para ajudá-lo na utilização do Programa, transfira todos os seus direitos contidos nessa Licença à outra pessoa e desde que ela concorde com os termos dessa Licença;
- 4 – Se o SOFTWARE for uma atualização de outro SOFTWARE, seja nosso ou de outro fornecedor, você só poderá utilizar ou transferir o SOFTWARE em conjunto com o SOFTWARE atualizado, a menos que o destrua. Se o SOFTWARE for uma atualização de um SOFTWARE Zevallos, só poderá ser usado de acordo com os termos dessa Licença.

C – NÃO É PERMITIDO:

- 1 – Usar o SOFTWARE ou fazer cópias, exceto conforme autorizado nessa Licença;
- 2 – Criar trabalhos escandalosos, obscenos ou imorais usando os exemplos incluídos no Programa;
- 3 – Traduzir, realizar engenharia inversa, descompilar ou desmontar o Programa, exceto se dentro dos limites acima relacionados; isto é expressamente proibido por lei
- 5 – Alugar, emprestar, atribuir ou transferir o SOFTWARE, exceto conforme acima determinado; ou

6 – Modificar o Programa ou fundi-lo, integral ou parcialmente a outro Programa.

D – **VALIDADE:** Esta Licença é válida durante o período em que você utilizar o SOFTWARE. Entretanto, ela se encerrará imediatamente se não for cumprido qualquer um de seus termos ou condições. Você concorda em destruir todas as cópias do SOFTWARE por ocasião do término da Licença. As restrições referentes à Garantia e Responsabilidade determinadas abaixo permanecem em vigor mesmo após o encerramento.

E – **GARANTIA:** Garantimos que os meios de armazenamento desse SOFTWARE não possuem qualquer defeito de fabricação ou de material, por um prazo de 90 (noventa) dias a partir da data de sua aquisição. Se houver defeito, remeta-os para o endereço abaixo e nós o substituiremos gratuitamente. Esse é o nosso único ressarcimento a que você faz jus por quebra desta garantia. Ele lhe dá certos direitos, sendo possível que outros tenham sido determinados conforme a jurisdição.

F – **RESTRIÇÕES DE GARANTIA E DE RESPONSABILIDADE:** À exceção da garantia expressa acima, o SOFTWARE é entregue na condição “no estado”, sem quaisquer garantias ou condições, expressas ou implícitas, incluindo mas não se limitando a, garantias de qualidade comercial, comerciabilidade ou adequação a um objetivo específico, ou o que provier de leis, estatutos, utilização de marcas ou má fé na comercialização. Ficam por sua própria conta todos os riscos referentes aos resultados e desempenho do SOFTWARE. Nem nós, nem nossos distribuidores ou fornecedores, assumirão qualquer responsabilidade em relação a você ou qualquer pessoa ou entidade quanto a qualquer dano indireto, acidental, especial ou consequencial, incluindo mas não se limitando a, perdas de lucros ou receitas, dados perdidos ou danificados, ou qualquer outra perda comercial ou financeira, mesmo que nós tenhamos sido advertidos da possibilidade de tais danos ou que eles fossem previsíveis; ou por qualquer queixa de terceiros. Nossa responsabilidade máxima se refere você, e a de nenhum de nossos distribuidores ou fornecedores irá além da quantia paga por você pela aquisição do SOFTWARE. As limitações contidas nessa seção se aplicam quer a quebra da garantia ou defeito alegados seja uma quebra de uma condição ou termo fundamental, ou uma violação fundamental. Alguns estados/países não permitem a exclusão o limitação da responsabilidade por danos acidentais ou consequências, e assim sendo, as restrições acima podem não se aplicar ao seu caso.

G – **DIREITOS RESTRITOS:** Se esse SOFTWARE for adquirido pelo ou para o Governo dos Estados Unidos, ele terá Direitos Restritos. O uso, duplicação ou quebra de sigilo pelo Governo dos Estados Unidos está sujeito às restrições determinadas no subparágrafos (c) (1) (II) da cláusula The Rights in Technical Data and Computer Software em DFARS252.227-7013 ou subparágrafos (c) (1) e (2) do Commercial Computer Software – Restricted Rights em 48 CFR 52.227 ou a cláusula 18-52.227-86 (d) do NASA Supplement to the FAR, se aplicável. O fornecedor/fabricante é a ZEVALLOS TECNOLOGIA EM INFORMAÇÃO, SCS Quadra 2 Bloco “C” Ed. Jamel Cecílio, Salas 301/302, Brasília, DF, Brasil.

H - **COPYRIGHT.** Todos os títulos e direitos autorais relativos ao SOFTWARE (incluindo quaisquer imagens, fotografias, animações, vídeos, audios, músicas, textos e “applets” incorporados ao SOFTWARE), os materiais impressos que o acompanham, e quaisquer cópias do SOFTWARE são de propriedade da Zevallos ou seus fornecedores. O SOFTWARE é protegido pelas leis de copyright e tratados internacionais. Assim, V.Sa deverá tratar o SOFTWARE como qualquer outro material protegido por direito autoral, exceto que V.Sa poderá **(a)** efetuar uma cópia do SOFTWARE unicamente para fins de salvaguarda ou arquivo, ou **(b)** instalar o SOFTWARE em um computador individual, desde que V.Sa mantenha o original unicamente para fins de salvaguarda ou arquivo. É vedada a reprodução dos materiais impressos que acompanham o SOFTWARE.

I – **GERAL:** Esta Licença representa o acordo único e integral existente entre nós, prevalecendo sobre qualquer outra discussão, verbal ou escrita e não pode ser alterada exceto por documento escrito ou assinado. Essa Licença será regida e interpretada de acordo com as leis do Brasil. Se qualquer termo dessa Licença for declarado, por um Tribunal da jurisdição competente, como inválido, ilegal ou inexecutável, deverá ser removido dessa Licença, sendo que os demais permanecerão em vigor e força integrais.

CONTRATO DE GARANTIA LIMITADA

GARANTIA LIMITADA: A Zevallos garante que

- (a) o SOFTWARE desempenhará suas funções substancialmente em conformidade com seu(s) manual(is), por um período de 90 (noventa) dias a contar da data da entrega;
- (b) Qualquer hardware fornecido pela Zevallos juntamente com o SOFTWARE estará isento de defeitos de fabricação e em relação aos materiais empregados, por um período de 01 (hum) ano, a contar da data de entrega, sob condições normais de utilização e manutenção. Quaisquer garantias implícitas com relação ao SOFTWARE e ao hardware ficam limitadas a 90 (noventa) dias e 01 (hum) ano respectivamente.

DIREITOS DO CLIENTES: A responsabilidade integral da Zevallos, e o único direito de V.Sa. será, a critério da Zevallos,

- (a) a devolução do preço pago, ou, alternativamente;
- (b) o conserto ou substituição do SOFTWARE ou hardware, sujeito aos termos da Garantia Limitada e à devolução dos mesmos à Zevallos com a cópia do recibo. Esta Garantia Limitada ficará prejudicada e não gerará efeitos se o defeito do SOFTWARE ou hardware resultar de acidente, utilização abusiva ou inadequada. Qualquer SOFTWARE substituído será garantido pelo prazo remanescente da garantia original ou por 30 (trinta) dias, no caso deste último prazo se mais extenso.

GARANTIAS ÚNICAS – AS GARANTIAS SUPRA SÃO AS ÚNICAS, QUER EXPRESSAS, QUER IMPLÍCITAS. FICAM EXCLUÍDAS, INCLUSIVE, EXEMPLIFICATIVAMENTE, QUAISQUER GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO E ADEQUAÇÃO PARA DETERMINADA APLICAÇÃO, COM RELAÇÃO AOS SOFTWARE, SEUS MANUAIS E AOS MATERIAIS ESCRITOS E QUALQUER HARDWARE QUE ACOMPANHE O MESMO. ESTA GARANTIA LIMITADA LHE CONFERE DIREITOS ESPECÍFICOS.

LIMITAÇÃO DE RESPONSABILIDADE (DANOS INDIRETOS) – EM NENHUMA HIPÓTESE A ZEVALLOS OU SEUS FORNECEDORES SERÃO RESPONSÁVEIS POR QUAISQUER OUTROS DANOS (INCLUINDO, MAS NÃO LIMITADOS, A LUCROS CESSANTES, INTERRUPÇÃO DE NEGÓCIOS, PERDA DE INFORMAÇÕES E OUTROS PREJUÍZOS PECUNIÁRIOS) DECORRENTES DO USO, OU DA IMPOSSIBILIDADE DE USAR ESTE PRODUTO ZEVALLOS, AINDA QUE A ZEVALLOS TENHA SIDO ALERTADA QUANTO A POSSIBILIDADE DESSES DANOS. EM QUALQUER CASO, A RESPONSABILIDADE INTEGRAL DA ZEVALLOS SOB ESTE CONTRATO LIMITAR-SE-Á AO VALOR EFETIVAMENTE PAGO POR V.Sa. PELO SOFTWARE.

Este Contrato é regido pelas leis da República Federativa do Brasil.

Caso V.Sa. possua dúvida em relação a este Contrato de Licença ou deseje entrar em contacto com a Zevallos, utilize as informações de endereçamento que se encontram neste produto para contactar a subsidiária Zevallos no seu país ou escreva para Atendimento ao Cliente Zevallos, SCS Q.2 Bl. "C" Ed. Jamel Cecílio Salas 301/301 – Brasília – DF – 70317-900.

Guia do Usuário do HiperTools® v3.0

As informações contidas neste documento estão sujeitas a alterações sem prévio aviso. As empresas, os nomes de pessoas e os dados aqui mencionados são fictícios, salvo indicação em contrário. Nenhuma parte deste documento pode ser reproduzida ou transmitida em qualquer forma o por qualquer meio, eletrônico ou mecânico, para qualquer propósito, sem a permissão expressa, por escrito, da Zevallos® Tecnologia em Informação. A Zevallos® poderá ter patentes o pedidos pendentes de patentes, marcas comerciais, direitos autorais ou outros direitos de propriedade intelectual relacionadas aos assuntos tratados neste documento. O fornecimento deste documento não lhe concede licença sobre tais patentes, marcas comerciais, direitos autorais ou outras propriedades intelectuais, exceto nos termos expressamente estipuladas em qualquer contrato de licença da Zevallos®.

© 1996 - 2000 Zevallos® Tecnologia em Informação. Todos os direitos reservados.

O Internet Information Server – IIS foi desenvolvido pela Microsoft Corporation.

Guia do Usuário do HiperTools® v3.0

Índice

Índice 1

O que é o HiperTools® v3.0 2

A Zevallos® 3

A Equipe do HiperTools® 3.0® 4

Serviço de Suporte 5

Listas de Discussão 5

Instalação 7

Apresentação do Ambiente 9

Convenções utilizadas nos programas de exemplo 9

Objeto Barcode 14

Método ShowI2o5 14

Objeto Border 18

Propriedade Color 18

Propriedade Width 18

Objeto Browse 20

Propriedade BarPages 20

Propriedade Current 20

Propriedade Line 21

Propriedade Page 21

Propriedade PageSize 22
Propriedade QueryString 22
Propriedade Recordset 23
Método Bar 23
Método EndOfPage 24
Método IncLine 25

Objeto Character 28

Propriedade Font 28
Propriedade FontDirectory 28
Método DrawString 29
Método DrawButton 29

Objeto Connection 32

Propriedade CommandTimeout 36
Propriedade ConnectionTimeout 37
Propriedade ConnectionType 37
Propriedade Database 38
Propriedade DBQ 39
Propriedade DefaultDir 39
Propriedade Password 40
Propriedade ServerAddress 41
Propriedade UserID 41
Método Clear 42
Método ConnectionString 43

Objeto Default 46

Propriedade BaseFont 46
Propriedade BodyALink 46
Propriedade BodyBackground 47
Propriedade BodyBGColor 47

Propriedade BodyBGProperties 48

Propriedade BodyLeftMargin 48

Propriedade BodyLink 48

Propriedade BodyOnLoad 49

Propriedade BodyOnUnLoad 49

Propriedade BodyStyle 50

Propriedade BodyText 50

Propriedade BodyTopMargin 51

Propriedade BodyVLink 51

Propriedade BodyWidth 52

Propriedade CenterBody 52

Propriedade FooterCopyright 53

Propriedade FooterSugestions 54

Propriedade FooterTitle 54

Propriedade HeaderTitle 54

Propriedade IgnoreBodyWidth 55

Propriedade LinkCopyrightHRef 56

Propriedade LinkCopyrightTitle 56

Propriedade LinkHomeHRef 57

Propriedade LinkHomeTitle 57

Propriedade LinkMadeHRef 58

Propriedade LinkMadeTitle 58

Propriedade LinkStyleSheetHRef 59

Propriedade MetaAuthor 59

Propriedade MetaCategory 60

Propriedade MetaCharSet 60

Propriedade MetaContentType 61

Propriedade MetaDescription 61

Propriedade MetaDistribution 62

Propriedade MetaExpires 62

<i>Propriedade MetaFormatter</i>	62
<i>Propriedade MetaGenerator</i>	63
<i>Propriedade MetaKeyPhrase</i>	63
<i>Propriedade MetaKeywords</i>	64
<i>Propriedade MetaLocale</i>	64
<i>Propriedade MetaPicsLabel</i>	65
<i>Propriedade MetaPragma</i>	65
<i>Propriedade MetaProduct</i>	66
<i>Propriedade MetaRefresh</i>	66
<i>Propriedade MetaReplyTo</i>	67
<i>Propriedade MetaTechnology</i>	67
<i>Propriedade PageFooter</i>	67
<i>Propriedade PageHeader</i>	68
<i>Propriedade SiteCounter</i>	68
<i>Propriedade SourceCode</i>	69
<i>Propriedade TableBorder</i>	69
<i>Propriedade UserID</i>	70
<i>Método BeginBody</i>	70
<i>Método BeginHead</i>	71
<i>Método BeginHTML</i>	72
<i>Método EndBody</i>	73
<i>Método EndHead</i>	74
<i>Método EndHTML</i>	75
<i>Método PageFooterDefault</i>	76
<i>Método WriteBaseFont</i>	77
<i>Método WriteHead</i>	78
<i>Método WriteLink</i>	79
<i>Método WriteMeta</i>	80
<i>Método WritePageFooter</i>	82
<i>Método WritePageHeader</i>	82

Objeto EditForm 86

Constantes Validation Locations 89
Constantes QueryString Parameters 89
Constantes QueryString Action Values 91
Constantes Data Types 92
Constantes Field Requirement 94
Constantes Relational Integrity 94
Constantes Relation Type 95
Constantes Field Types 95
Constantes Validation Options 97
Constantes Field Disable Condition 99
Constantes Operator types 100
Constantes Character Case 100
Constantes Alphabetic Index Types 101
Constantes Connection Types 101
Constantes Identity 102
Propriedade AllowIdentityUpdate 102
Propriedade AlphabeticIndex 103
Propriedade Buttons 103
Propriedade ConnectionString 105
Propriedade ConnectionType 106
Propriedade CurrConnection 106
Propriedade DebugMode 107
Propriedade Errors 107
Propriedade FirstPageLink 108
Propriedade FilterBooleanOperator 108
Propriedade LinkTarget 109
Propriedade ListTitle 109
Propriedade ListReportsCaption 110
Propriedade ListSetOfOrdersCaption 110

<i>Propriedade LockTimeOut</i>	111
<i>Propriedade Password</i>	111
<i>Propriedade parWhat</i>	112
<i>Propriedade parAction</i>	112
<i>Propriedade ReportTitle</i>	113
<i>Propriedade ShowButtons</i>	113
<i>Propriedade ShowListUpperBar</i>	114
<i>Propriedade ShowSetOfOrdersLink</i>	114
<i>Propriedade ShowRecordCount</i>	114
<i>Propriedade ShowSQLQuery</i>	115
<i>Propriedade ShowReportsLink</i>	115
<i>Propriedade ShowUnitUpperBar</i>	116
<i>Propriedade Style</i>	116
<i>Propriedade UnitTitle</i>	116
<i>Propriedade UserID</i>	117
<i>Método IsMyAction</i>	117
<i>Método FieldValue</i>	118
<i>Método HasTable</i>	118
<i>Método ShowActions</i>	119
<i>Método RedirectActions</i>	119
<i>Método OpenConnection</i>	120
<i>Método DataTable</i>	121
<i>Método DataAddField</i>	121
<i>Método DataAddPrimaryKey</i>	122
<i>Método DataAddRelation</i>	123
<i>Método DataAddRelationFields</i>	124
<i>Método DataAddIndex</i>	125
<i>Método DataIndexClustered</i>	126
<i>Método BeginForm</i>	126
<i>Método EndForm</i>	128

Método FormFind 129

Método FormList 130

Método FormUnit 131

Método FormTabs 132

Método AddField 133

Método AddOrder 134

Método AddFilter 135

Método AddHeader 136

Método AddReport 137

Método EmptyListMessage 139

Método FieldAlphabeticIndex 139

Método FieldTextAreaHeight 140

Método FieldImage 141

Método FieldRadioColumns 143

Método FieldCharCase 144

Método FieldDisableValue 145

Método FieldListSize 146

Método FieldListChars 147

Método FieldMask 149

Método FieldShowSize 149

Método FieldSepDate 151

Método FieldExternalLink 151

Método FieldInternalLink 152

Método FieldAddILTarget 153

Método FieldAddValue 153

Método FieldHint 154

Método FieldLookup 155

Método FieldLookupFilter 156

Método FieldDoubleValidate 157

Método FieldIsUnique 157

Método FieldIsUnique 158

Método CreateTable 159

Método DropTable 159

Método CreateIndex 160

Objeto File 164

Método Exists 164

Método Copy 164

Método ExtractFileName 165

Método DateCreated 166

Método DateLastModified 166

Método Size 167

Objeto Font 170

Propriedade Bold 170

Propriedade Color 170

Propriedade Face 171

Propriedade Italic 171

Propriedade Size 171

Propriedade Underline 172

Método BeginFont 172

Método EndFont 173

Objeto Form 176

Propriedade AccessKey 176

Propriedade Alt 176

Propriedade Class 177

Propriedade Enable 177

Propriedade EncType 177

Propriedade MaxLength 178

Propriedade onChange 178

Propriedade onClick 179

Propriedade ReadOnly 179

Propriedade Title 179

Método BeginForm 180

Método BeginSelect 181

Método CheckBox 182

Método EndForm 183

Método EndSelect 184

Método File 184

Método FormInput 185

Método Hidden 186

Método Image 187

Método Palette 188

Método Password 189

Método Radio 190

Método Reset 191

Método SelectChooseOption 191

Método SelectCountry 193

Método SelectMonthName 202

Método SelectOption 203

Método SelectState 204

Método SelectWeekDay 207

Método Submit 208

Método Text 209

Método TextArea 210

Objeto Initializer 214

Propriedade ErrorMessage 214

Propriedade HiperToolsWorkPath 214

Propriedade HTTPHost 215
Propriedade LastQueryString 215
Propriedade QueryString 216
Propriedade ScriptFull 216
Propriedade ScriptName 217
Propriedade ScriptURL 217
Propriedade SiteRootDir 218
Propriedade WhenStarted 218
Método InitDefault 219

Objeto Pack 222

Propriedade Count 223
Propriedade Item 224
Método Add 225
Método Find 226
Método Remove 226

Objeto PackItem 230

Propriedade Key 230
Propriedade Pack 230
Propriedade Position 231
Propriedade Value 231

Objeto Path 234

Método Drive 234
Método InvertSlash 234
Método NormalSlash 235
Método Only 235
Método RemoveLastSlash 236
Método SlashType 236
Método URLSlash 237

Objeto Registry 240

Propriedade AutoCreate 240

Propriedade HKEY 240

Propriedade HKEYName 241

Propriedade Key 241

Propriedade SubKeys 242

Propriedade Values 242

Método CreateKey 243

Método CreateValue 243

Método DeleteKey 244

Método DeleteValue 245

Método ValueAsBinary 245

Método ValueAsBoolean 246

Método ValueAsLong 246

Método ValueAsString 247

Objeto RegistrySubKey 250

Propriedade LastWrite 250

Propriedade Name 250

Objeto RegistrySubKeys 254

Propriedade Count 254

Propriedade Item 254

Objeto RegistryValue 258

Propriedade DataType 258

Propriedade Name 258

Objeto RegistryValues 262

Propriedade Count 262

Propriedade Item 262

Objeto Show 266

Propriedade AutoWrite 266
Propriedade CrLf 266
Propriedade CrLfAfter 267
Propriedade CrLfBefore 267
Propriedade ShowComments 268
Método AllServerVariables 268
Método BeginItemList 269
Método Bold 270
Método BoldEnd 270
Método BoldText 271
Método BR 271
Método Center 272
Método CenterEnd 272
Método CRHTML 273
Método CRHTMLCR 273
Método EndItemList 274
Método HTML 275
Método HTMLComment 275
Método HTMLCR 276
Método Image 276
Método Italic 277
Método ItalicEnd 278
Método ItalicText 278
Método ItemList 279
Método ItemListError 280
Método JSAlert 280
Método Link 281
Método LinkRel 282
Método LinkRelStyleSheet 282

Método LinkRev 283

Método Message 283

Método MessageAlert 284

Método MessageError 285

Método Meta 285

Método MetaHTTPEquiv 286

Método MetaName 286

Método Nbsp 287

Método Paragraph 287

Método Underline 288

Método UnderlineEnd 288

Método UnderlineText 289

Objeto String 292

Propriedade DateSeparator 292

Propriedade Piece 292

Método BreakString 293

Método DayOfWeek 293

Método FormatCGC 294

Método FormatCPF 295

Método FormatDigit 295

Método FormatInt 296

Método FormatMoney 297

Método FormatNumber 297

Método FormatPhone 298

Método GetPiece 299

Método GetPieceAsDate 300

Método GetPieceAsStr 300

Método GetPieceRest 301

Método GetWord 302

<i>Método ImportDate</i>	302
<i>Método ImportMoney</i>	303
<i>Método ImportInt</i>	305
<i>Método ImportLong</i>	305
<i>Método LeadingZeroes</i>	306
<i>Método LongDate</i>	306
<i>Método LongDateTime</i>	307
<i>Método LongTime</i>	308
<i>Método LTrimZeroes</i>	308
<i>Método Mask</i>	309
<i>Método MonthName</i>	310
<i>Método MonthSize</i>	311
<i>Método NormalizeAccents</i>	312
<i>Método NormalizeNumber</i>	312
<i>Método NormalizeString</i>	313
<i>Método Reverse</i>	313
<i>Método SecondsAsTime</i>	314
<i>Método ShortTime</i>	315
<i>Método ShowPiece</i>	315
<i>Método SkipPiece</i>	316
<i>Método SQLLongDate</i>	317
<i>Método SQLLongDateBR</i>	317
<i>Método SQLDateTime</i>	318
<i>Método SQLLong</i>	318
<i>Método SQLString</i>	319
<i>Método TimeAsSeconds</i>	320
<i>Método ToString</i>	320
<i>Método USADate</i>	321
Objeto Table	324

Propriedade CellAlign 324
Propriedade CellColSpan 324
Propriedade CellHeight 325
Propriedade CellNoWrap 325
Propriedade CellRowSpan 326
Propriedade CellVAlign 326
Propriedade CellWidth 327
Propriedade GraphMax 328
Propriedade GraphTotal 328
Propriedade Padding 328
Propriedade ShowBorder 329
Propriedade Spacing 329
Propriedade Style 330
Método BeginCell 330
Método BeginRow 331
Método BeginTable 332
Método Cell 333
Método CellGraph 334
Método EndCell 335
Método EndRow 336
Método EndTable 337
Método GraphLegend 338

Objeto TableStyle 342

Propriedade AlternateColor 342
Propriedade BackgroundFormat 342
Propriedade BaseColor 343
Propriedade BorderColor 343
Propriedade BorderFormat 344
Propriedade BorderWidth 344

Propriedade CellBackground 345
Propriedade ColorFormat 345
Propriedade ExternalBorder 345
Propriedade FirstAltBackground 346
Propriedade FirstAltColor 346
Propriedade FontColor 347
Propriedade FontSize 347
Propriedade FontBold 348
Propriedade FontItalic 348
Propriedade FontUnderline 349
Propriedade HeaderBorder 349
Propriedade HeaderCellBackground 350
Propriedade HeaderColor 350
Propriedade HeaderFont 351
Propriedade HeaderRowBackground 351
Propriedade InternalBorder 351
Propriedade InternalFont 352
Propriedade LastColor 352
Propriedade NeedTwoTables 353
Propriedade RowBackground 353
Propriedade SecondAltBackground 354
Propriedade TableBackground 354
Propriedade TitleColor 355
Propriedade TitleFont 355

Objeto Tip 358

Propriedade ContentClass 358
Propriedade OutColor 358
Propriedade OverColor 359
Propriedade TitleClass 359

Método AddTip 360

Método AddTipBegin 360

Método AddTipEnd 361

Método BeginTip 362

Método EndTip 362

Objeto TreeView 366

Propriedade DefaultImage 368

Propriedade ItemBGColor 368

Propriedade ItemFontColor 369

Propriedade ItemFontFace 369

Propriedade ItemFontSize 370

Propriedade ItemBold 370

Propriedade ItemItalic 371

Propriedade ItemUnderline 371

Propriedade ItemNoWrap 372

Propriedade ItemHAlign 372

Propriedade ItemVAlign 373

Propriedade ItemTarget 374

Propriedade ItemWidth 374

Propriedade ItemHeight 375

Propriedade TreeBGColor 375

Propriedade ItemMaxSize 376

Propriedade ItemStyle 376

Propriedade TreeStyleLink 377

Propriedade TreeStyleActive 377

Propriedade TreeStyleVisited 378

Propriedade TreeStyleHover 378

Propriedade CurrentChild 379

Propriedade SetLevelID 380

Propriedade GetLevelID 380
Propriedade IsLevelID 380
Propriedade ItemHaveMore 381
Propriedade ItemState 381
Método BeginTree 382
Método EndTree 382
Método BeginNode 383
Método EndNode 384
Método AddItem 385
Método AddSubCaption 385
Método AddImage 386

Objeto Upload 390

Coleções Form 393
Coleções Items 393
Propriedade MaxBytes 394
Propriedade OverwriteFiles 394
Propriedade Path 395
Propriedade ServerName 395
Método Save 396
Método SaveAs 397

Objeto UploadItem 400

Propriedade IsFile 400
Propriedade Name 401
Propriedade Value 401
Propriedade UserFilename 402
Propriedade TotalBytes 402
Propriedade ServerName 403
Método Save 403
Método SaveAs 404

Objeto UploadItems 408

Propriedade Count 408

Propriedade Item 408

Propriedade Name 409

Propriedade Value 409

Objeto URL 412

Propriedade Ampersand 412

Propriedade Parameters 412

Método Add 413

Método AddEditForm 414

Método BeginURL 415

Método EndURL 415

Método Equal 416

Método GetURL 417

Método Show 417

Objeto Validate 420

Propriedade AllowJSAlert 420

Propriedade JSStart 420

Método BeginJava 421

Método BeginJS 422

Método CompareDate 423

Método EndJava 423

Método EndJS 424

Método FormatDate 425

Método FormatTime 426

Método FormJavaCompareDate 426

Método FormJavaCompareDateSep 427

Método FormJavaValidCEP 428

<i>Método FormJavaValidCGC</i>	428
<i>Método FormJavaValidCPF</i>	429
<i>Método FormJavaValidEmail</i>	429
<i>Método FormJavaValidSepDate</i>	430
<i>Método FormJavaValidDate</i>	430
<i>Método IsCGC</i>	431
<i>Método IsCPF</i>	432
<i>Método IsEmail</i>	432
<i>Método IsValidDate</i>	433
<i>Método IsValidTime</i>	433
<i>Método JavaCompare</i>	434
<i>Método JavaClearField</i>	434
<i>Método JavaEmpty</i>	435
<i>Método JavaFocus</i>	436
<i>Método JavaRequired</i>	436
<i>Método JSMaskVal</i>	437
<i>Método JSDateToString</i>	437
<i>Método JSIsDate</i>	438
<i>Método JSLeadingZeroes</i>	438
<i>Método JSLeft</i>	439
<i>Método JSMid</i>	439
<i>Método JSMidToEnd</i>	440
<i>Método JSNormalize</i>	440
<i>Método JSStringToDate</i>	441

Introdução

O que é o HiperTools® v3.0

O HiperTools® v3.0 é um poderoso conjunto de objetos que visam aumentar a produtividade de sistemas com a Interface Web.

Desenvolvido utilizando o que há de mais moderno na tecnologia Web, o HiperTools® é a sua melhor escolha.

Com ele você e a sua equipe de desenvolvimento conseguirá acelerar o processo de desenvolvimento, obtendo melhores resultados, já que seus objetos integram com pequenos comandos grandes facilidades.

Todo o conjunto de objetos foi desenvolvido visando reduzir a necessidade do relacionamento do programador com a linguagem HTML. Com o HiperTools® v3.0, o desenvolvedor usará o HTML somente para pequenos ajustes estéticos.

O HiperTools® v3.0 é composto por **24** objetos principais e 7 objetos que implementam propriedades destes:

Barcode	Initializer	String
Browse	Pack	Table
Character	PackItem	TableStyle
Connection	Path	Tip
Default	Registry	TreeView
Directory	RegistrySubKey	URL
EditForm	RegistrySubKeys	Upload
File	RegistryValue	UploadItems
Font	RegistryValues	Validate
Form	Show	

Fácil de aprender e de usar

O aprendizado do HiperTools® é muito rápido, devido aos vários métodos que visam abstrair o HTML do desenvolvedor, de forma a obter um código limpo e legível mesmo por outros programadores.

Todas as propriedades e métodos dos objetos foram criados para serem intuitivos que durante o desenvolvimento o programador irá lembrá-los praticamente sem a necessidade de ir ao manual muitas vezes.

Para iniciar o desenvolvimento de sistemas com a Interface Web, basta ter conhecimento de qualquer linguagem.

O HiperTools® v3.0 funciona muito bem com as linguagens VBScript, JScript e PerlScript que estão disponíveis no servidor Microsoft IIS 4.0.

Saída de HTML Puro

A maioria dos objetos do HiperTools® v3.0 geram puro HTML independente de plataforma e browser. Não têm controles ActiveX, eventualmente utilizam JavaScript validações no cliente ou para manipulação de DHTML.

Requerimentos do Sistema

Para utilizar os objetos, você precisará ter no seu equipamento pelo menos o Microsoft Internet Information Server (IIS) versão 4.0 para o Microsoft Windows NT v4.0 ou o Microsoft Personal Web Server (PWS) para o Microsoft Windows 95 ou Windows 98.

Compatibilidade de Browser

Os objetos do HiperTools® v3.0 são compatíveis dos maiores browsers, por causa do HTML que eles geram. Eles vão funcionar corretamente com qualquer dos Browsers abaixo:

Internet Explorer versions 4.0 e superiores

Netscape Navigator versions 4.0 e superiores

Compatibilidade de Código

Os objetos do HiperTools® v3.0 usam as mesmas propriedades e métodos que os controles comuns utilizados no Microsoft Visual Basic. Isso possibilita você a copiar e colar seu código existente no seu script ASP.

ActiveX

O HiperTools® é um componente ActiveX.DLL que se integra com o Active Server Pages – ASP do Internet Information Server 4.0 – IIS de forma transparente e sem a necessidade de executar comandos complexos ou obscuros.

Após o registro do arquivo HiperTools30.DLL, você já pode iniciar a utilizar o HiperTools® v3.0 e passar a usufruir das facilidades contidas em seus objetos.

Mult Thread e Apartment Model

O HiperTools® v3.0 foi desenvolvido de forma utilizar poucos recursos de memória e processamento do servidor. Por isso que escolhemos o sistema Mult Thread e Apartment Model.

Desta forma é possível que o seu programa tenha centenas ou milhares de acessos sem que haja uma grande queda dos recursos do servidor.

Principais Facilidades

- Abstração quase que total do código HTML;
- Acesso aos principais servidores de banco de dados SQL;
- Formulários de inclusão, alteração, exclusão e localização;

A Zevallos®

A Zevallos® Tecnologia em Informação iniciou suas atividades em novembro/1996 procurando desenvolver tecnologias para acelerar o desenvolvimento de aplicações com a Interface Web.

Não somente isso, procuramos transformar o programador de Scripts para Internet, em verdadeiros programadores, onde o VBScript ou o Jscript seriam mais uma linguagem para o desenvolvimento e o Browser mais um terminal para apresentação de informações.

O Active Server Pages – ASP, tem uma grande característica de misturar o HTML com a linguagem utilizando os marcadores <% e %>. Desta forma, o programador pode misturar o HTML ao seu código ASP.

Esta forma de programação difuculta em muito a visualição do código bem como a manutenção.

Então procuramos abstrair do programador a mistura de código com o HTML. Foi iniciado então a criação do HiperTools, mas isso somente não nos satisfiz, porque o processo de desenvolvimento ainda estava dificultado, bem como a baixa produtividade.

Então criamos também uma forma especial de organizar e documentar os códigos, como apresentado neste documento.

Nossa equipe é formada vários profissionais com as mais variadas experiências na área de desenvolvimento.

Zevallos® Tecnologia em Informação

SCS Quadra 2 Bloco “C” Ed. Jamel Cecílio, Cobertura

Setor Comercial Sul

70000-000 - Brasília – DF – Brasil

+55 61 321-4711 Voz

+55 61 323-7267 Fax

<http://www.zevallos.com.br>

info@zevallos.com.br

A Equipe do HiperTools® 3.0®

A Zevallos Tecnologia em Informação tem uma equipe de desenvolvedores com várias especializações que participaram das várias partes do projeto. Sendo que a principal atividade foi a de utilizar exaustivamente o produto internamente para que o HiperTools® v3.0 fosse realmente uma grande ferramenta e principalmente sem erros.

Técnicos	Atuação
Eduardo Alves	Desenvolvedor do EditForm e vários outros objetos e o principal componentizador do HiperTools® v3.0
Eduardo Gonçalves	Validador das Interfaces e Desenvolvedor do HiperTools® v3.0
Flávio Costa	Desenvolvedor do Registry, corretor de várias interfaces e revisor do manual
Fernado Aquino	Validador das Interfaces e Desenvolvedor do HiperTools® v3.0
Kelson Medeiros	Desenvolvedor do TreeView e desenvolvedor HiperTools® v3.0

Ridai Pombo	Componentizador e corretor de várias interfaces, desenvolvedor HiperTools® v3.0
Ruben Zevallos Jr.	Idealizador e desenvolvedor do HiperTools® v1.0 ao 2.0 e coordenador do projeto.
Ueslei	Validador das Interfaces e desenvolvedor HiperTools® v3.0

Como desenvolvimento não termina somente com o desenvolvimento e testes do produto, o HiperTools® v3.0 teve outras pessoas que participaram do processo da colocação do produto do mercado.

Sendo:

Ruben Zevallos Jr. – Redação dos manuais Guia do Usuário e Usando O HiperTools® v3.0

Priscilla Zevallos – Revisão dos textos dos manuais

Wladimir Ferreira – Definição do formato de comercialização do produto

Cláudia – Secretariado do projeto, bem como a produção das cópias BETA e materiais diversos

Serviço de Suporte

Zevallos® Tecnologia em Informação

SCS Quadra 2 Bloco “C” Ed. Jamel Cecílio, Salas 301/302

Setor Comercial

70317-900 - Brasília – DF – Brasil

+55 61 321-4711 Voz

+55 61 323-7267 Fax

<http://www.zevallos.com.br/>

info@zevallos.com.br

HiperTools® v3.0 Web Site (Português)

<http://www.HiperTools.com.br>

info@HiperTools.com.br

HiperTools® v3.0 Web Site (Inglês)

<http://www.HiperTools.com>

info@HiperTools.com.br

Listas de Discussão

Criamos várias listas de discussão visando a troca de informações e experiências dos usuários do HiperTools®, bem como das tecnologias para desenvolvimento para Interface Web.

Para assinar as listas abaixo, você precisa enviar uma mensagem para a caixa postal lists@HiperTools.com.br com o comando SUBSCRIBE de acordo com o exemplo abaixo:

Para assinar a lista HiperTools-L envie no corpo da mensagem o seguinte comando:

SUBSCRIBE HiperTools-L Nome Sobrenome

Para cancelar a assinatura, envie no corpo da mensagem o seguinte comando:

SUBSCRIBE HiperTools-L

As seguinte listas estão disponíveis para assinatura:

Lista	Descrição
HiperTools-L	Lista para discussão e informação sobre o HiperTools.
ASP-L	Lista para discussão geral sobre o ASP.
VBScript-L	Lista para discussão sobre o VBScript.
JScript-L	Lista para discussão sobre o JScript.

Instalação

A instalação é muito simples, sem precisar de grandes ajustes.

Necessidades mínimas recomendadas

Hardware	
Processador:	Pentium 133
Memória:	64MB
HD:	2GB

Software	
SO	Windows NT 4.0 SP4
HTTP Server	IIS 4.0
Componente BD	MDAC 2.1
MS-SQL	6.5 SP5

Estrutura de diretórios recomendada:

Crie um diretório no seu servidor HTTP (Web), com o nome HiperTools® de acordo com a sua configuração. Nossa configuração tem a seguinte estrutura de diretórios:

Diretório	Descrição
D:\Inetpub	Diretório raiz do IIS 4.0
D:\Inetpub\wwwroot	Diretório raiz do servidor web
D:\Inetpub\wwwroot\components	Diretório para todos os componentes do servidor
D:\Inetpub\wwwroot\components\HiperTools	Diretório para todos os arquivos do HiperTools® no servidor
D:\Inetpub\wwwroot\Clients	Diretório para os sites dos clientes
D:\Inetpub\wwwroot\Projects	Diretório para os sites dos projetos internos

Copiando os arquivos do HiperTools® v3.0 para o servidor:

Extraia todos os arquivos do HiperTools.zip no D:\Inetpub\wwwroot\components\HiperTools® ou no diretório equivalente no seu servidor.

Depois da extração, você terá a seguinte lista de arquivos.

Arquivo	Descrição
---------	-----------

Adovbs.inc	Arquivo com os includes utilizados para controlar o ADO
Constants.inc	Arquivo com a chamada para todos os outros arquivos de constantes .INC
EditForm.inc	Arquivo de constantes do objeto EditForm
String.inc	Arquivo de constantes do objeto String
HiperTools30.dll	O componente ActiveX do HiperTools® v3.0o
HiperTools30.pdf	Guia do Usuário do HiperTools® v3.0 em Acrobat 3.0

Registrando o componente HiperTools® v3.0:

No prompt do DOS, vá no diretório do HiperTools® v3.0 e digite o comando abaixo, que o HiperTools® v3.0 será registrado.

REGSVR32 HIPERTOOLS30.DLL

Cancelando o registro do componente HiperTools® v3.0:

No prompt do DOS, vá no diretório do HiperTools® v3.0 e digite o comando abaixo, que o registro do HiperTools® v3.0 será cancelado.

REGSVR32 /u HIPERTOOLS30.DLL

Configurando um site no servidor IIS:

Recomendamos sempre utilizada um dominio virtual para cada projeto que você for criar dentro do seu servidor. Desta forma, você poderá controlar todos os ambientes, além de ter um global.asa diferente para cada projeto.

Apresentação do Ambiente

Nós da Zevallos® desenvolvemos um ambiente de desenvolvimento, visando facilitar a interação entre desenvolvedores em um mesmo projeto ou em projetos diferentes.

São diversas definições que vem do ambiente de trabalho até a definição das colunas nas tabelas do banco de dados.

As informações a seguir são de cunho informativo e também para que o leitor consiga compreender os exemplos utilizados neste manual, bem como nos existentes no nosso site <http://www.HiperTools.com.br>.

Convenções utilizadas nos programas de exemplo

Convenção para nomenclatura de variáveis

Tipo	Prefixo	Exemplo
Boolean	bln	blnFound
Byte	byt	bytRasterDate
Date (Time)	dtm	dtmStart
Double	dbl	dblTolerance
Error	err	errOrderInt
Integer	int	intQuantity
Long	lng	lngDistance
Objetct	obj	objCurrent
Single	sng	sngAverage
String	str	strFirstName
Constant	con	conAppName
Binary	Bin	binImage

Escopo das variaveis

Escopo	Prefixo	Exemplo
Procedure level	none	strFirstName
Script Level	s	sstrFirstName

Convenção para nomenclatura de objetos

Tipo do Objeto	Prefixo	Exemplo
3D Panel	pnl	pnlGroup
Animated button	ani	aniMailBox
Check box	chk	chkReadOnly

Instalação

Combo Box, drop-down list box	cbo	cboEnglish
Command button	cmd	cmdExit
Common dialog	dlg	dlgFileOpen
Frame	fra	fraLanguage
Horizontal scroll bar	hsb	hsbVolume
Image	img	imgIcon
Label	lbl	lblHelpMessage
Line	lin	linVertical
List Box	lst	lstPolicyCodes
Spin	spn	spnPages
Text box	txt	txtLastName
Vertical scroll bar	vsb	vsbRate
Slider	sld	sldScale
Parameter	par	parConversionSource
Radio Button	rad	radSourceType

Sistema de padronização da documentação

Padrão	Descrição
[conchetes]	Os itens dentro de colchetes são opcionais. Para a inclusão de informações opcionais digite somente a informação entre os colchetes; não digite os colchetes.
... (reticencias)	As reticencias (...) significam que um item poderá repetir quantas vezes quanto forem necessárias.
- separadores	A menos que seja especificado o contrario, os comandos devem ser separados das opções utilizando espaços.
Funções e Subs	Nomes com as iniciais maiusculas, bem detalhados, utilizando o bom senso. Ex: SaveScreen() (é mais facil de utilizar e visualizar)

Cabeçalho dos programas

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#INCLUDE VIRTUAL="/HiperTools/Constants.inc"-->
<!--#INCLUDE VIRTUAL="/Objects.inc"-->
<%
REM -----
REM  Contratos.asp
REM -----
```

```

REM  Descricao   : Gerencia dos contratos do provedor ZYC
REM  Criacao     : 12/07/99 00:04
REM  Local       : Brasilia/DF
REM  Elaborado    : Ruben Zevallos Jr. <zevallos@zevallos.com.br>
REM  Versao      : 1.0.0
REM  Copyright    : 1999 by Zevallos(r) Tecnologia em Informacao
REM  -----
%>

```

Padronização dos Programas

Comandos com as iniciais maiusculas e com

Separar com um espaco todos os argumentos das expressoes aritmeticas e funcoes, sem excecao.

REM Ex: strAux = Chr(intA + intB + Int(strX))

Objeto Barcode

Objeto Barcode

O objeto Barcode permite a criação de códigos de barra no formato "i2o5" para a visualização em páginas web.

Sintaxe

```
Set Barcode = Server.CreateObject("HiperTools30.Barcode")
```

Método ShowI2o5

Descrição

Desenha o código de barras do valor informado no resultado HTML da página.

Sintaxe

objeto.**ShowI2o5** *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. <i>String</i> é o valor que deverá ser desenhado.

Comentários

O código abaixo ilustra o uso do método **ShowI2o5**.

```
Table.Padding = 8
Table.Spacing = 0

Table.BeginTable 650
  Table.BeginRow

    Table.ColumnVAlign = "TOP"
    Table.BeginColumn
      BarCode.ShowI2o5(strCodigoBarras)
    Table.EndColumn

    Table.ColumnAlign = "Center"
    Table.ColumnWidth = 220

    Table.BeginColumn
      Show.HTML "Autentica&ccedil;&atilde;o Mec&acirc;nica"
      Show.Message <B>Ficha de Compensação</B></FONT><BR>"
    Table.ColumnAlign = " "

  Table.EndRow
Table.EndTable
```

Aplica-se

Barcode.

Veja também

Objeto Border

Objeto Border

Este objeto armazena informações sobre bordas. Esse objeto é usado internamente como propriedade de outros objetos.

Sintaxe

Esse é um objeto de uso interno do sistema e não deve ser instanciado via código.

Propriedade Color

Descrição

Especifica a cor da borda.

Sintaxe

objeto.**Color** [= *String*]

objeto sempre será o objeto *Font*.

Comentários

Os valores válidos para essa propriedade são os mesmos do HTML, especificado como nome da cor ou valor RGB em hexadecimal.

O código abaixo ilustra o uso da propriedade **Color**.

```
TableStyle.Border.Color = "#F0F0F0"
```

Aplica-se

Border

Veja também

Propriedade Width

Descrição

Especifica, em pixels, a espessura da borda.

Sintaxe

objeto. **Width** [= *String*]

objeto sempre será o objeto *Font*.

Comentários

O código abaixo ilustra o uso da propriedade **Width**.

```
TableStyle.Border.Width = "3"
```

Aplica-se

Border

Veja também

Objeto Browse

Objeto Browse

O objeto Browse tem por finalidade criar listagens de dados selecionados a partir de uma consulta em um banco de dados, criando uma barra de navegação entre páginas com um determinado número de registros por página.

Sintaxe

Set Browse = Server.CreateObject("HiperTools30.Browse")

Propriedade BarPages

Descrição

Especifica a quantidade de páginas a serem apresentadas na barra de navegação.

Sintaxe

objeto.**BarPages** [= *Integer*]

O argumento *objeto* sempre será o objeto Browse.

Comentários

O valor para **BarPages** é o número máximo de páginas que aparecerá na barra de navegação.

O código abaixo ilustra o uso da propriedade **BarPages**.

```
Browse.Current = "MyBrowse"  
Browse.PageSize = 10  
Browse.BarPages = 15  
Browse.Recordset = subjRS
```

Aplica-se

Browse

Veja também

Propriedade Current

Descrição

Identificação do Browse atual. Essa propriedade é particularmente útil caso se deseje exibir duas listagens na mesma página.

Sintaxe

objeto.**Current** [= *String*]

O argumento *objeto* sempre será o objeto Browse.

Comentários

O valor para **Current** deverá ser uma *String* que identifique o browse atual.

O código abaixo ilustra o uso da propriedade **Current**.

```
Browse.Current = "MyBrowse"  
Browse.PageSize = 10  
Browse.Recordset = subjRS
```

Aplica-se

Browse

Veja também

Propriedade Line

Descrição

Retorna a quantidade de linhas apresentadas pelo objeto Browse na página atual. Essa propriedade é *read-only*.

Sintaxe

[*Integer* =] objeto.**Line**

O argumento objeto sempre será o objeto Browse.

Comentários

O código abaixo ilustra o uso da propriedade **Line**.

```
Browse.Current = "MyBrowse"
Browse.PageSize = 10
Browse.Recordset = sobjRS

Table.BeginTable "400", "Listagem", 3, True
Do Until (sojRS.EOF Or Browse.EndOfPage)
    Table.BeginRow 2
    Table.Column sobjRS("Nome")
    Table.Column sobjRS("Endereco")
    Table.Column sobjRS("Telefone")
    Table.EndRow

    sobjRS.MoveNext
    Browse.IncLine

Loop
Browse.Status = True
Response.Write Browse.Line
Retorna: 10 se houver mais de 10 registros no Recordset Senão retorna a
quantidade de registros.

Browse.Bar
```

Aplica-se

Browse

Veja também

Propriedade Page

Descrição

Especifica a página atual que o Browse deve apresentar.

Sintaxe

objeto.**Page** [= *Integer*]

O argumento *objeto* sempre será o objeto Browse.

Comentários

O código abaixo ilustra o uso da propriedade **Page**.

```
Browse.Current = "MyBrowse"  
Browse.Page = 1  
Browse.Recordset = subjRS
```

Aplica-se

Browse

Veja também

Propriedade PageSize

Descrição

Especifica a quantidade de registros a ser exibida em cada página.

Sintaxe

objeto.**PageSize** [= *Integer*]

O argumento *objeto* sempre será o objeto Browse.

Comentários

O código abaixo ilustra o uso da propriedade **PageSize**.

```
Browse.Current = "MyBrowse"  
Browse.PageSize = 10  
Browse.Recordset = subjRS
```

Aplica-se

Browse

Veja também

Propriedade QueryString

Descrição

Especifica a QueryString com os parâmetros atuais do Browse.

Sintaxe

objeto.**QueryString** [= *String*]

O argumento *objeto* sempre será o objeto Browse.

Comentários

O valor da propriedade *QueryString* define os parâmetros que são passados nos links criados pelo objeto *Browse*.

O código abaixo ilustra o uso da propriedade **QueryString**.

```
Response.Write Browse.QueryString
```

Aplica-se

Browse

Veja também

Propriedade Recordset

Descrição

Especifica o Recordset (conjunto de registros) que será utilizado pelo Browse para fazer a paginação.

Sintaxe

objeto.**Recordset** [= *Recordset*]

O argumento *objeto* sempre será o objeto Browse.

Comentários

O valor para **Recordset** deve ser um objeto Recordset do ADO que o Browse utilizará para navegar.

O código abaixo ilustra o uso da propriedade **Recordset**.

```
Dim objRS, strSQL
Set objRS = Server.CreateObject("ADODB.Recordset")
strSQL = "SELECT * FROM tabela"
objRS.Open strSQL, objConnection, adOpenKeyset, adLockPessimistic, adCmdText
Browse.Recordset = objRS
```

Aplica-se

Browse

Veja também

Método Bar

Descrição

Apresenta a barra de navegação do objeto Browse.

Sintaxe

objeto.**Bar**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto Browse.

Comentários

O código abaixo ilustra o uso do método **Bar**.

```
Browse.Bar
```

Aplica-se

Browse

Veja também

Método EndOfPage

Descrição

Retorna se a próxima linha a ser apresentada pertence à página atual.

Sintaxe

objeto.**EndOfPage**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto Browse.

Comentários

O código abaixo ilustra o uso do método **EndOfPage**.

```
Browse.Current = "MyBrowse"
Browse.PageSize = 10
Browse.Recordset = subjRS

Table.BeginTable "400", "Listagem", 3, True
Do Until (subjRS.EOF Or Browse.EndOfPage)
    Table.BeginRow 2
    Table.Column subjRS("Nome")
    Table.Column subjRS("Endereco")
    Table.Column subjRS("Telefone")
    Table.EndRow

    subjRS.MoveNext
    Browse.IncLine

Loop
Browse.Status = True
Browse.Bar
```

Aplica-se

Browse

Veja também

Método IncLine

Descrição

Incrementa o contador de linhas apresentadas pelo objeto Browse. Esse método deve ser chamado sempre que for avançado um registro.

Sintaxe

objeto.**IncLine**

O argumento *objeto* sempre será o objeto Browse.

Comentários

O código abaixo ilustra o uso da propriedade **ConnectionString**.

```
Browse.Current = "MyBrowse"
Browse.PageSize = 10
Browse.Recordset = sobjRS
Table.BeginTable "400", "Listagem", 3, True
Do Until (sobjRS.EOF Or Browse.EndOfPage)
    Table.BeginRow 2
    Table.Column sobjRS("Nome")
    Table.Column sobjRS("Endereco")
    Table.Column sobjRS("Telefone")
    Table.EndRow
    sobjRS.MoveNext
    Browse.IncLine

Loop
Browse.Status = True
Browse.Bar
```

Aplica-se

Browse

Veja também

Objeto Character

Objeto Character

O objeto *Character* é utilizado para a exibição de texto utilizando imagens para desenho dos caracteres. Esse objeto é particularmente útil para gerar contadores de acesso.

Sintaxe

```
Set Character = Server.CreateObject("Hipertools30.Character")
```

Propriedade Font

Descrição

Especifica qual será a fonte no formato do objeto *Character* que será utilizada para desenha os textos.

Sintaxe

objeto. **Font** [= *String*]

O argumento *objeto* sempre será o objeto *Character*.

Comentários

As fontes utilizadas pelo objeto *Character* são imagens no formato *GIF* ou *JPEG* que obedecem à nomenclatura de *NomeDaFonte* + *CódigoASCII* + *Extensão* (.GIF ou .JPG).

A fontes deverão estar localizadas no diretório especificado pela propriedade *FontDirectory*.

O código abaixo ilustra o uso do método **Font**.

```
Character.Font = "Win"  
Character.ShowButton "Enviar"
```

Aplica-se

Character.

Veja também

FontDirectory, DrawString, DrawButton.

Propriedade FontDirectory

Descrição

Especifica em qual diretório deverão estar armazenadas as imagens das fontes.

Sintaxe

objeto. **FontDirectory** [= *String*]

O argumento *objeto* sempre será o objeto *Character*.

Comentários

O valor padrão para esta propriedade é */Img/GraphChar/*. As fontes ficarão cada uma dentro de um diretório com seu respectivo nome.

O código abaixo ilustra o uso do método **FontDirectory**.

```
Character.FontDirectory = "/Img/GraphChar/"  
Character.ShowButton "Enviar"
```

Aplica-se

Character.

Veja também

Font, DrawString, DrawButton.

Método DrawString

Descrição

Desenha o texto especificado utilizando a fonte definida como caracteres gráficos (imagens).

Sintaxe

objeto.**DrawString** *String*

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Character</i> .
<i>String</i>	Requerido. <i>String</i> é o texto que será desenhado.

Comentários

O código abaixo ilustra o uso do método **DrawString**.

```
Character.Font = "Win"  
Character.DrawString "String a ser Escrita."
```

Aplica-se

Character

Veja também

Font, FontDirectory, DrawButton.

Método DrawButton

Descrição

Desenha um botão utilizando a fonte selecionada.

Sintaxe

objeto.**DrawButton** *String*

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Character</i> .
<i>String</i>	Requerido. <i>String</i> é o texto a ser apresentado no botão.

Comentários

O código abaixo ilustra o uso do método **DrawButton**.

```
Character.Font "Cil"
```

Character

Character.**DrawButton** "Enviar"

Aplica-se

Character.

Veja também

Font, FontDirectory, DrawString.

Objeto Connection

Objeto Connection

O objeto *Connection* visa facilitar o acesso aos banco de dados via ODBC.

Sintaxe

Set *Connection* = Server.CreateObject("ZTITools30.Connection")

Tipo de Conexão		
Constante	Valor	Descrição
conConnSQL	1	Driver={SQL Server}; Server=SQLServer;UID=sa;PWD=; Database=Database
conConnDBase	2	Driver={Microsoft dBASE Driver (*.dbf)}; DBQ=c:\temp; DriverID=2772
conConnAccess	3	Driver={Microsoft Access Driver (*.mdb)}; DBQ=c:\temp\sample.mdb
conConnExcel	4	Driver={Microsoft Excel Driver (*.xls)}; DBQ=c:\temp; DriverID=278 (v3.0/4.0) DBQ=c:\temp\sample.xls; DriverID=22 (v.5.0/7.0) DBQ=c:\temp\sample.xls; DriverID=790 (v. 97)
conConnFoxPro	5	Driver={Microsoft FoxPro Driver (*.dbf)}; DBQ=c:\temp; DriverID=536
conConnText	6	Driver={Microsoft Text Driver (*.txt;*.csv)}; DefaultDir=c:\temp
conConnParadox	7	Driver={Microsoft Paradox Driver (*.db)}; DBQ=c:\temp; DriverID=26

Comandos Disponíveis para o Connection	
Comando	Descrição
DSN	O nome de uma fonte de dados disponível na lista de fontes de dados do ODBC. O comando DSN não é utilizado se o comando DRIVER for especificado.
DRIVER	O nome de um driver disponível na lista de drivers do ODBC. O comando DRIVER não é utilizado se o comando DSN for especificado.
SERVER	O nome de um servidor na rede onde está instalado o SQL-Server. Em um computador com Microsoft Windows NT, "(local)" pode ser utilizado para indicar a própria máquina.

UID	A identificação do usuário na fonte de dados.
PWD	A senha do usuário para acesso à fonte de dados.
APP	(Opcional) O nome da aplicação que está abrindo a conexão.
WSID	(Opcional) A identificação da estação de trabalho que está abrindo a conexão.
DATABASE	(Opcional) O nome do banco de dados SQL Server.
LANGUAGE	(Opcional) O linguagem que deverá ser utilizada.

Tipos de Drives para a sua Conexão		
Constante	Driver ID	Descrição
conConnDriverAccess	25	(Microsoft Access)
conConnDriverdBaseIII	21	(dBASE III)
conConnDriverdBaseIV	277	(dBASE IV)
conConnDriverdBase5	533	(dBASE 5.0)
conConnDriverExcel3	534	(Microsoft Excel 3.0)
conConnDriverExcel4	278	(Microsoft Excel 4.0)
conConnDriverExcel7 conConnDriverExcel5	22	(Microsoft Excel 5.0/7.0)
conConnDriverExcel97	790	(Microsoft Excel 97)
conConnDriverFoxPro20	24	(Microsoft FoxPro 2.0)
conConnDriverFoxPro25	280	(Microsoft FoxPro 2.5)
conConnDriverFoxPro26	536	(Microsoft FoxPro 2.6)
conConnDriverParadox3	26	(Paradox 3.x)
conConnDriverParadox4	282	(Paradox 4.x)
conConnDriverParadox5	538	(Paradox 5.x)
conConnDriverText	27	(Text)

Comentários

O código abaixo ilustra o uso do objeto **Connection**.

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#INCLUDE VIRTUAL="/HiperTools/HiperTools30.inc"-->
<!--#INCLUDE VIRTUAL="/Objects.inc"-->
<%
REM =====
REM /Connection.asp
```

Objeto Connection

```
REM -----
REM  Descricao   : Aplicação de teste do objeto "Connection"
REM  Criação     : 12:00h 10/03/2000
REM  Local      : Brasília/DF
REM  Versao     : 1.0.0
REM -----

Main

REM =====
REM Rotina Principal do Sistema
REM -----
Private Sub Main

MainBody

End Sub
REM -----
REM Final da Sub Main
REM =====

REM =====
REM Corpo Principal do sistema
REM -----
Private Sub MainBody

Default.HTMLBegin
Default.HeadAll "Teste do Objeto Connection"
Default.BodyBegin

Connection.CommandTimeout = 300
Connection.ConnectionTimeout = 300
Connection.ConnectionType = conConnSQL
Connection.Database = "SQLDatabase"
Connection.UserID = "Usuario"
Connection.Password = "Senha"
Connection.ServerAddress = "ServidorSQL"

Show.Message Connection.Connection

Connection.Clear

Connection.ConnectionType = conConnAccess
Connection.DBQ = Initializer.SiteRootDir & _
```

```
                                "Data\Databas.mdb"
Connection.DefaultDir         = "c:\temp"

Show.Message Connection.Connection

Default.PageFooterDefault
Default.BodyEnd
Default.HTMLEnd

End Sub
REM -----
REM Final da Sub MainBody
REM =====

REM -----
REM Fim do Connection.asp
REM =====
%>
```

Propriedade CommandTimeout

Descrição

Especifica o tempo máximo da execução dos comandos durante a conexão com o servidor de Banco de Dados em número de segundos.

Sintaxe

objeto.**CommandTimeout** [= Integer]

O argumento *objeto* sempre será o objeto *Connection*. O valor para **CommandTimeout** é o número de segundos máximo da execução dos comandos no servidor de banco de dados.

Comentários

O código abaixo ilustra o uso da propriedade **CommandTimeout**.

```
Connection.CommandTimeout      = 300
Connection.ConnectionTimeout = 300
Connection.ConnectionType       = conConnSQL
Connection.Database             = "SQLDatabase"
Connection.UserID               = "Usuario"
Connection.Password             = "Senha"
Connection.ServerAddress        = "ServidorSQL"

Show.Message Connection.ConnectionString
Retorna: "DRIVER={SQL Server};SERVER=ServidorSQL;UID=Usuario;PWD=Senha
;DATABASE=SQLDatabase;"
```

Aplica-se

Connection

Veja também

ConnectionTimeout, ConnectionString

Propriedade ConnectionTimeout

Descrição

Especifica tempo máximo da conexão com o servidor de Banco de Dados em segundos.

Sintaxe

objeto.**ConnectionTimeout** [= Integer]

O argumento *objeto* sempre será o objeto *Connection*. O valor para **ConnectionTimeout** é o número de segundos que será o tempo máximo da conexão.

Comentários

O código abaixo ilustra o uso da propriedade **ConnectionTimeout**.

```
Connection.CommandTimeout      = 300
Connection.ConnectionTimeout = 300
Connection.ConnectionType      = conConnSQL
Connection.Database            = "SQLDatabase"
Connection.UserID              = "Usuario"
Connection.Password            = "Senha"
Connection.ServerAddress       = "ServidorSQL"

Show.Message Connection.ConnectionString
Retorna: "DRIVER={SQL Server};SERVER=ServidorSQL;UID=Usuario;PWD=Senha
;DATABASE=SQLDatabase;"
```

Aplica-se

Connection

Veja também

CommandTimeout, ConnectionString

Propriedade ConnectionType

Descrição

Especifica o tipo de conexão que será utilizada para acessar o banco de dados.

Sintaxe

objeto.**ConnectionType** [= String]

O argumento *objeto* sempre será o objeto *Connection*. O valor para **ConnectionType** com base nas constantes conConn...

Comentários

O código abaixo ilustra o uso da propriedade **ConnectionType**.

```
Connection.CommandTimeout      = 300
Connection.ConnectionTimeout    = 300
Connection.ConnectionType     = conConnSQL
Connection.Database            = "SQLDatabase"
```

```
Connection.UserID          = "Usuario"
Connection.Password        = "Senha"
Connection.ServerAddress   = "ServidorSQL"
```

```
Show.Message Connection.ConnectionString
```

```
Retorna: "DRIVER={SQL Server};SERVER=ServidorSQL;UID=Usuario;PWD=Senha
;DATABASE=SQLDatabase;"
```

Aplica-se

Connection

Veja também

Database, DBQ, ServerAddress

Propriedade Database

Descrição

Especifica o nome do banco de dados (Database) que será utilizado pelo ODBC.

Sintaxe

objeto.**Database** [= *String*]

O argumento *objeto* sempre será o objeto *Connection*. O valor para **Database** será um database válido do servidor de SQL.

Comentários

O código abaixo ilustra o uso da propriedade **Database**.

```
Connection.CommandTimeout  = 300
Connection.ConnectionTimeout = 300
Connection.ConnectionType   = conConnSQL
Connection.Database        = "SQLDatabase"
Connection.UserID           = "Usuario"
Connection.Password         = "Senha"
Connection.ServerAddress    = "ServidorSQL"
```

```
Show.Message Connection.ConnectionString
```

```
Retorna: "DRIVER={SQL Server};SERVER=ServidorSQL;UID=Usuario;PWD=Senha
;DATABASE=SQLDatabase;"
```

Aplica-se

Connection

Veja também

ConnectionType, DBQ, ServerAddress

Propriedade DBQ

Descrição

Especifica o diretório onde está o arquivo que será utilizado pelo ODBC.

Sintaxe

objeto.**DBQ** [= *String*]

O argumento *objeto* sempre será o objeto *Connection*. O valor para **DBQ** deverá ser um diretório válido com o nome de arquivo de acordo com o driver utilizado.

Comentários

O código abaixo ilustra o uso da propriedade **DBQ**.

```
Connection.ConnectionType      = conConnAccess
Connection.DBQ                 = Initializer.SiteRootDir & "Data\Databas.mdb"
Connection.DefaultDir          = "c:\temp"

Show.Message Connection.Connection
Retorna: "DRIVER={Microsoft Access Driver (*.mdb)};" _
        "DBQ=d:\inetpub\wwwroot\Data\Databas.mdb;DefaultDir=c:\temp;"
```

Aplica-se

Connection

Veja também

ConnectionType, Database, ServerAddress

Propriedade DefaultDir

Descrição

Especifica o diretório padrão onde serão manipulados os arquivos pelo ODBC.

Sintaxe

objeto.**DefaultDir** [= *String*]

O argumento *objeto* sempre será o objeto *Connection*. O valor para **DefaultDir** é o diretório completo que será utilizado pelo ODBC.

Comentários

O código abaixo ilustra o uso da propriedade **DefaultDir**.

```
Connection.CommandTimeout      = 300
Connection.ConnectionTimeout    = 300
Connection.ConnectionType       = conConnSQL
Connection.Database             = "SQLDatabase"
Connection.UserID               = "Usuario"
Connection.Password             = "Senha"
Connection.ServerAddress        = "ServidorSQL"
Connection.DefaultDir           = "C:\Temp"

Show.Message Connection.ConnectionString
Retorna: "DRIVER={SQL Server};SERVER=ServidorSQL;UID=Usuario;PWD=Senha
;DATABASE=SQLDatabase;"
```


Aplica-se

Connection

Veja também

Propriedade Password

Descrição

Especifica a senha que será utilizada para validar o usuário no banco de dados.

Sintaxe

objeto.**Password** [= *String*]

O argumento *objeto* sempre será o objeto *Connection*. O valor para **Password** é a senha do usuário válido do banco de dados.

Comentários

O código abaixo ilustra o uso da propriedade **Password**.

```
Connection.CommandTimeout      = 300
Connection.ConnectionTimeout    = 300
Connection.ConnectionType       = conConnSQL
Connection.Database             = "SQLDatabase"
Connection.UserID               = "Usuario"
Connection.Password             = "Senha"
Connection.ServerAddress        = "ServidorSQL"

Show.Message Connection.ConnectionString
Retorna: "DRIVER={SQL Server};SERVER=ServidorSQL;UID=Usuario;PWD=Senha
;DATABASE=SQLDatabase;"
```

Aplica-se

Connection

Veja também

UserID

Propriedade ServerAddress

Descrição

Especifica o endereço do servidor.

Sintaxe

objeto.**ServerAddress** [= *String*]

O argumento *objeto* sempre será o objeto *Connection*. O valor para **ServerAddress** ser o endereço do servidor de banco de dados utilizado pelo ODBC.

Comentários

O código abaixo ilustra o uso da propriedade **ServerAddress**.

```
Connection.CommandTimeout      = 300
Connection.ConnectionTimeout    = 300
Connection.ConnectionType       = conConnSQL
Connection.Database             = "SQLDatabase"
Connection.UserID               = "Usuario"
Connection.Password             = "Senha"
Connection.ServerAddress       = "ServidorSQL"
```

```
Show.Message Connection.ConnectionString
```

```
Retorna: "DRIVER={SQL Server};SERVER=ServidorSQL;UID=Usuario;PWD=Senha
;DATABASE=SQLDatabase;"
```

Aplica-se

Connection

Veja também

ConnexionType, Database, DBQ

Propriedade UserID

Descrição

Especifica o usuário que será utilizado para validar o usuário no banco de dados.

Sintaxe

objeto.**UserID** [= *String*]

O argumento *objeto* sempre será o objeto *Connection*. O valor para **UserID** é um usuário válido do banco de dados.

Comentários

O código abaixo ilustra o uso da propriedade **UserID**.

```
Connection.CommandTimeout      = 300
Connection.ConnectionTimeout    = 300
Connection.ConnectionType       = conConnSQL
Connection.Database             = "SQLDatabase"
Connection.UserID              = "Usuario"
Connection.Password             = "Senha"
Connection.ServerAddress        = "ServidorSQL"
```

```
Show.Message Connection.ConnectionString
```

```
Retorna: "DRIVER={SQL Server};SERVER=ServidorSQL;UID=Usuario;PWD=Senha
;DATABASE=SQLDatabase;"
```

Aplica-se

Connection

Veja também

Password

Método Clear

Descrição

Apagar todas as propriedades do objeto *Connection*.

Sintaxe

objeto.Clear

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Connection</i> .

Comentários

O código abaixo ilustra o uso do método **Clear**.

```
Connection.CommandTimeout      = 300
Connection.ConnectionTimeout    = 300
Connection.ConnectionType      = conConnSQL
Connection.Database            = "SQLDatabase"
Connection.UserID              = "Usuario"
Connection.Password            = "Senha"
Connection.ServerAddress       = "ServidorSQL"
```

```
Show.Message Connection.ConnectionString
```

```
Retorna: "DRIVER={SQL Server};SERVER=ServidorSQL;UID=Usuario;PWD=Senha
;DATABASE=SQLDatabase;"
```

```
Connection.CommandTimeout      = 300
```

```
Connection.Clear
```

```
Connection.ConnectionType      = conConnAccess
Connection.DBQ                  = Initializer.SiteRootDir & "Data\Databas.mdb"
Connection.DefaultDir           = "c:\temp"
```

```
Show.Message Connection.Connection
```

```
Retorna: "DRIVER={Microsoft Access Driver (*.mdb)};" _
"DBQ=d:\inetpub\wwwroot\Data\Databas.mdb;DefaultDir=c:\temp;"
```

Aplica-se

Connection

Vea também

ConnectionString

Método ConnectionString

Descrição

Retorna a string que será utilizada para conectar ao banco de dados.

Sintaxe

objeto.**ConnectionString**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Connection</i> .

Comentários

O código abaixo ilustra o uso do método **ConnectionString**.

```
Connection.CommandTimeout      = 300
Connection.ConnectionTimeout    = 300
Connection.ConnectionType       = conConnSQL
Connection.Database             = "SQLDatabase"
Connection.UserID               = "Usuario"
Connection.Password             = "Senha"
Connection.ServerAddress        = "ServidorSQL"
```

```
Show.Message Connection.ConnectionString
```

```
Retorna: "DRIVER={SQL Server};SERVER=ServidorSQL;UID=Usuario;PWD=Senha
;DATABASE=SQLDatabase;"
```

Aplica-se

Connection

Veja também

Clear

Objeto Default

Objeto Default

Esse objeto é responsável pela escrita do código dos cabeçalhos, corpo dos documentos e outras definições padrão.

Sintaxe

```
Set Default = Server.CreateObject("HiperTools30.Default")
```

Propriedade BaseFont

Descrição

Especifica a fonte base a ser utilizada no corpo do documento.

Sintaxe

```
objeto.BaseFont [= String]
```

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **BaseFont**.

```
Default.BaseFont = "Verdana"
```

Aplica-se

Default

Veja também

Propriedade BodyALink

Descrição

Especifica a cor dos links ao serem clicados a ser utilizada no corpo do documento.

Sintaxe

```
objeto.BodyALink [= String]
```

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

Os valores válidos para essa propriedade são os mesmos do HTML, especificado como nome da cor ou valor RGB em hexadecimal.

O código abaixo ilustra o uso da propriedade **BodyALink**.

```
Default.BodyALink = "#FFFF00"
```

Aplica-se

Default

Veja também

Propriedade BodyBackground

Descrição

Especifica a imagem de fundo a ser utilizada no corpo do documento.

Sintaxe

objeto.**BodyBackground** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **BodyBackground**.

```
Default.BodyBackground = "tijolos.gif"
```

Aplica-se

Default

Veja também

Propriedade BodyBGColor

Descrição

Especifica a cor de fundo a ser utilizada no corpo do documento.

Sintaxe

objeto.**BodyBGColor** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

Os valores válidos para essa propriedade são os mesmos do HTML, especificado como nome da cor ou valor RGB em hexadecimal.

O código abaixo ilustra o uso da propriedade **BodyBGColor**.

```
Default.BodyBGColor = "black"
```

Aplica-se

Default

Veja também

Propriedade BodyBGProperties

Descrição

Especifica as propriedades da imagem de fundo a serem utilizada no corpo do documento.

Sintaxe

objeto.**BodyBGProperties** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **BodyBGProperties**.

```
Default.BodyBGProperties = "fixed"
```

Aplica-se

Default

Veja também

Propriedade BodyLeftMargin

Descrição

Especifica o espaço, em pixels, entre o lado esquerdo da página e o conteúdo do documento.

Sintaxe

objeto.**BodyLeftMargin** [= *Integer*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **BodyLeftMargin**.

```
Default.BodyLeftMargin = 0
```

Aplica-se

Default

Veja também

Propriedade BodyLink

Descrição

Especifica a cor dos links a ser utilizada no corpo do documento.

Sintaxe

objeto.**BodyLink** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

Os valores válidos para essa propriedade são os mesmos do HTML, especificado como nome da cor ou valor RGB em hexadecimal.

O código abaixo ilustra o uso da propriedade **BodyLink**.

```
Default.BodyLink = "crimson"
```

Aplica-se

Default

Veja também

Propriedade BodyOnLoad

Descrição

Especifica as instruções em JavaScript a serem executadas após a carga da página.

Sintaxe

objeto.**BodyOnLoad** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **BodyOnLoad**.

```
Default.BodyOnLoad = "alert('carregada')"
```

Aplica-se

Default

Veja também

Propriedade BodyOnUnload

Descrição

Especifica as instruções em JavaScript a serem executadas ao sair da página.

Sintaxe

objeto.**BodyOnUnload** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **BodyOnUnload**.

```
Default.BodyOnUnload = "alert('descarregada')"
```

Aplica-se

Default

Veja também

Propriedade BodyStyle

Descrição

Especifica o estilo a ser utilizado no corpo do documento.

Sintaxe

objeto.**BodyStyle** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **BodyStyle**.

```
Default.BodyStyle = "text-align: right"
```

Aplica-se

Default

Veja também

Propriedade BodyText

Descrição

Especifica a cor do texto a ser utilizada no corpo do documento.

Sintaxe

objeto.**BodyText** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

Os valores válidos para essa propriedade são os mesmos do HTML, especificado como nome da cor ou valor RGB em hexadecimal.

O código abaixo ilustra o uso da propriedade **BodyText**.

```
Default.BodyText = "white"
```

Aplica-se

Default

Veja também

Propriedade BodyTopMargin

Descrição

Especifica o espaço, em pixels, entre o lado superior da página e o conteúdo do documento.

Sintaxe

objeto.**BodyTopMargin** [= *Integer*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **BodyTopMargin**.

```
Default.BodyTopMargin = 0
```

Aplica-se

Default

Veja também

Propriedade BodyVLink

Descrição

Especifica a cor dos links já visitados a ser utilizada no corpo do documento.

Sintaxe

objeto.**BodyVLink** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

Os valores válidos para essa propriedade são os mesmos do HTML, especificado como nome da cor ou valor RGB em hexadecimal.

O código abaixo ilustra o uso da propriedade **BodyVLink**.

```
Default.BodyVLink = "#00FFFF"
```

Aplica-se

Default

Veja também

Propriedade BodyWidth

Descrição

Define se o conteúdo do documento vai estar dentro de uma tabela com largura delimitada em pixels. Essa propriedade é útil no caso de um layout desenvolvido para uma determinada resolução.

Sintaxe

objeto.**BodyWidth** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **BodyWidth**.

```
Default.BeginHTML
Default.BeginHead "Suporte de Informática"
Default.WriteMeta
Default.EndHead
Default.WriteBaseFont
Default.WriteLink

Default.BodyWidth = "795"
```

```
' ou
Default.BodyWidth = "80%"

Default.BeginBody
```

Aplica-se
Default

Veja também

Propriedade CenterBody

Descrição

Define se o conteúdo do documento será centralizado. Se utilizado em conjunto com a propriedade *BodyWidth*, faz com que a área central da janela seja usada para a exibição do documento, dentro da largura especificada.

Sintaxe

objeto.**CenterBody** [= *Boolean*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **CenterBody**.

```
Default.BeginHTML
Default.BeginHead "Suporte de Informática"
Default.WriteMeta
Default.EndHead
Default.WriteBaseFont
Default.WriteLink

Default.CenterBody = True
Default.BodyWidth = 635
Default.BeginBody
```

Aplica-se
Default

Veja também

Propriedade FooterCopyright

Descrição

Especifica a nota de *copyright* do rodapé padrão.

Sintaxe

objeto.**FooterCopyright** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **FooterCopyright**.

```
Show.HTML Default.FooterCopyRight
```

Aplica-se

Default

Veja também

PageFooterDefault

Propriedade FooterSugestions

Descrição

Especifica a nota de sugestões do rodapé padrão.

Sintaxe

objeto.**FooterSugestions** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **FooterSugestions**.

```
Show.HTML Default.FooterSugestions
```

Aplica-se

Default

Veja também

PageFooterDefault

Propriedade FooterTitle

Descrição

Especifica a nota de título do rodapé padrão.

Sintaxe

objeto.**FooterTitle** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **FooterTitle**.

```
Show.HTML Default.FooterTitle
```

Aplica-se

Default

Veja também

PageFooterDefault

Propriedade HeaderTitle

Descrição

XxXxXx.

Sintaxe

objeto.HeaderTitle [= String]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **HeaderTitle**.

Exemplo

Aplica-se

Default

Veja também

Propriedade IgnoreBodyWidth

Descrição

Define se a largura definida na propriedade *BodyWidth* deve ser ignorada.

Sintaxe

objeto.IgnoreBodyWidth [= Boolean]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **IgnoreBodyWidth**.

```
Default.BeginHTML
Default.BeginHead "Suporte de Informática"
Default.WriteMeta
Default.EndHead
Default.BodyWidth = "775"
Default.IgnoreBodyWidth
Default.BeginBody
```

Aplica-se

Default

Veja também

Propriedade LinkCopyrightHref

Descrição

Define se a largura definida na propriedade *BodyWidth* deve ser ignorada.

Sintaxe

objeto.IgnoreBodyWidth [= *Boolean*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **IgnoreBodyWidth**.

```
Default.BeginHTML
Default.BeginHead "Suporte de Informática"
Default.WriteMeta
Default.EndHead
Default.BodyWidth = "775"
Default.IgnoreBodyWidth
Default.BeginBody
```

Aplica-se

Default

Veja também

Propriedade LinkCopyrightTitle

Descrição

XxXxXx.

Sintaxe

objeto.LinkCopyrightTitle [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

As propriedades que iniciam com link indicam o relacionamento da página gerada com outros documentos.

O código abaixo ilustra o uso da propriedade **LinkCopyrightTitle**.

Exemplo

Aplica-se

Default

Veja também

Propriedade LinkHomeHref

Descrição

XxXxXx.

Sintaxe

objeto.**LinkHomeHref** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

As propriedades que iniciam com link indicam o relacionamento da página gerada com outros documentos.

O código abaixo ilustra o uso da propriedade **LinkHomeHref**.

Exemplo

Aplica-se

Default

Veja também

Propriedade LinkHomeTitle

Descrição

XxXxXx.

Sintaxe

objeto.**LinkHomeTitle** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

As propriedades que iniciam com link indicam o relacionamento da página gerada com outros documentos.

O código abaixo ilustra o uso da propriedade **LinkHomeTitle**.

Exemplo

Aplica-se

Default

Veja também

Propriedade LinkMadeHref

Descrição

XxXxXx.

Sintaxe

objeto.**LinkMadeHRef** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

As propriedades que iniciam com link indicam o relacionamento da página gerada com outros documentos.

O código abaixo ilustra o uso da propriedade **LinkMadeHRef**.

Exemplo

Aplica-se

Default

Veja também

Propriedade LinkMadeTitle

Descrição

XxXxXx.

Sintaxe

objeto.**LinkMadeTitle** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

As propriedades que iniciam com link indicam o relacionamento da página gerada com outros documentos.

O código abaixo ilustra o uso da propriedade **LinkMadeTitle**.

Exemplo

Aplica-se

Default

Veja também

Propriedade LinkStyleSheetHRef

Descrição

XxXxXx.

Sintaxe

objeto.**LinkStyleSheetHRef** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

As propriedades que iniciam com link indicam o relacionamento da página gerada com outros documentos.

O código abaixo ilustra o uso da propriedade **LinkStyleSheetHref**.

Exemplo

Aplica-se

Default

Veja também

Propriedade MetaAuthor

Descrição

XxXxXx.

Sintaxe

objeto.**MetaAuthor** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **MetaAuthor**.

Exemplo

Aplica-se

Default

Veja também

Propriedade MetaCategory

Descrição

XxXxXx.

Sintaxe

objeto.**MetaCategory** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **MetaCategory**.

Exemplo

Aplica-se
Default

Veja também

Propriedade MetaCharSet

Descrição

XxXxXx.

Sintaxe

objeto.**MetaCharSet** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **MetaCharSet**.

Exemplo

Aplica-se
Default

Veja também

Propriedade MetaContentType

Descrição

XxXxXx.

Sintaxe

objeto.**MetaContentType** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **MetaContentType**.

Exemplo

Aplica-se
Default

Veja também

Propriedade MetaDescription

Descrição

XxXxXx.

Sintaxe

objeto.**MetaDescription** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **MetaDescription**.

Exemplo

Aplica-se

Default

Veja também

Propriedade MetaDistribution

Descrição

XxXxXx.

Sintaxe

objeto.**MetaDistribution** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **MetaDistribution**.

Exemplo

Aplica-se

Default

Veja também

Propriedade MetaExpires

Descrição

XxXxXx.

Sintaxe

objeto.**MetaExpires** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **MetaExpires**.

Exemplo

Aplica-se
Default

Veja também

Propriedade MetaFormatter

Descrição

XxXxXx.

Sintaxe

objeto.**MetaFormatter** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **MetaFormatter**.

Exemplo

Aplica-se
Default

Veja também

Propriedade MetaGenerator

Descrição

XxXxXx.

Sintaxe

objeto.**MetaGenerator** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **MetaGenerator**.

Exemplo

Aplica-se
Default

Veja também

Propriedade MetaKeyPhrase

Descrição

XxXxXx.

Sintaxe

objeto.**MetaKeyPhrase** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **MetaKeyPhrase**.

Exemplo

Aplica-se

Default

Veja também

Propriedade MetaKeywords

Descrição

XxXxXx.

Sintaxe

objeto.**MetaKeywords** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **MetaKeywords**.

Exemplo

Aplica-se

Default

Veja também

Propriedade MetaLocale

Descrição

XxXxXx.

Sintaxe

objeto.**MetaLocale** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **MetaLocale**.

Exemplo

Aplica-se
Default

Veja também

Propriedade MetaPicsLabel

Descrição

XxXxXx.

Sintaxe

objeto.**MetaPicsLabel** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **MetaPicsLabel**.

Exemplo

Aplica-se
Default

Veja também

Propriedade MetaPragma

Descrição

XxXxXx.

Sintaxe

objeto.**MetaPragma** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **MetaPragma**.

Exemplo

Aplica-se
Default

Veja também

Propriedade MetaProduct

Descrição

XxXxXx.

Sintaxe

objeto.**MetaProduct** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **MetaProduct**.

Exemplo

Aplica-se

Default

Veja também

Propriedade MetaRefresh

Descrição

XxXxXx.

Sintaxe

objeto.**MetaRefresh** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **MetaRefresh**.

Exemplo

Aplica-se

Default

Veja também

Propriedade MetaReplyTo

Descrição

XxXxXx.

Sintaxe

objeto.**MetaReplyTo** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **MetaReplyTo**.

Exemplo

Aplica-se
Default

Veja também

Propriedade MetaTechnology

Descrição

XxXxXx.

Sintaxe

objeto.**MetaTechnology** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **MetaTechnology**.

Exemplo

Aplica-se
Default

Veja também

Propriedade PageFooter

Descrição

Especifica o cabeçalho da página. O cabeçalho, por padrão, é exibido centralizado.

Sintaxe

objeto.**PageFooter** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **PageFooter**.

```
Default.PageFooter = "<HR>Webmaster: webmaster@hipertools.com.br"
```

Aplica-se
Default

Veja também
PageFooterDefault , WritePageFooter

Propriedade PageHeader

Descrição

Especifica o cabeçalho da página. O cabeçalho, por padrão, é exibido centralizado.

Sintaxe

objeto.**PageHeader** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **PageHeader**.

```
Default.PageHeader = "<H1>HiperTools' Site</H1>"
```

Aplica-se

Default

Veja também

WritePageHeader

Propriedade SiteCounter

Descrição

XxXxXx.

Sintaxe

objeto.**SiteCounter** [= *Long*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **SiteCounter**.

```
Exemplo
```

Aplica-se

Default

Veja também

Propriedade SourceCode

Descrição

XxXxXx.

Sintaxe

objeto.**SourceCode** [= *Boolean*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **SourceCode**.

```
Exemplo
```

Aplica-se
Default

Veja também

Propriedade TableBorder

Descrição

XxXxXx.

Sintaxe

objeto.**TableBorder** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **TableBorder**.

Exemplo

Aplica-se
Default

Veja também

Propriedade UserID

Descrição

XxXxXx.

Sintaxe

objeto.**UserID** [= *String*]

O argumento *objeto* sempre terá que ser um objeto *Default*.

Comentários

O código abaixo ilustra o uso da propriedade **UserID**.

Exemplo

Aplica-se
Default

Veja também

Método BeginBody

Descrição

Inicia o corpo do documento.

Sintaxe

objeto.**BeginBody**

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Default</i> .

Comentários

O código abaixo ilustra o uso do método **BeginBody**.

```
Default.BeginHTML
Default.BeginHead "Suporte de Informática"
Default.WriteMeta
Default.EndHead
Default.WriteBaseFont
Default.WriteLink
Default.BeginBody
Retorna: <BODY>

Show.Center

Table.BeginTable "65%"
Table.Style.Color2 = ""
Table.BeginRow 4

Table.ColumnVAlign = "Middle"
Table.Column "<img src=""/img/Zevallos.gif"">"
Table.Column "<Font Color=""DarkBlue"">Suporte de Informática"

Table.EndRow
Table.EndTable

Default.PageFooterDefault
Default.EndBody
Default.EndHTML
```

Aplica-se

Default

Veja também

Método BeginHead

Descrição

Inicia o cabeçalho do documento.

Sintaxe*objeto.BeginHead String*

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Default</i> .
<i>String</i>	Requerido. Título da página.

Comentários

O código abaixo ilustra o uso do método **BeginHead**.

```
Default.BeginHTML
Default.BeginHead "Suporte de Informática"
Retorna: <HEAD><TITLE> Suporte de Informática</TITLE>

Default.WriteMeta
Default.EndHead
Default.WriteBaseFont
Default.WriteLink
Default.BeginBody
Show.Center

Table.BeginTable "65%"
Table.Style.Color2 = " "
Table.BeginRow 4

Table.ColumnVAlign = "Middle"
Table.Column "<img src=""/img/Zevallos.gif"">"
Table.Column "<Font Color=""DarkBlue"">Suporte de Informática"

Table.EndRow
Table.EndTable

Default.PageFooterDefault
Default.EndBody
Default.EndHTML
```

Aplica-se

Default

Veja também

Método BeginHTML

Descrição

Inicia o código HTML.

Sintaxe

objeto.**BeginHTML**

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Default</i> .

Comentários

O código abaixo ilustra o uso do método **BeginHTML**.

```
Default.BeginHTML
Retorna: <HTML>
Default.WriteHead "Suporte de Informática"
Default.BeginBody
Show.Center

Table.BeginTable "65%"
Table.Style.Color2 = ""
Table.BeginRow 4

Table.ColumnVAlign = "Middle"
Table.Column "<img src=""/img/Zevallos.gif"">"
Table.Column "<Font Color=""DarkBlue"">Suporte de Informática"

Table.EndRow
Table.EndTable

Default.PageFooterDefault
Default.EndBody
Default.EndHTML
```

Aplica-se

Default

Veja também

Método EndBody

Descrição

Finaliza o corpo do documento.

Sintaxe

objeto.**EndBody**

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Default</i> .

Comentários

O código abaixo ilustra o uso do método **EndBody**.

```
Default.BeginHTML
Default.BeginHead "Suporte de Informática"
Default.WriteMeta
Default.EndHead
Default.WriteBaseFont
Default.WriteLink

Default.IgnoreBodyWidth = True
Default.CenterBody = True
Default.BeginBody

Show.Center

Table.BeginTable "65%"
Table.Style.Color2 = ""
Table.BeginRow 4

Table.ColumnVAlign = "Middle"
Table.Column "<img src=\"/img/Zevallos.gif\">"
Table.Column "<Font Color=\"DarkBlue\">Suporte de Informática"

Table.EndRow
Table.EndTable

Default.PageFooterDefault
Default.EndBody
Retorna: </TD></TR><TABLE></CENTER></BODY>

Default.EndHTML
```

Aplica-se

Default

Veja também

Método EndHead

Descrição

Finaliza o cabeçalho do documento.

Sintaxe

objeto.**EndHead**

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Default</i> .

Comentários

O código abaixo ilustra o uso do método **EndHead**.

```
Default.BeginHTML
Default.BeginHead "Suporte de Informática"
Default.WriteMeta
Default.EndHead
Retorna: </HEAD>

Default.WriteBaseFont
Default.WriteLink
Default.BeginBody
Show.Center

Table.BeginTable "65%"
Table.Style.Color2 = ""
Table.BeginRow 4

Table.ColumnVAlign = "Middle"
Table.Column "<img src=""/img/Zevallos.gif"">"
Table.Column "<Font Color=""DarkBlue"">Suporte de Informática"

Table.EndRow
Table.EndTable

Default.PageFooterDefault
Default.EndBody
Default.EndHTML
```

Aplica-se

Default

Veja também

Método EndHTML

Descrição

Finaliza o código HTML.

Sintaxe

objeto.**EndHTML**

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Default</i> .

Comentários

O código abaixo ilustra o uso do método **EndHTML**.

```
Default.BeginHTML
Default.WriteHead "Suporte de Informática"
Default.BeginBody
Show.Center

Table.BeginTable "65%"
Table.Style.Color2 = ""
Table.BeginRow 4

Table.ColumnVAlign = "Middle"
Table.Column "<img src=\"/img/Zevallos.gif\">"
Table.Column "<Font Color=\"DarkBlue\">Suporte de Informática"

Table.EndRow
Table.EndTable

Default.PageFooterDefault
Default.EndBody
Default.EndHTML
Retorna: </HTML>
```

Aplica-se

Default

Veja também

Método PageFooterDefault

Descrição

Exibe o rodapé padrão.

Sintaxe

objeto.**PageFooterDefault**

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Default</i> .

Comentários

O rodapé padrão consiste em uma linha de título, uma linha de sugestões e uma linha de *copyright*, definidas respectivamente nas propriedades *FooterTitle*, *FooterSugestions* e *FooterCopyright*.

O código abaixo ilustra o uso do método **PageFooterDefault**.

```
Default.BeginHTML
Default.WriteHead "Suporte de Informática"
Default.BeginBody
Show.Center

Table.BeginTable "65%"
Table.Style.Color2 = ""
Table.BeginRow 4

Table.ColumnVAlign = "Middle"
Table.Column "<img src=""/img/Zevallos.gif"">"
Table.Column "<Font Color=""DarkBlue"">Suporte de Informática"

Table.EndRow
Table.EndTable

Default.PageFooterDefault

Retorna:
Default.EndBody
Default.EndHTML
```

Aplica-se

Default

Veja também

Método WriteBaseFont

Descrição

Usa a fonte definida na propriedade *BaseFont* como a fonte padrão para o documento atual.

Sintaxe

objeto.**WriteBaseFont**

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Default</i> .

Comentários

O código abaixo ilustra o uso do método **WriteBaseFont**.

```
Default.BeginHTML
Default.BeginHead "Suporte de Informática"
Default.WriteMeta
Default.EndHead
Default.WriteBaseFont
Default.WriteLink

Default.IgnoreBodyWidth = True
Default.CenterBody = True
Default.BeginBody

Show.Center

Table.BeginTable "65%"
Table.Style.Color2 = ""
Table.BeginRow 4

Table.ColumnVAlign = "Middle"
Table.Column "<img src=\"/img/Zevallos.gif\">"
Table.Column "<Font Color=\"DarkBlue\">Suporte de Informática"

Table.EndRow
Table.EndTable

Default.PageFooterDefault
Default.EndBody

Default.EndHTML
```

Aplica-se

Default

Veja também

Método WriteHead

Descrição

Cria um cabeçalho com todos os valores padrão (equivalentes aos métodos *BeginHead*, *WriteBaseFont*, *WriteLink*, *WriteMeta* e *EndHead*).

Sintaxe

objeto.**WriteHead** *String*

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Default</i> .
<i>String</i>	Requerido. Título da página.

Comentários

O código abaixo ilustra o uso do método **WriteHead**.

```
Default.BeginHTML
Default.WriteHead "Suporte de Informática"
Default.BeginBody
Show.Center

Table.BeginTable "65%"
Table.Style.Color2 = ""
Table.BeginRow 4

Table.ColumnVAlign = "Middle"
Table.Column "<img src=""/img/Zevallos.gif"">"
Table.Column "<Font Color=""DarkBlue"">Suporte de Informática"

Table.EndRow
Table.EndTable

Default.PageFooterDefault
Default.EndBody
Default.EndHTML
```

Aplica-se

Default

Veja também

Método WriteLink

Descrição

Inclui no documento as *tags* LINK definidas nas propriedades *LinkCopyrightHRef*, *LinkCopyrightTitle*, *LinkHomeHRef*, *LinkHomeTitle*, *LinkMadeHRef*, *LinkMadeTitle* e *LinkStyleSheetHRef*.

Sintaxe*objeto*.WriteLink

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Default</i> .

Comentários

O código abaixo ilustra o uso do método **WriteLink**.

```
Default.BeginHTML
Default.BeginHead "Suporte de Informática"
Default.WriteMeta
Default.EndHead
Default.WriteBaseFont
Default.WriteLink

Default.IgnoreBodyWidth = True
Default.CenterBody = True
Default.BeginBody

Show.Center

Table.BeginTable "65%"
Table.Style.Color2 = ""
Table.BeginRow 4

Table.ColumnVAlign = "Middle"
Table.Column "<img src=""/img/Zevallos.gif"">"
Table.Column "<Font Color=""DarkBlue"">Suporte de Informática"

Table.EndRow
Table.EndTable

Default.PageFooterDefault
Default.EndBody

Default.EndHTML
```

Aplica-se

Default

Veja também

Método WriteMeta

Descrição

Inclui no documento as *tags* LINK definidas nas propriedades *MetaAuthor*, *MetaCategory*, *MetaCharSet*, *MetaContentType*, *MetaDescription*, *MetaDistribution*, *MetaExpires*, *MetaFormatter*, *MetaGenerator*, *MetaKeyPhrase*, *MetaKeywords*, *MetaLocale*, *MetaPicsLabel*, *MetaPragma*, *MetaProduct*, *MetaRefresh*, *MetaReplyTo* e *MetaTechnology*.

Sintaxe

objeto.**WriteMeta**

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Default</i> .

Comentários

O código abaixo ilustra o uso do método **WriteMeta**.

```
Default.BeginHTML
Default.BeginHead "Suporte de Informática"
Default.WriteMeta
Default.EndHead
Default.WriteBaseFont
Default.WriteLink

Default.IgnoreBodyWidth = True
Default.CenterBody = True
Default.BeginBody

Show.Center

Table.BeginTable "65%"
Table.Style.Color2 = ""
Table.BeginRow 4

Table.ColumnVAlign = "Middle"
Table.Column "<img src=""/img/Zevallos.gif"">"
Table.Column "<Font Color=""DarkBlue"">Suporte de Informática"

Table.EndRow
Table.EndTable

Default.PageFooterDefault
Default.EndBody

Default.EndHTML
```

Aplica-se
Default

Veja também

Método WritePageFooter

Descrição

Exibe o rodapé das páginas, definido na propriedade *PageFooter*.

Sintaxe

objeto.**WritePageFooter**

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Default</i> .

Comentários

O código abaixo ilustra o uso do método **WritePageFooter**.

```
Default.BeginHTML
Default.WriteHead "Suporte de Informática"
Default.BeginBody
Show.Center

Table.BeginTable "65%"
Table.Style.Color2 = ""
Table.BeginRow 4

Table.ColumnVAlign = "Middle"
Table.Column "<img src="" /img/Zevallos.gif"">"
Table.Column "<Font Color=""DarkBlue"">Suporte de Informática"

Table.EndRow
Table.EndTable

Default.PageFooterDefault

Retorna:
Default.EndBody
Default.EndHTML
```

Aplica-se
Default

Veja também

Método WritePageHeader

Descrição

Exibe o cabeçalho das páginas, definido na propriedade *PageHeader*.

Sintaxe

objeto.**WritePageHeader** *String*

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Default</i> .

Comentários

O código abaixo ilustra o uso do método **WritePageHeader**.

```
Default.BeginHTML
Default.BeginBody
Default.PageHeader = "<H1>HiperTools' Site</H1>"
Default.WritePageHead
Show.Center

Table.BeginTable "65%"
Table.Style.Color2 = ""
Table.BeginRow 4

Table.ColumnVAlign = "Middle"
Table.Column "<img src=""/img/Zevallos.gif"">"
Table.Column "<Font Color=""DarkBlue"">Suporte de Informática"

Table.EndRow
Table.EndTable

Default.PageFooterDefault
Default.EndBody
Default.EndHTML
```

Aplica-se

Default

Veja também

Objeto EditForm

Objeto EditForm

Descrição

Objeto que acelera o desenvolvimento formulários de edição de dados. Cria formulários, listagens, localizações de registros, relatórios de resumos, com pouca mão-de-obra, mas com recursos avançados.

Via este objeto, você poderá criar verdadeiros formulários para manipulação de dados diretamente via a Interface Web sem as preocupações da interação com o sistema, servidor Web e servidor de banco de dados.

O sistema inclui um mini dicionário de dados que facilitar a criação das tabelas e manutenção dos dados, com segurança e integridade relacional.

O EditForm também dispõe de um sistema completo para localização/seleção de registros integrados entre si, bem como com a possibilidade de vários tipos de relatórios gráficos ou não simplificados via uma macro linguagem definidos em uma só linha.

Boa parte das definições do EditForm são definidas em uma linha, como o Edit.FormUnit e Edit.FormFind que formatam a forma de apresentação dos dados e os campos que serão utilizados na localização respectivamente.

Sintaxe

```
Set Edit = Server.CreateObject("HiperTools30.EditForm")
```

Comentários

O código abaixo ilustra como obter o objeto EditForm e como utilizar alguma de suas propriedades.

```
<!--#INCLUDE VIRTUAL="/HiperTools/HiperTools30.inc"-->
<%
Dim Edit

Set Edit = Server.CreateObject("HiperTools30.EditForm")

Const conChamaTecnico = 1
Const conListSetor    = 2
Const conListSoma     = 3
Const conListUsuario  = 4

MainBody

REM -----
REM  Formulário de chamado técnico
REM -----
Public Sub GetChamaTecnico

    Edit.DataTable "sncChamaTecnico"

    Edit.DataAddField "chtCodigo",      efDataTypeInt,      "4",
efRequired
    Edit.DataAddField "chtSolicitante", efDataTypeVarChar, "100",
efRequired
    Edit.DataAddField "chtSetor",       efDataTypeVarChar,  "50",
efNotRequired
    Edit.DataAddField "chtDataSolicita", efDataTypeDateTime, "10",
efNotRequired
```

```

    Edit.DataAddField "chtHoraSolicita", efDataTypeVarChar, "5",
efNotRequired
    Edit.DataAddPrimaryKey "chtCodigo"

    Edit.BeginForm "sncChamaTecnico", "Chamado Técnico", 1, sparEditWhat,
-
    efValLocClient

    Edit.FormUnit
"chtSolicitante=Setor;chtSetor,chtDataSolicita,chtHoraSolicita"

    Edit.FormList "chtSolicitante,chtSetor,chtDataSolicita"

    Edit.FormFind "chtSolicitante,chtSetor,chtDataSolicita"

    Edit.AddReport "chtSetor,#1(l=Nº,gc)", "Por Setor", conListSetor
    Edit.AddReport "chtSolicitante,#1(l=Nº,gc)", "Por Usuário",
conListUsuario
    Edit.AddReport "chtSetor,#chtSetor(l=Soma,gs)", "Soma", conListSoma

    Edit.AddField "chtCodigo", "Código", efFldTypeText, , , efNext
    Edit.AddField "chtSolicitante", "Solicitante"
    Edit.AddField "chtSetor", "Setor"
    Edit.AddField "chtDataSolicita", "Data", efFldTypeText, efValOptDate,
' -
    String.LongDate(Now)
    Edit.AddField "chtHoraSolicita", "Horário", efFldTypeText, , , _
    String.LeadingZeroes(Hour(Now), 2) & ":" &
String.LeadingZeroes(Minute(Now), 2)

    Edit.FieldInternalLink "chtSolicitante", sparEditWhat, "chtCodigo"

    Edit.EndForm

End Sub

REM -----
REM Corpo Principal do sistema
REM -----
Private Sub MainBody

    Edit.ConnectionString = "DRIVER={SQL Server};" _
        "SERVER=MyZYCServer;" _
        "UID=Usuario;" _
        "PWD=senha;" _
        "DATABASE=ZTIDatabase"
    Edit.UserID = "Usuario"
    Edit.Password = "Senha"

    Edit.OpenConnection

    Select Case Edit.parWhat
    Case conChamaTecnico
        GetChamaTecnico
    ' Para criar a tabela retire o rem abaixo
    REM Edit.TableCreate "anuMembros"

    Case Else

```

```
End Select

Edit.RedirectActions

Default.HTMLBegin
Default.HeadAll "ANUI - Gerência de Membros"
Default.BodyBegin
Show.Center

Table.BeginTable "65%"
Table.Style.Color2 = ""
Table.BeginRow 4

Table.ColumnVAlign = "Middle"
Table.Column "<img src=""/img/Logo2.gif"" width=""50%""
height=""50%"">"

Table.Column "<Font Color=""DarkBlue"">ANUI - Gerência de Membros"

Table.EndRow
Table.EndTable

FormatTable

Set Edit.Style = Table.Style

If Edit.IsMyAction Then
    ShowFirstPage

End If

Default.PageFooterDefault
Default.BodyEnd
Default.HTMLEnd

End Sub
REM -----
REM Final da Sub MainBody
%>
```

Alguns dos seus recursos são:

- Validação automática de campos;
- Edição de multiplas tabelas num só formulário;
- Localização simples e avançada de registros;
- Dicas para preenchimentos de campos;
- Campos contendo arquivos(Ex.:Imagens, DOCs, PDFs);
- Multiplas ordenações de listagens;
- Relatórios com campos externos, calculados, contados, gráficos e etc;
- Cópias de registros;

E muitas outras facilidades que tornam as edições bastante simples.

Constantes Validation Locations

Estas constantes definem a forma de validação que será utilizada pelo objeto EditForm.

Constante	Valor	Descrição
efValLocClient	True	As validações serão no cliente.
efValLocServer	False	As validações serão no servidor logo.

Aplica-se

EditForm

Veja também

AddField

Constantes QueryString Parameters

Estas constantes definem os parametros que serão utilizados para a criação das QueryStrings durante a execução do EditForm.

Constante	Valor	Descrição
efQueryStrAction	"EA"	Ações feitas pelo EditForm
efQueryStrWhat	"EW"	Tabela que está sendo editada
efQueryStrEditable	"EE"	Indica a editabilidade da tabela
efQueryStrEditableStr	"&EE=1"	Simplifica a colocação da constante acima
efQueryStrFind	"EF"	Posição absoluta de um registro a ser encontrado
efQueryStrFilter	"ER"	Filtro de registros(WHERE) aplicado à tabela
efQueryStrList	"EL"	Relatório a ser apresentado
efQueryStrDefaults	"ED"	Valores default de campos na inclusão
efQueryStrTab	"ET"	Tab da edição
efQueryStrGridStr	"&EGO=1"	
efQueryStrGrid	"EGO"	

Aplica-se
EditForm

Veja também
parWhat, parAction

Constantes QueryString Action Values

Constante	Valor	Descrição
efQSActionEditor	"h01"	Formulário
efQSActionList	"h02"	Listagem
efQSActionSummary	"h03"	Relatório
efQSActionCommonFind	"h04"	Localização simples
efQSActionAdvancedFind	"h05"	Localização avançada
efQSActionAdd	"h06"	Inclusão
efQSActionCopy	"h07"	Cópia
efQSActionEdit	"h08"	Edição
efQSActionDelete	"h09"	Exclusão
efQSActionSave	"h10"	Gravação
efQSActionSaveAdd	"h11"	Gravação e inclusão
efQSActionSaveCopy	"h12"	Gravação e Cópia
efQSActionScreen	"h14"	Apresenta o formulário

Aplica-se
EditForm

Veja também
parAction

Constantes Data Types

Tipos de dados para as colunas das tabelas dos banco de dados.

Constante	Valor	Descrição
EfDataTypeFloat	"float"	Ponto flutuante
EfDataTypeReal	"real"	Real
EfDataTypeVarChar	"varchar"	String de tamanho variante

efDataTypeChar	"char"	String de tamanho fixo
efDataTypeText	"text"	Texto longo (Memo/"Blob")
efDataTypeInt	"int"	Inteiro
efDataTypeDateTime	"datetime"	Data e hora
efDataTypeMoney	"money"	Money
efDataTypeTinyInt	"tinyint"	Inteiro de 1 byte
efDataTypeSmallInt	"smallint"	Inteiro de 2 bytes
efDataTypeBit	"bit"	Booleana

Aplica-se

EditForm

Veja também

DataAddField

Constantes Field Requirement

Constante	Valor	Descrição
efRequired	False	Coluna na tabela obrigatória "NOT NULL"
efNotRequired	True	Coluna na tabela não obrigatória "NULL"

Aplica-se

EditForm

Veja também

AddField

Constantes Relational Integrity

Constante	Valor	Descrição
efRelIntegrCascade	0	Exclusão em cascata
efRelIntegrRestricted	1	Exclusão de registro pai apenas se não tiver filhos
efRelIntegrNullifies	2	Exclusão com a retirada do relacionamento do pai no filho

Aplica-se

EditForm

Veja também

DataAddRelation, DataAddRelationFields

Constantes Relation Type

Constante	Valor	Descrição
efRelType1to1	0	1 para 1
efRelType1toN	1	1 para muitos

Aplica-se

EditForm

Veja também

DataAddRelation, DataAddRelationFields

Constantes Field Types

Tipos de campos utilizados.

Constante	Valor	Descrição
efFldTypeText	0	Texto
efFldTypeUF	1	Estado
efFldTypeLookup	2	Busca os valores apresentados em outra tabela
efFldTypeCheck	3	Marcação
efFldTypeTextArea	4	Texto longo
efFldTypeRadio	5	Botões de seleção
efFldTypeSelect	6	Caixa de seleção
efFldTypePassword	7	Senha
efFldTypeHTTP	8	HTTP
efFldTypeEMail	9	E-Mail
efFldTypeSeparateDate	10	Data com campos separados
efFldTypeImage	11	Imagem
efFldTypeFile	12	Arquivo
efFldTypeAtualization	13	Data de atualização

efFldTypeColor	14	Cor
efFldTypeCountry	15	País
efFldTypeFTP	16	FTP
efFldTypeGopher	17	Gopher

Aplica-se

EditForm

Veja também

AddField

Constantes Validation Options

Tipos de validação disponíveis.

Constante	Valor	Descrição
efValOptNone	0	Nenhuma
efValOptCGC	1	CGC
efValOptCPF	2	CPF
efValOptDate	3	Date
efValOptSepDate	4	Data com campos separados
efValOptDateMToday	5	Data maior que hoje
efValOptSepDateMToday	6	Data com campos separados maior que hoje
efValOptTime	7	Hora
efValOptEmail	8	E-Mail
efValOptCompareDates	9	Comparação de datas(Ex.:A data do primeiro campo não pode ser maior que a do segundo)
efValOptCEP	10	CEP
efValOptCGCCPF	11	CGC ou CPF
efValOptCPF CGC	12	CPF ou CGC

Aplica-se

EditForm

Veja também

AddField

Constantes Field Disable Condition

Constante	Valor	Descrição
efCondDisable	True	Desabilitante
efCondEnable	False	Abilitante

Aplica-se

EditForm

Veja também

FieldDisableValue

Constantes Operator types

Constante	Valor	Descrição
efBooleanOperatorOR	1	operador OR
efBooleanOperatorAND	2	operador AND

Aplica-se

EditForm

Veja também

AddFilter, FilterBooleanOperator

Constantes Character Case

Constante	Valor	Descrição
efCharCaseNormal	0	Maiúsculas/Minúsculas
efCharCaseUpper	1	Maiúsculas
efCharCaseLower	2	Minúsculas

Aplica-se

EditForm

Veja também

AddField

Constantes Alphabetic Index Types

Constante	Valor	Descrição
efAlphaIndexNumeric	1	Somente numérico
efAlphaIndexAlphaNumeric	2	Alfanumérico

efAlphaIndexAlphabetic	3	Somente alfabético (padrão)
------------------------	---	-----------------------------

Aplica-se

EditForm

Veja também

AlphabeticIndex, FieldAlphabeticIndex

Constantes Connection Types

Constante	Valor	Descrição
efConnDBDriverSQL	1	SQL Server ODBC driver*
efConnDBDriverDBase	2	DBase ODBC driver (Jet)*
efConnDBDriverAccess	3	Access ODBC driver (Jet)*
efConnDBDriverExcel	4	Excel ODBC driver (Jet)*
efConnDBDriverFoxPro	5	FoxPro ODBC driver (Jet)*
efConnDBDriverText	6	Text ODBC driver (Jet)*
efConnDBDriverParadox	7	Paradox ODBC driver*
efConnDBProviderJet35	8	Microsoft Jet 3.5 Provider
efConnDBProviderJet40	9	Microsoft Jet 4.0 Provider
efConnDBProviderOracle	10	Oracle Provider 1.0
efConnDBProviderSQL	11	SQL Server Provider 1.0
efConnDBProviderODBC	12	ODBC Provider
efConnDBProviderDTS	13	DTS Provider
efConnDBProviderSQLDTS	14	SQL Server DTS Provider
efConnDBProviderSimple	15	Microsoft Simple Provider
efConnDBProviderRemote	16	Microsoft Remote Provider

* ODBC Provider (MSDASQL Provider)

Aplica-se

EditForm

Veja também

ConnectionType

Constantes Identity

Constante	Valor	Descrição
efNext	"proximo"	Autonumeração/contador/identity

Aplica-se

EditForm

Veja também

AddField

Propriedade AllowIdentityUpdate

Descrição

Define ou retorna se é permitido alterar campos definidos com a constante efNext, ou seja um campo do tipo *Identity*, *Counter* ou Auto-numeração. Ela normalmente está definida como false.

Sintaxeobjeto.**AllowIdentityUpdate** [= *Boolean*]

O argumento objeto sempre será o objeto EditForm.

ComentáriosO código abaixo ilustra o uso da propriedade **AllowIdentityUpdate**.

```
Edit.AllowIdentityUpdate = True
```

Aplica-se

EditForm

Veja também

AddField, "Constantes Identity"

Propriedade AlphabeticIndex

Descrição

Define ou retorna se será apresentado o índice alfabético na listagem para o campo sendo ordenado no momento.

Sintaxeobjeto. **AlphabeticIndex** [= *Boolean*]

O argumento objeto sempre será o objeto EditForm.

ComentáriosO código abaixo ilustra o uso da propriedade **AlphabeticIndex**.

```
Edit. AlphabeticIndex = True
```

Aplica-se

EditForm

Veja também

BeginForm. FormList, AddOrder

Propriedade Buttons

Descrição

Daqui se acessa a área que personaliza os botões do Formulário do EditForm.

Sintaxe

objeto.**Buttons**.<nome_do_botão>Caption [= *String*]
 objeto.**Buttons**.<nome_do_botão>Title [= *String*]
 objeto.**Buttons**. Show<nome_do_botão> [= *Boolean*]
 objeto.**Buttons**.<nome_do_botão>Class [= *String*]
 objeto.**Buttons**.<nome_do_botão>StyleSheet [= *String*]
 objeto.**Buttons**.<nome_do_botão>AccessKey [= *String*]
 objeto.**Buttons**.<nome_do_botão>Image [= *String*]

Parte	Descrição
Objeto	Requerido. Sempre terá que ser um objeto EditForm.
Show	<i>Boolean</i> faz com que se apresente ou não o botão.
Caption	<i>String</i> é o rótulo do campo
Title	<i>String</i> é o texto explicativo que é mostrado quando se passa o mouse sobre o botão.
Class	<i>String</i> é a classe <i>Cascade Style Sheet</i> definida anteriormente que pode ser passada para o botão.
StyleSheet	<i>String</i> é a classe <i>Cascade Style Sheet in-line</i> a ser definida no botão.
AccessKey	<i>String</i> é o caracter que quando teclado no formato <i>Alt + <tecla></i> ativa o botão.
Image	<i>String</i> quando preenchido com algum valor apresenta uma imagem ao invés de um botão

nome_do_botão:

Parte	Descrição
First	Primeiro registro. Rótulo padrão: : "<"
Previous	Registro anterior. Rótulo padrão: : "<"
Add	Incluir registro. Rótulo padrão: : "Incluir"
Edit	Alterar registro. Rótulo padrão: : "Alterar"
Salvar	Salvar registro. Rótulo padrão: : "Salvar"
Find	Pesquisa. Rótulo padrão: : "Localizar"
List	Acesso à listagem. Rótulo padrão: : "Listagem"
SaveAdd	Salvar e Incluir registro. Rótulo padrão: : "Salvar e Incluir"
SaveCopy	Salvar e Incluir cópia do registro. Rótulo padrão: : "Salvar e

	Copiar
Cancel	Cancelar ação. Rótulo padrão: "Cancelar"
Copy	Copiar registro. Rótulo padrão: "Copiar"
Next	Próximo registro. Rótulo padrão: ">"
Last	Ultimo registro. Rótulo padrão: "> "

Comentários

O código abaixo ilustra o uso da propriedade **ConnectionType**.

```
`Defina a classe MyClass com a tag <STYLE>

Edit.Buttons.FindCaption = "Procurar"
Edit.Buttons.FindTitle = "Entra no módulo de pesquisa de registros"
Edit.Buttons.FindClass = "MyClass"
Edit.Buttons.SaveImage = "/img/save.gif"
Edit.Buttons.SaveStyleSheet = "decoration:none; back-color:blue"
```

Aplica-se

EditForm

Veja também

ShowButtons

PropriedadeConnectionString

Descrição

Define ou retorna a ConnectionString que será utilizada para a conexão com o servidor de banco de dados.

Sintaxe

objeto.**ConnectionString** [= String]

O argumento objeto sempre será o objeto EditForm. O valor para **ConnectionString** deverá ser uma string completa para a conexão com o banco de dados.

Comentários

O código abaixo ilustra o uso da propriedade **ConnectionString**.

```
`Recomenda-se o uso do Provider exato e não do driver ODBC
` (MSDASQL.1)

Edit.ConnectionString = "Provider= Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=; " & Initializer.SiteRootDir & "\MyData.mdb" & _
    "User ID=Usuario; " & _
    "Password=senha; "

`ou

Edit.ConnectionString = "Provider=SQLOLEDB.1;" & _
    "Data Source=MyServer; " & _
    "User ID=Usuario; " & _
    "Password=senha; " & _
    "Initial Catalog=MyDatabase"
```

'ou

```
Edit.ConnectionString = "DRIVER={SQL Server};" & _  
    "SERVER=MyServer;" & _  
    "UID=Usuario;" & _  
    "PWD=senha;" & _  
    "DATABASE=MyDatabase"  
  
Edit.UserID = "Usuario"  
Edit.Password = "Senha"
```

Aplica-se

EditForm

Veja também

UserID, Password

Propriedade ConnectionType

Descrição

Define ou retorna o tipo de conexão com o Banco de Dados. O valor está no escopo das constantes *Connection Types*. O EditForm altera o valor de retorno desta propriedade toda vez que se altera a propriedade ConnectionString.

Sintaxe

objeto.**ConnectionType** [= Integer]

O argumento objeto sempre será o objeto EditForm.

Comentários

O código abaixo ilustra o uso da propriedade **ConnectionType**.

```
Edit.ConnectionString = "Provider=SQLOLEDB.1;" & _  
    "Data Source=MyServer;" & _  
    "User ID=Usuario;" & _  
    "Password=senha;" & _  
    "Initial Catalog=MyDatabase"  
  
Response.Write Edit.ConnectionType  
Retorna: 11 'efConnDBProviderSQL  
  
Edit.UserID = "Usuario"  
Edit.Password = "Senha"
```

Aplica-se

EditForm

Veja também

ConnectionString

Propriedade CurrConnection

Descrição

Retorna uma referência para a conexão atual com o Banco de Dados. O retorno é do tipo "ADODB.Connection". Esta propriedade só retorna algo após o método OpenConnection.

Sintaxe

objeto.**CurrConnection**

O argumento objeto sempre será o objeto EditForm.

Comentários

O código abaixo ilustra o uso da propriedade **UserID**.

```
Dim myConn
Set myConn = Server.CreateObject("ADODB.Connection")

Edit.ConnectionString = "Provider=SQLOLEDB.1;" & _
    "Data Source=MyServer;" & _
    "User ID=Usuario;" & _
    "Password=senha;" & _
    "Initial Catalog=MyDatabase"

Edit.UserID = "Usuario"
Edit.Password = "Senha"

Edit.OpenConnection

If IsObject(Edit.CurrConnection) Then
    Set myConn = Edit.CurrConnection
End If
```

Aplica-se

EditForm

Veja também

ConnectionString, ConnectionType, OpenConnection

Propriedade DebugMode

Descrição

Define ou retorna o estado de depuração de erros de Queries SQL no banco de dados. Se uma query gerada pelo EditForm gerar um erro de ADO (objeto "ADODB.Error"), serão apresentados todos os erros gerado e também é exposta a coleção Errors que é do tipo "ADODB.Errors".

Sintaxe

objeto.**DebugMode** [= *Boolean*]

O argumento objeto sempre será o objeto EditForm.

Comentários

O código abaixo ilustra o uso da propriedade **DebugMode**.

```
Edit.DebugMode = True
```

Aplica-se

EditForm

Veja também

Errors, ShowSQLQuery

Propriedade Errors

Descrição

Retorna uma referência para uma coleção do tipo "ADODB.Errors" com todos os erros que ocorreram no momento. Veja a documentação padrão do ADO.

Sintaxe

objeto.**Errors**

O argumento objeto sempre será o objeto EditForm.

Comentários

O código abaixo ilustra o uso da propriedade **Errors**.

```
Dim objErrors, objError
Set objErrors = Server.CreateObject("ADODB.Errors")

Edit.DebugMode = True

' Definição dos dados...
.
.
.

If Edit.Errors.Count > 0 Then
    Set objErrors = Edit.Errors

    For Each objError In objErrors
        Response.Write objError.Number & " " & objError.Description
    Next

End If
```

Aplica-se

EditForm

Veja também

ShowSQLQuery, DebugMode

Propriedade FirstPageLink

Descrição

Define o link de página inicial apresentado no final das páginas dos formulários gerados pelo EditForm.

Sintaxe

objeto.**FirstPageLink** = *String*

O argumento objeto sempre será o objeto EditForm.

Comentários

O código abaixo ilustra o uso da propriedade **FirstPageLink**.

```
Edit.FirstPageLink = "outro.asp"
```

Aplica-se

EditForm

Veja também

ShowFirstLastPageLink

Propriedade FilterBooleanOperator

Descrição

Define o operador utilizado entre os AddFilter definidos.

Sintaxe

objeto.**FilterBooleanOperator** [= *Integer*]

O argumento objeto sempre será o objeto EditForm.

Comentários

O código abaixo ilustra o uso da propriedade **FilterBooleanOperator**.

```
Edit.FilterBooleanOperator = efBooleanOperatorOR
Edit.AddFilter "chtCodigo", "5"
Edit.AddFilter "chtCodigo", "1"
Edit.FilterBooleanOperator = efBooleanOperatorAND
Edit.AddFilter "chtCodigo", "25", "<>"
```

Aplica-se

EditForm

Vea também

AddFilter, “Constantes Operator Types”

Propriedade LinkTarget

Descrição

Define ou retorna o target dos links apresentados pelo EditForm.

Sintaxe

objeto.**LinkTarget** [= *String*]

O argumento objeto sempre será o objeto EditForm.

Comentários

O código abaixo ilustra o uso da propriedade **LinkTarget**

```
Edit.LinkTarget = "_blank"
```

Aplica-se

EditForm

Vea também

FieldInternalLink, FieldExternalLink

Propriedade ListTitle

Descrição

Define ou retorna o título do Listagem, possibilitando assim, retirar o prefixo padrão “Lista de”.

Sintaxe

objeto.**ListTitle** [= *String*]

O argumento objeto sempre será o objeto EditForm.

Comentários

O código abaixo ilustra o uso da propriedade **ListTitle**

```
Edit.BeginForm "sncChamaTecnico", "Chamado Técnico", 1, sparEditWhat, _
    efValLocClient
```

```
Edit.FormUnit "chtSolicitante=Setor;chtSetor,chtDataSolicita,chtHoraSolicita"
```

```
Edit.FormList "chtSolicitante,chtSetor,chtDataSolicita"
```

```
Edit.FormFind "chtSolicitante,chtSetor,chtDataSolicita"
```

```
Edit.ListTitle = "Listagem Especial"
.
.
Edit.EndForm
```

Aplica-se

EditForm

Veja também

FormList

Propriedade ListReportsCaption

Descrição

Define ou retorna a frase que aparece no link “Outras Listagens”.

Sintaxe

objeto.**ListReportsCaption** [= *String*]

O argumento objeto sempre será o objeto EditForm.

Comentários

O código abaixo ilustra o uso da propriedade **ListReportsCaption**.

```
Edit.ListReportsCaption = "Listagens diferentes"
```

Aplica-se

EditForm

Veja também

FormList, AddList, ShowReportsLink

Propriedade ListSetOfOrdersCaption

Descrição

Define ou retorna a frase que aparece no link “Outras Ordenações”.

Sintaxe

objeto.**ListSetOfOrdersCaption** [= *String*]

O argumento objeto sempre será o objeto EditForm.

Comentários

O código abaixo ilustra o uso da propriedade **ListSetOfOrdersCaption**.

```
Edit.ListSetOfOrdersCaption = "Ordenações diferentes"
```

Aplica-se

EditForm

Veja também

FormList, AddOrder, ShowSetOfOrdersLink

Propriedade LockTimeOut

Descrição

Define ou retorna tempo, em minutos, que expira-se o travamento de registro ou *record lock*. Normalmente esta propriedade inicia-se com o valor da propriedade Session.Timeout.

Sintaxe

objeto.**LockTimeOut** [= Integer]

O argumento objeto sempre será o objeto EditForm.

Comentários

O código abaixo ilustra o uso da propriedade **ListTitle**

```
Edit.BeginForm "sncChamaTecnico", "Chamado Técnico", 1, sparEditWhat, _  
    efValLocClient
```

```
Edit.FormUnit "chtSolicitante=Setor;chtSetor,chtDataSolicita,chtHoraSolicita"  
Edit.FormList "chtSolicitante,chtSetor,chtDataSolicita"  
Edit.FormFind "chtSolicitante,chtSetor,chtDataSolicita"
```

```
Edit.LockTimeOut = 15 'minutos
```

```
Edit.EndForm
```

Aplica-se

EditForm

Veja também

RedirectActions, ShowActions

Propriedade Password

Descrição

Define ou retorna a Password que será utilizado para a conexão com o servidor de banco de dados.

Sintaxe

objeto.**Password** [= String]

O argumento objeto sempre será o objeto EditForm. O valor para **Password** deverá ser uma string com o usuário com direitos de acesso ao Banco de Dados.

Comentários

O código abaixo ilustra o uso da propriedade **Password**.

```
Edit.ConnectionString = "Provider=SQLOLEDB.1;" & _  
    "Data Source=MyServer;" & _  
    "User ID=Usuario;" & _  
    "Password=senha;" & _  
    "Initial Catalog=MyDatabase"  
Edit.UserID = "Usuario"  
Edit.Password = "Senha"
```

Aplica-se

EditForm

Veja também

UserID, ConnectionString

Propriedade parWhat

Descrição

Para desenvolver com o EditForm deve-se modularizar todos as definições de dados – DataAddField, DataTable – e de apresentação de formulários – BeginForm, AddField – em módulos e deve-se controlar qual módulo executar com constantes para cada módulo; veja o exemplo – conChamaTecnico. De acordo com o que foi definido nos módulos, no exemplo GetChamaTecnico, o método ShowActions ou IsMyAction apresentará o que for adequado – ver parAction.

Sintaxe

objeto.parWhat

Comentários

O código abaixo ilustra o uso do método **parWhat**.

```
Edit.OpenConnection

Select Case Edit.parWhat
Case conChamaTecnico
    GetChamaTecnico

Case Else

End Select
```

Aplica-se

EditForm

Veja também

parAction, ShowActions, IsMyAction, RedirectActions, DataAddField, DataTable, AddField, BeginForm

Propriedade parAction

Descrição

Retorna a ação que esta sendo executada pelo EditForm neste momento. Esta informação esta no parametro QueryString da constante “efQueryStrAction” obtido pelo EditForm

Sintaxe

objeto.parAction

Comentários

O código abaixo ilustra o uso do método **parAction**.

```
Edit.OpenConnection

Select Case Edit.parAction
Case efQSActionList
    ShowAcaoListar

Case Else

End Select
```

Aplica-se

EditForm

Veja também

parWhat, ShowActions, IsMyAction, RedirectActions, "Constantes QueryString Action Values"

Propriedade ReportTitle

Descrição

Define ou retorna o título do relatório.

Sintaxe

objeto.**ReportTitle** [= *String*]

O argumento objeto sempre será o objeto EditForm.

Comentários

O código abaixo ilustra o uso da propriedade **ReportTitle**.

```
Edit.ReportTitle = "Gráfico de Barras"
```

Aplica-se

EditForm

Veja também

FormList, AddList

Propriedade ShowButtons

Descrição

Define ou retorna se deve-se apresentar a barra de botões do formulário.

Sintaxe

objeto.**ShowButtons** [= *Boolean*]

O argumento objeto sempre será o objeto EditForm.

Comentários

O código abaixo ilustra o uso da propriedade **ShowButtons**.

```
Edit.ShowButtons = False
```

Aplica-se

EditForm

Veja também

Buttons, FormUnit, ShowUnitUpperBar

Propriedade ShowListUpperBar

Descrição

Define ou retorna se deve-se apresentar a barra superior da Listagem.

Sintaxe

objeto.**ShowListUpperBar** [= *Boolean*]

O argumento objeto sempre será o objeto EditForm.

Comentários

O código abaixo ilustra o uso da propriedade **ShowListUpperBar**.

```
Edit.ShowListUpperBar = False
```

Aplica-se

EditForm

Veja também

FormList, ShowReportsLink, ShowSetOfOrdersLink

Propriedade ShowSetOfOrdersLink

Descrição

Define ou retorna se deve-se apresentar o link “Outras Ordenações” na listagem.

Sintaxe

objeto.**ShowSetOfOrdersLink** [= *Boolean*]

O argumento objeto sempre será o objeto EditForm.

Comentários

O código abaixo ilustra o uso da propriedade **ShowSetOfOrdersLink**.

```
Edit.ShowSetOfOrdersLink = False
```

Aplica-se

EditForm

Veja também

FormList, AddOrder, ListSetOfOrdersCaption, ShowListUpperBar

Propriedade ShowRecordCount

Descrição

Define ou retorna se deve-se apresentar a contagem de registros – “Registro x de n” ou “Registros: n” – no formulário.

Sintaxe

objeto.**ShowRecordCount** [= *Boolean*]

O argumento objeto sempre será o objeto EditForm.

Comentários

O código abaixo ilustra o uso da propriedade **ShowRecordCount**.

```
Edit.ShowRecordCount = False
```

Aplica-se

EditForm

Veja também

FormUnit, ShowListUpperBar

Propriedade ShowSQLQuery

Descrição

Define ou retorna se deve-se apresentar as Querys SQL geradas pelo EditForm. Quando verdadeira todas as Querys dentro do escopo da propriedade, que não é persistente, são apresentadas.

Sintaxe

objeto.**ShowSQLQuery** [= *Boolean*]

O argumento objeto sempre será o objeto EditForm.

Comentários

O código abaixo ilustra o uso da propriedade **ShowSQLQuery**.

```
Edit.ShowSQLQuery = False
```

Aplica-se

EditForm

Veja também

DebugMode, CurrConnection, Errors

Propriedade ShowReportsLink

Descrição

Define ou retorna se deve-se apresentar o link “Outras listagens” na Listagem.

Sintaxe

objeto.**ShowReportsLink** [= *Boolean*]

O argumento objeto sempre será o objeto EditForm.

Comentários

O código abaixo ilustra o uso da propriedade **ShowReportsLink**.

```
Edit.ShowReportsLink = False
```

Aplica-se

EditForm

Veja também

FormList, ListReportsCaption, ShowListUpperBar

Propriedade ShowUnitUpperBar

Descrição

Define ou retorna se deve-se apresentar a barra superior do formulário.

Sintaxe

objeto.**ShowUnitUpperBar** [= *Boolean*]

O argumento objeto sempre será o objeto EditForm.

Comentários

O código abaixo ilustra o uso da propriedade **ShowUnitUpperBar**.

```
Edit.ShowUnitUpperBar = False
```

Aplica-se

EditForm

Veja também

FormUnit, ShowRecordCount, ShowListUpperBar

Propriedade Style

Descrição

Define a formatação das tabelas apresentadas pelo EditForm.

Sintaxe

objeto.**Style**.*Propriedade*

O argumento objeto sempre será o objeto EditForm.

Comentários

Aplica-se

EditForm, Table

Veja também

TableStyle, Table.Style

Propriedade UnitTitle

Descrição

Define ou retorna o título do Formulário corrente.

Sintaxe

objeto.**UnitTitle** [= *String*]

O argumento objeto sempre será o objeto EditForm.

Comentários

O código abaixo ilustra o uso da propriedade **UnitTitle**.

```
Edit.UnitTitle = "Formulário de Inscrição"
```

Aplica-se

EditForm

Veja também

FormUnit

Propriedade UserID

Descrição

Define ou retorna o UserID que será utilizado para a conexão com o servidor de banco de dados.

Sintaxe

objeto.**UserID** [= *String*]

O argumento objeto sempre será o objeto EditForm. O valor para **UserID** deverá ser uma string com o usuário com direitos de acesso ao Banco de Dados.

Comentários

O código abaixo ilustra o uso da propriedade **UserID**.

```
Edit.ConnectionString = "Provider=SQLOLEDB.1;" & _  
    "Data Source=MyServer;" & _  
    "User ID=Usuario;" & _  
    "Password=senha;" & _  
    "Initial Catalog=MyDatabase"
```

```
Edit.UserID = "Usuario"  
Edit.Password = "Senha"
```

Aplica-se

EditForm

Veja também

ConnectionString, Password

Método IsMyAction

Descrição

Retorna false e monta a tela do formulário que foi definido no FormBegin, e retorna verdadeiro se não for uma ação do EditForm e sim .

Sintaxe

objeto.**IsMyAction**

Comentários

O código abaixo ilustra o uso do método **IsMyAction**.

```
If Edit.IsMyAction Then  
    ShowFirstPage
```

End If

Aplica-se

EditForm

Veja também

ShowActions, RedirectActions, parAction, parWhat

Método FieldValue

Descrição

Retorna o valor de de um campo de uma determinada tabela durante a execução do EditForm. Este valor só é preenchido após o a apresentação da “ação “ do EditForm

Sintaxe

objeto.**FieldValue** *String1*, *String2*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String1</i>	Requerido. Nome válido de uma tabela que está em uso pelo EditForm.
<i>String2</i>	Requerido. Nome válido de uma coluna que esta sendo utilizado pelo EditForm.

Comentários

O código abaixo ilustra o uso do método **FieldValue**.

```
Response.Write Edit.FieldValue("sncChamaTecnico", "chtSolicitante")
```

Aplica-se

EditForm

Veja também

AddField, DataAddField, IsMyAction, ShowActions

Método HasTable

Descrição

Retorna se existe a tabela discriminada no banco de dados. Compatível com SQL Server e Access

Sintaxe

objeto.**HasTable** *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. <i>String</i> é nome válido de uma tabela a ser

	consultada no servidor SQL.
--	-----------------------------

Comentários

O código abaixo ilustra o uso do método **HasTable**.

```
If Edit.HasTable("sncChamaTecnico") Then
    Response.Write "Tabela existe!"
Else
    Response.Write "Tabela não existe!"
End If
```

Aplica-se

EditForm

Veja também

Método ShowActions

Descrição

Monta a tela de acordo com a ação que esta no parametro QueryString da constante "efQueryStrAction" obtido pelo EditForm.

Sintaxe

objeto.ShowActions

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.

Comentários

O código abaixo ilustra o uso do método **ShowActions**.

```
Edit.ShowActions
```

Aplica-se

EditForm

Veja também

IsMyAction, RedirectActions

Método RedirectActions

Descrição

Gerencia a execução das funções de gravação no Banco de Dados, Validação do formulário etc. Decide qual função a ser executada de acordo com as ações do parametro QueryString da constante "efQueryStrAction" obtido pelo EditForm.

Sintaxe

objeto.RedirectActions

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.

Comentários

O código abaixo ilustra o uso do método **RedirectActions**.

```
Edit.RedirectActions
```

Aplica-se

EditForm

Veja também

ShowActions, IsMyAction

Método OpenConnection

Descrição

Processa todos os parametros do OpenConnection que serão utilizados pelo EditForm. Também faz a conexão com o banco de dados.

Sintaxe

objeto.OpenConnection

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.

Comentários

O código abaixo ilustra o uso do método **OpenConnection**.

```
Edit.OpenConnection
```

```
Select Case Edit.parWhat  
Case conChamaTecnico  
    GetChamaTecnico
```

```
Case Else
```

```
End Select
```

Aplica-se

EditForm

Veja também

parAction, parWhat

Método DataTable

Descrição

Define início da definição da tabela do dicionário de dados que será utilizada pelo EditForm.

Sintaxe*objeto.DataTable String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. Nome válido de uma tabela que será utilizada pelo EditForm.

Comentários

O código abaixo ilustra o uso do método **DataTable**.

```

Edit.DataTable "sncChamaTecnico"

Edit.DataAddField "chtCodigo",          efDataTypeInt,          "4",
efRequired
Edit.DataAddField "chtSolicitante",      efDataTypeVarChar, "100",
efRequired
Edit.DataAddField "chtSetor",            efDataTypeVarChar,  "50",
efNotRequired
Edit.DataAddField "chtDataSolicita",     efDataTypeDateTime, "10",
efNotRequired
Edit.DataAddField "chtHoraSolicita",     efDataTypeVarChar,  "5",
efNotRequired
Edit.DataAddPrimaryKey "chtCodigo"

```

Aplica-se

EditForm

Veja também

DataAddField, DataAddPrimaryKey, DataAddRelation

Método DataAddField**Descrição**

Define o nome da coluna, tipo do dado, tamanho e se será requerido ou não.

Sintaxe*objeto.DataAddField String1, String2, Integer, Boolean*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String1</i>	Requerido. Nome válido de uma coluna que será utilizado na tabela que será utilizada pelo EditForm.
<i>String2</i>	Requerido. Tipo de dado que será utilizado pelo EditForm.
<i>Integer</i>	Requerido. Tamanho da coluna, que será utilizado pelo EditForm.
<i>Boolean</i>	Requerido. Define se a coluna é ou não requerida. Equivalente ao NOT NULL e NULL do SQL.

Comentários

O código abaixo ilustra o uso do método **DataAddField**.

```
Edit.DataTable "sncChamaTecnico"

    Edit.DataAddField "chtCodigo",          efDataTypeInt,          "4",
efRequired
    Edit.DataAddField "chtSolicitante",     efDataTypeVarChar, "100",
efRequired
    Edit.DataAddField "chtSetor",           efDataTypeVarChar,  "50",
efNotRequired
    Edit.DataAddField "chtDataSolicita",    efDataTypeDateTime, "10",
efNotRequired
    Edit.DataAddField "chtHoraSolicita",    efDataTypeVarChar,  "5",
efNotRequired
    Edit.DataAddPrimaryKey "chtCodigo"
```

Aplica-se

EditForm

Veja também

DataTable, DataAddPrimaryKey, DataAddRelation, "Constantes Data Types"

Método DataAddPrimaryKey

Descrição

Define a chave primária da tabela do dicionário de dados que será utilizada pelo EditForm.

Sintaxe

objeto.DataAddPrimaryKey String

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. Nome válido de uma coluna definina pelo DataAddField.

Comentários

O código abaixo ilustra o uso do método **DataAddPrimaryKey**.

```
Edit.DataTable "sncChamaTecnico"

    Edit.DataAddField "chtCodigo",          efDataTypeInt,          "4",
efRequired
    Edit.DataAddField "chtSolicitante",     efDataTypeVarChar, "100",
efRequired
    Edit.DataAddField "chtSetor",           efDataTypeVarChar,  "50",
efNotRequired
    Edit.DataAddField "chtDataSolicita",    efDataTypeDateTime, "10",
efNotRequired
    Edit.DataAddField "chtHoraSolicita",    efDataTypeVarChar,  "5",
efNotRequired
    Edit.DataAddPrimaryKey "chtCodigo"
```

Aplica-se

EditForm

Veja também

DataTable, DataAddField, DataAddRelation

Método DataAddRelation

Descrição

Define o relacionamento entre a tabela atual com outra e define a opção da manutenção da integridade para exclusão e atualização, bem como o tipo de relacionamento.

Sintaxe

objeto. **DataAddRelation** *String*, *Integer1*, *Integer2*, *Integer3*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. Nome válido de uma tabela que será relacionada pelo EditForm.
<i>Integer1</i>	Requerido. Tipo de atualização durante uma exclusão.
<i>Integer2</i>	Requerido. Tipo de atualização durante uma atualização.
<i>Integer3</i>	Requerido. Tipo de relacionamento entre as tabelas.

Comentários

O código abaixo ilustra o uso do método **DataAddRelation**.

```

Edit.DataTable "sncChamaTecnico"

    Edit.DataAddField "chtCodigo",          efDataTypeInt,          "4",
efRequired
    Edit.DataAddField "chtSolicitante",    efDataTypeVarChar, "100",
efRequired
    Edit.DataAddField "chtSetor",          efDataTypeVarChar,  "50",
efNotRequired
    Edit.DataAddField "chtDataSolicita",  efDataTypeDateTime, "10",
efNotRequired
    Edit.DataAddField "chtHoraSolicita",  efDataTypeVarChar,  "5",
efNotRequired
    Edit.DataAddPrimaryKey "chtCodigo"
    Edit.DataAddRelation "sncSolicitantes", efRelIntegrCascade, _
    efRelIntegrCascade, efRelType1to1
    Edit.DataAddRelation "chtSolicitante", "cliCodigo"

```

Aplica-se

EditForm

Veja também

DataTable, DataAddPrimaryKey, DataAddRelation, “Constantes Relational Integrity”

Método DataAddRelationFields

Descrição

Define as colunas que serão utilizados durante o relacionamento. Somente a última definição será aceita.

Sintaxe

objeto.DataAddRelationFields String1, String2

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String1</i>	Requerido. Nome válido da coluna que será utilizada no relacionamento da tabela atual.
<i>String2</i>	Requerido. Nome válido da coluna que será utilizada no relacionamento da tabela relacionada.

Comentários

O código abaixo ilustra o uso do método **DataAddRelationFields**

```
Edit.DataTable "sncChamaTecnico"

    Edit.DataAddField "chtCodigo",          efDataTypeInt,          "4",
efRequired
    Edit.DataAddField "chtSolicitante",    efDataTypeVarChar, "100",
efRequired
    Edit.DataAddField "chtSetor",          efDataTypeVarChar,  "50",
efNotRequired
    Edit.DataAddField "chtDataSolicita",   efDataTypeDateTime, "10",
efNotRequired
    Edit.DataAddField "chtHoraSolicita",   efDataTypeVarChar,  "5",
efNotRequired
    Edit.DataAddPrimaryKey "chtCodigo"
    Edit.DataAddRelation "sncSolicitantes", efRelIntegrCascade, _
efRelIntegrCascade, efRelType1to1
    Edit.DataAddRelationFields "chtSolicitante", "cliCodigo"
```

Aplica-se

EditForm

Veja também

DataTable, DataAddPrimaryKey, DataAddRelation

Método DataAddIndex

Descrição

Definição dos índices normais que serão criados para a tabela.

Sintaxe

objeto.DataAddIndex String1, String2

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String1</i>	Requerido. Nome válido para o índice.

<i>String2</i>	Requerido. Coluna ou as colunas que serão utilizadas no índice.
----------------	---

Comentários

O código abaixo ilustra o uso do método **DataAddIndex**

```

Edit.DataTable "sncChamaTecnico"

    Edit.DataAddField "chtCodigo",          efDataTypeInt,          "4",
efRequired
    Edit.DataAddField "chtSolicitante",      efDataTypeVarChar, "100",
efRequired
    Edit.DataAddField "chtSetor",           efDataTypeVarChar,  "50",
efNotRequired
    Edit.DataAddField "chtDataSolicita",    efDataTypeDateTime, "10",
efNotRequired
    Edit.DataAddField "chtHoraSolicita",    efDataTypeVarChar,  "5",
efNotRequired
    Edit.DataAddPrimaryKey "chtCodigo"
    Edit.DataAddRelation "sncSolicitantes", efRelIntegrCascade, _
efRelIntegrCascade, efRelType1to1
    Edit.DataAddRelationFields "chtSolicitante", "cliCodigo"
    Edit.DataAddIndex "Solicitante", "chtSolicitante, chtSetor"
    Edit.DataAddIndex "Setor", "chtSetor"

```

Aplica-se

EditForm

Veja também

DataTable, DataAddPrimaryKey, DataAddRelation

Método DataIndexClustered

Descrição

Definição do índice clustered que será criados para a tabela.

Sintaxe

objeto.**DataIndexClustered** *String1*, *String2*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String1</i>	Requerido. Nome válido para o arquivo de índice clustered.
<i>String2</i>	Requerido. Coluna ou as colunas, separadas por vírgulas, que serão utilizadas no índice.

Comentários

O código abaixo ilustra o uso do método **DataIndexClustered**

```

Edit.DataTable "sncChamaTecnico"

    Edit.DataAddField "chtCodigo",          efDataTypeInt,          "4",
efRequired

```

```
Edit.DataAddField "chtSolicitante", efDataTypeVarChar, "100",  
efRequired  
Edit.DataAddField "chtSetor", efDataTypeVarChar, "50",  
efNotRequired  
Edit.DataAddField "chtDataSolicita", efDataTypeDateTime, "10",  
efNotRequired  
Edit.DataAddField "chtHoraSolicita", efDataTypeVarChar, "5",  
efNotRequired  
Edit.DataAddPrimaryKey "chtCodigo"  
Edit.DataAddRelation "sncSolicitantes", efRelIntegrCascade, _  
efRelIntegrCascade, efRelType1to1  
Edit.DataAddRelationFields "chtSolicitante", "cliCodigo"  
Edit.DataAddIndex "Solicitante", "chtSolicitante, chtSetor"  
Edit.DataIndexClustered "Setor", "chtSetor"
```

Aplica-se

EditForm

Veja também

DataTable, DataAddPrimaryKey, DataAddRelation

Método BeginForm

Descrição

Define o início da definição do formulário. Especifica-se o nome da tabela, definida no DataTable e o *What* definido com alguma constante (veja – parWhat).

Sintaxe

objeto.**BeginForm** String1, String2, Integer, String3, Boolean

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String1</i>	Requerido. Nome válido de uma tabela que será utilizada definida pelo DataTable.
<i>String2</i>	Requerido. Texto do cabeçalho do formulário.
<i>Integer</i>	Requerido. Opção da edição, indica se este formulário será ou não editável – se for zero (0), o formulário será somente de leitura.
<i>String3</i>	Requerido. Identificador do formulário, retornado pelo parWhat.
<i>Boolean</i>	Requerido. Forma validação do formulário, no cliente ou no servidor.

Comentários

O código abaixo ilustra o uso do método **BeginForm**.

```
Edit.BeginForm "sncChamaTecnico", "Chamado Técnico", 1, sparEditWhat,  
_ efValLocClient
```

```

Edit.FormUnit
"chtSolicitante=Setor;chtSetor,chtDataSolicita,chtHoraSolicita"

Edit.FormList "chtSolicitante,chtSetor,chtDataSolicita"

Edit.FormFind "chtSolicitante,chtSetor,chtDataSolicita"

Edit.AddReport "chtSetor,#1(l=Nº,gc)", "Por Setor", conListSetor
Edit.AddReport "chtSolicitante,#1(l=Nº,gc)", "Por Usuário",
conListUsuario

Edit.AddField "chtCodigo", "Código", efFldTypeText, , , efNext
Edit.AddField "chtSolicitante", "Solicitante"
Edit.AddField "chtSetor", "Setor"
Edit.AddField "chtDataSolicita", "Data", efFldTypeText, efValOptDate,
, _
String.LongDate(Now)
Edit.AddField "chtHoraSolicita", "Horário", efFldTypeText, , , _
String.LeadingZeroes(Hour(Now), 2) & ":" &
String.LeadingZeroes(Minute(Now), 2)

Edit.FieldInternalLink "chtSolicitante", sparEditWhat, "chtCodigo"

Edit.EndForm

```

Aplica-se

EditForm

Veja também

FormFind, AddField, AddList, EndForm, FormUnit, FormList, “Constantes de Validation Location”

Método EndForm

Descrição

Término da definição do formulário que iniciou com o FormBegin.

Sintaxe*objeto*.EndForm

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.

ComentáriosO código abaixo ilustra o uso do método **EndForm**.

```

Edit.BeginForm "sncChamaTecnico", "Chamado Técnico", 1, sparEditWhat, _
efValLocClient

Edit.FormUnit
"chtSolicitante=Setor;chtSetor,chtDataSolicita,chtHoraSolicita"

Edit.FormList "chtSolicitante,chtSetor,chtDataSolicita"

Edit.FormFind "chtSolicitante,chtSetor,chtDataSolicita"

```

```
Edit.AddReport "chtSetor,#1(l=Nº,gc)", "Por Setor", conListSetor
Edit.AddReport "chtSolicitante,#1(l=Nº,gc)", "Por Usuário",
conListUsuario

Edit.AddField "chtCodigo", "Código", efFldTypeText, , , efNext
Edit.AddField "chtSolicitante", "Solicitante"
Edit.AddField "chtSetor", "Setor"
Edit.AddField "chtDataSolicita", "Data", efFldTypeText, efValOptDate, ,
_
String.LongDate(Now)
Edit.AddField "chtHoraSolicita", "Horário", efFldTypeText, , , _
String.LeadingZeroes(Hour(Now), 2) & ":" &
String.LeadingZeroes(Minute(Now), 2)

Edit.FieldInternalLink "chtSolicitante", sparEditWhat, "chtCodigo"

Edit.EndForm
```

Aplica-se

EditForm

Veja também

BeginForm

Método FormFind

Descrição

Determina as as colunas que serão utilizadas no sistema de localização.

Sintaxe*objeto*.**FormFind** *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. Lista sequencial com as identificações das colunas separadas por vírgulas.

ComentáriosO código abaixo ilustra o uso do método **FormFind**.

```
Edit.BeginForm "sncChamaTecnico", "Chamado Técnico", 1, sparEditWhat, _
efValLocClient

Edit.FormUnit
"chtSolicitante=Setor;chtSetor,chtDataSolicita,chtHoraSolicita"

Edit.FormList "chtSolicitante,chtSetor,chtDataSolicita"

Edit.FormFind "chtSolicitante,chtSetor,chtDataSolicita"

Edit.AddReport "chtSetor,#1(l=Nº,gc)", "Por Setor", conListSetor
Edit.AddReport "chtSolicitante,#1(l=Nº,gc)", "Por Usuário",
conListUsuario
```

```

Edit.AddField "chtCodigo",      "Código",  efFldTypeText, , , efNext
Edit.AddField "chtSolicitante", "Solicitante"
Edit.AddField "chtSetor",      "Setor"
Edit.AddField "chtDataSolicita", "Data", efFldTypeText, efValOptDate,
, _
String.LongDate(Now)
Edit.AddField "chtHoraSolicita", "Horário", efFldTypeText, , , _
String.LeadingZeroes(Hour(Now), 2) & ":" &
String.LeadingZeroes(Minute(Now), 2)

Edit.FieldInternalLink "chtSolicitante", sparEditWhat, "chtCodigo"

Edit.EndForm

```

Aplica-se

EditForm

Veja também

BeginForm, EndForm, FormUnit, FormList

Método FormList

Descrição

Determina as as colunas que serão utilizadas no sistema de listagem padrão do EditForm.

Sintaxe

objeto.FormList *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. Lista sequencial com as identificações das colunas separadas por vírgulas.

Comentários

O código abaixo ilustra o uso do método **FormList**.

```

Edit.BeginForm "sncChamaTecnico", "Chamado Técnico", 1, sparEditWhat, _
efValLocClient

Edit.FormUnit
"chtSolicitante=Setor;chtSetor,chtDataSolicita,chtHoraSolicita"

Edit.FormList "chtSolicitante,chtSetor,chtDataSolicita"

Edit.FormFind "chtSolicitante,chtSetor,chtDataSolicita"

Edit.AddReport "chtSetor,#1(l=Nº,gc)", "Por Setor", conListSetor
Edit.AddReport "chtSolicitante,#1(l=Nº,gc)", "Por Usuário",
conListUsuario

Edit.AddField "chtCodigo",      "Código",  efFldTypeText, , , efNext
Edit.AddField "chtSolicitante", "Solicitante"
Edit.AddField "chtSetor",      "Setor"

```

```
Edit.AddField "chtDataSolicita", "Data", efFldTypeText, efValOptDate,
' -
String.LongDate(Now)
Edit.AddField "chtHoraSolicita", "Horário", efFldTypeText, , , _
String.LeadngZeroes(Hour(Now), 2) & ":" &
String.LeadngZeroes(Minute(Now), 2)

Edit.FieldInternalLink "chtSolicitante", sparEditWhat, "chtCodigo"

Edit.EndForm
```

Aplica-se

EditForm

Veja também

BeginForm. EndForm, FormUnit, FormFind

Método FormUnit

Descrição

Determina o forma que as colunas que serão utilizadas na formatação da tela do formulário ou texto HTML que será apresentado.

Sintaxe*objeto*.FormUnit *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. Lista sequencial com as identificações das colunas separadas por vírgulas, ponto e vírgulas, travessões, sinais de igual e textos diversos.

Comentários

Sintaxe de uso da Lista sequencial com as identificações das colunas:

conteúdo = [nome da coluna]

conteúdo + separador + conteúdo

Caracteres Formataadores (separadores):

Caracter	Resultado obtido
" , "	" <i>vírgula</i> ", quebra de colunas. Continua na mesma linha
" . "	" <i>ponto e vírgula</i> ", quebra de linha. Salta para a próxima linha.
" - "	" <i>travessão</i> ",Linha separadora. Separa a tela com uma linha.
" = "	" <i>igual</i> ", Título separador (Ex.: coluna1=Título separador;coluna2).
" "	" <i>linha vertical</i> ", Separador dos Tabs. Vide .

O código abaixo ilustra o uso do método **FormUnit**.

```

Edit.BeginForm "sncChamaTecnico", "Chamado Técnico", 1, sparEditWhat,
- efValLocClient

Edit.FormUnit
"chtSolicitante=Setor;chtSetor,chtDataSolicita,chtHoraSolicita"

Edit.FormList "chtSolicitante,chtSetor,chtDataSolicita"

Edit.FormFind "chtSolicitante,chtSetor,chtDataSolicita"

Edit.AddReport "chtSetor,#1(l=Nº,gc)", "Por Setor", conListSetor
Edit.AddReport "chtSolicitante,#1(l=Nº,gc)", "Por Usuário",
conListUsuario

Edit.AddField "chtCodigo", "Código", efFldTypeText, , , efNext
Edit.AddField "chtSolicitante", "Solicitante"
Edit.AddField "chtSetor", "Setor"
Edit.AddField "chtDataSolicita", "Data", efFldTypeText, efValOptDate,
, -
String.LongDate(Now)
Edit.AddField "chtHoraSolicita", "Horário", efFldTypeText, , , _
String.LeadingZeroes(Hour(Now), 2) & ":" &
String.LeadingZeroes(Minute(Now), 2)

Edit.FieldInternalLink "chtSolicitante", sparEditWhat, "chtCodigo"

Edit.EndForm

```

Aplica-se

EditForm

Veja também

BeginForm, EndForm, FormUnit, FormFind

Método FormTabs

Descrição

Define os rótulos apresentados nos *tabs* definidos no FormUnit utilizando o separador "|".

Sintaxe

objeto.**FormTabs** *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. Lista dos títulos que serão utilizados para os tabs, o separador poderá ser o " ".

Comentários

O código abaixo ilustra o uso do método **FormTabs**.

```
Edit.BeginForm "sncChamaTecnico", "Chamado Técnico", 1, sparEditWhat,
-
efValLocClient

Edit.FormUnit
"chtSolicitante|chtSetor,chtDataSolicita,chtHoraSolicita"

Edit.FormTabs "Solicitante|Setor"

Edit.FormList "chtSolicitante,chtSetor,chtDataSolicita"

Edit.FormFind "chtSolicitante,chtSetor,chtDataSolicita"

Edit.AddReport "chtSetor,#1(l=Nº,gc)", "Por Setor", conListSetor
Edit.AddReport "chtSolicitante,#1(l=Nº,gc)", "Por Usuário",
conListUsuario

Edit.AddField "chtCodigo", "Código", efFldTypeText, , , efNext
Edit.AddField "chtSolicitante", "Solicitante"
Edit.AddField "chtSetor", "Setor"
Edit.AddField "chtDataSolicita", "Data", efFldTypeText, efValOptDate,
'
-
String.LongDate(Now)
Edit.AddField "chtHoraSolicita", "Horário", efFldTypeText, , , _
String.LeadingZeroes(Hour(Now), 2) & ":" &
String.LeadingZeroes(Minute(Now), 2)

Edit.FieldInternalLink "chtSolicitante", sparEditWhat, "chtCodigo"

Edit.EndForm
```

Aplica-se

EditForm

Veja também

Método AddField

Descrição

Define cada campo que será utilizado para edição dos dados. Poderá se definir os mais diversos tipos de entradas e validações, relacionamentos (lookups), valores padrão para inclusão, listas de valores fixos, máscaras de apresentação.

Sintaxe

objeto.**AddField** *String1*, *String2*, [*Integer1*], [*Integer2*], [*Integer3*], [*String3*], [*String4*]

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String1</i>	Requerido. Nome da coluna ou tabela.coluna que será utilizada para edição.
<i>String2</i>	Requerido. Texto ou rótulo que será apresentado para identificar o campo durante a edição. Utilize o "&" para

	utilizar os atalhos usando <i>ALT+Letra</i> – AccessKey. Ex.: “&rótulo” utilize alt+r.
<i>Integer1</i>	Opcional. Tipo do campo que será utilizado. O padrão é efFldTypeText. Utilize as constantes efFldType*.
<i>Integer2</i>	Opcional. Tipo de validação para o campo utilizado. O padrão efValOptNone. Utilize as constantes efValOpt...
<i>Integer3</i>	Opcional. Opção de apresentação. O padrão é zero (0).
<i>String3</i>	Opcional. Valor padrão para este campo durante a inclusão.
<i>String4</i>	Opcional. Texto da mensagem que será apresentada quando o campo estiver vazio.

Comentários

O código abaixo ilustra o uso do método **AddField**.

```

Edit.BeginForm "sncChamaTecnico", "Chamado Técnico", 1, sparEditWhat,
-
efValLocClient

Edit.FormUnit
"chtSolicitante|chtSetor,chtDataSolicita,chtHoraSolicita"

Edit.FormTabs "Solicitante|Setor"

Edit.FormList "chtSolicitante,chtSetor,chtDataSolicita"

Edit.FormFind "chtSolicitante,chtSetor,chtDataSolicita"

Edit.AddReport "chtSetor,#1(l=Nº,gc)", "Por Setor", conListSetor
Edit.AddReport "chtSolicitante,#1(l=Nº,gc)", "Por Usuário",
conListUsuario

Edit.AddField "chtCodigo", "&Código", efFldTypeText, , , efNext
Edit.AddField "chtSolicitante", "&Solicitante"
Edit.AddField "chtSetor", "S&etor"
Edit.AddField "chtDataSolicita", "&Data", efFldTypeText, efValOptDate,
, -
String.LongDate(Now)
Edit.AddField "chtHoraSolicita", "&Horário", efFldTypeText, , , -
String.LeadingZeroes(Hour(Now), 2) & ":" &
String.LeadingZeroes(Minute(Now), 2)

Edit.FieldInternalLink "chtSolicitante", sparEditWhat, "chtCodigo"

Edit.EndForm

```

Aplica-se

EditForm

Veja também

DataAddField, BeginForm, EndForm, FormUnit, FormList, FormFind, “Constantes Field Types”

Método AddOrder

Descrição

Adiciona uma forma de ordenação à tabela apresentada pelo formulário definido. Deve ser usado no escopo do BeginForm, EndForm

Sintaxe

objeto.AddOrder String

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. Nome do campo definido com o DataAddField com qual se ordena a listagem.

Comentários

O código abaixo ilustra o uso do método **AddOrder**.

```
Edit.BeginForm "sncChamaTecnico", "Chamado Técnico", 1, sparEditWhat, _
    efValLocClient

Edit.FormUnit "chtSolicitante|chtSetor,chtDataSolicita,chtHoraSolicita"

Edit.FormTabs "Solicitante|Setor"

Edit.FormList "chtSolicitante,chtSetor,chtDataSolicita"

Edit.FormFind "chtSolicitante,chtSetor,chtDataSolicita"

Edit.AddReport "chtSetor,#1(l=Nº,gc)", "Por Setor", conListSetor
Edit.AddReport "chtSolicitante,#1(l=Nº,gc)", "Por Usuário",
conListUsuario

Edit.AddField "chtCodigo", "&Código", efFldTypeText, , , efNext
Edit.AddField "chtSolicitante", "&Solicitante"
Edit.AddField "chtSetor", "S&etor"
Edit.AddField "chtDataSolicita", "&Data", efFldTypeText, efValOptDate, ,
_
String.LongDate(Now)
Edit.AddField "chtHoraSolicita", "&Horário", efFldTypeText, , , _
String.LeadingZeroes(Hour(Now), 2) & ":" &
String.LeadingZeroes(Minute(Now), 2)

Edit.FieldInternalLink "chtSolicitante", sparEditWhat, "chtCodigo"

Edit.AddOrder "chtCodigo"
Edit.AddOrder "chtSetor"

Edit.EndForm
```

Aplica-se

EditForm

Vea também

BeginForm, EndForm, ListSetOfOrdersCaption, ShowSetOfOrdersLink

Método AddFilter

Descrição

Adiciona um filtro à apresentação (em listagem, e na navegação) da tabela, este filtro funciona internamente diferente do filtro passado via QueryString com o parâmetro efQueryStrFilter.

Sintaxe

objeto.AddFilter *String1*, *String2*, [*String3*]

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String1</i>	Requerido. Nome da coluna que será aplicada o filtro
<i>String2</i>	Requerido. Valor que será utilizado no filtro.
<i>String3</i>	Opcional. String contendo o operador a ser utilizado. Ex.: "<>", "LIKE". O padrão é "=".

Comentários

O código abaixo ilustra o uso do método **AddFilter**.

```
Edit.AddFilter "chtCodigo", "5"
Edit.AddFilter "chtCodigo", "1"
Edit.AddFilter "chtCodigo", "25", "<>"
```

Aplica-se

EditForm

Veja também

BeginForm, EndForm, "Constantes QueryString Parameters – efQueryStrFilter"

Método AddHeader

Descrição

Adiciona um super-título aos cabeçalhos das listagens e relatórios.

Sintaxe

objeto.AddHeader *String1*, *Integer1*, *Integer2*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. Texto do título que será apresentado no título.
<i>Integer1</i>	Requerido. Coluna inicial da listagem que o cabeçalho será utilizado.
<i>Integer2</i>	Requerido. Tamanho em colunas do título.

Comentários

O código abaixo ilustra o uso do método **AddHeader**.

```
Edit.BeginForm "sncChamaTecnico", "Chamado Técnico", 1, sparEditWhat, _
    efValLocClient

Edit.FormUnit "chtSolicitante|chtSetor,chtDataSolicita,chtHoraSolicita"

Edit.FormTabs "Solicitante|Setor"

Edit.FormList "chtSolicitante,chtSetor,chtDataSolicita"

Edit.AddField "chtCodigo",          "&Código",  efFldTypeText, , , efNext
Edit.AddField "chtSolicitante",    "&Solicitante"
Edit.AddField "chtSetor",          "S&etor"
Edit.AddField "chtDataSolicita",   "&Data", efFldTypeText, efValOptDate, ,
_
String.LongDate(Now)
Edit.AddField "chtHoraSolicita",   "&Horário", efFldTypeText, , , _
String.LeadingZeroes(Hour(Now), 2) & ":" &
String.LeadingZeroes(Minute(Now), 2)

Edit.AddHeader "Chamado Técnico especial", 2, 50

Edit.EndForm
```

Aplica-se

EditForm

Veja também

BeginForm, FormList

Método AddReport

Descrição

Adiciona um relatório com diversas possibilidades de relatórios, com gráficos agrupados, soma dos valores, contagem de linhas e campos etc.

Sintaxe

objeto.**AddReport** *String1*, *String2*, *String3*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String1</i>	Requerido. Conteúdo "programável" do relatório utilizando as regras descritas abaixo.
<i>String2</i>	Requerido. Título do relatório.
<i>String3</i>	Requerido. Identificador do relatório que será utilizado nas QueryString.

Comentários

Sintaxe para o segundo parâmetro *String1*:

[Campo ([Parametros_do_campo])] [<separador> Campo (...) [, ...]]

separador = "=" ou "," ou ";" ou "-"

Campo = [Nome_do_campo] | [# Campo_Calculado]

Nome_do_campo é o nome de um campo existente na tabela.

Campo_Calculado é o nome do campo calculado (criado a partir de uma fórmula – veja Parâmetros_do_campo).

Parametros_do_campo:

Parametro	Descrição
"s"	Soma dos valores do campo.
"c"	Contagem dos registros do agrupamento.
"g"	Com gráfico.
"gs"	Gráfico de soma.
"gc"	Gráfico de contagem.
"a"	Campo agrupador.
"t"	Totalizado na mundaça do valor deste campo.
"f"	Tipo de dados de um campo calculado.
"l"	Rótulo. Utilize um <i>separador</i> para identificar o rótulo. Ex.: "l;Rótulo" ou "l=Rótulo"
"o"	Operação.
"v"	Valor da primeira linha em um contador montado pela operação.

O código abaixo ilustra o uso do método **AddReport**.

```

Edit.BeginForm "sncChamaTecnico", "Chamado Técnico", 1, sparEditWhat, _
    efValLocClient

Edit.FormUnit "chtSolicitante|chtSetor,chtDataSolicita,chtHoraSolicita"

Edit.FormTabs "Solicitante|Setor"

Edit.FormList "chtSolicitante,chtSetor,chtDataSolicita"

Edit.FormFind "chtSolicitante,chtSetor,chtDataSolicita"

Edit.AddReport "chtSetor,#l(l=Nº,gc)", "Por Setor", conListSetor
Edit.AddReport "chtSolicitante,#l(l=Nº,gc)", "Por Usuário",
conListUsuario
Edit.AddReport "chtSetor,#chtSetor(l=Soma,gs)", "Por Usuário",
conListSoma

Edit.AddField "chtCodigo", "&Código", efFldTypeText, , , efNext

```

Objeto EditForm

```
Edit.AddField "chtSolicitante", "&Solicitante"
Edit.AddField "chtSetor", "S&etor"
Edit.AddField "chtDataSolicita", "&Data", efFldTypeText, efValOptDate, ,
_ String.LongDate(Now)

Edit.AddField "chtHoraSolicita", "&Horário", efFldTypeText, , , _
String.LeadingZeroes(Hour(Now), 2) & ":" &
String.LeadingZeroes(Minute(Now), 2)

Edit.FieldInternalLink "chtSolicitante", sparEditWhat, "chtCodigo"

Edit.EndForm
```

Aplica-se

EditForm

Veja também

AddField, DataAddField, ListReportsCaption, ShowReportsLink

Método EmptyListMessage

Descrição

Define uma mensagem a ser apresentada no caso da listagem não conter registros.

Sintaxe

objeto.**EmptyListMessage** *String*, [*Boolean*]

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. Mensagem a ser apresentada caso não existam registros na listagem.
<i>Boolean</i>	Opcional. Quando for verdadeiro faz com que se apresente um link para a inclusão de um registro.

Comentários

O código abaixo ilustra o uso do método **EmptyListMessage**.

```
Edit.EmptyListMessage "Não há registros a serem apresentados para as opções  
escolhidas", True
```

Aplica-se

EditForm

Veja também

BeginForm, FormList

Método FieldAlphabeticIndex

Descrição

Define o tipo de índice alfabético apresentado na listagem.

Sintaxe

objeto.**FieldAlphabeticIndex** *String*, [*Integer*]

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. Mensagem a ser apresentada caso não existam registros na listagem.
<i>Integer</i>	Opcional. O tipo de índice alfabético apresentado, somente numérico, alfanumérico, somente alfabético.

Comentários

O código abaixo ilustra o uso do método **FieldAlphabeticIndex**.

```
Edit.AddField "chtCodigo", "&Codigo"
Edit.FieldAlphabeticIndex "chtCodigo", efAlphaIndexNumeric
```

Aplica-se

EditForm

Veja também

"Constantes Alphabetic Index types", AlphabeticIndex, FormList

Método FieldTextAreaHeight

Descrição

Define a altura de um campo do tipo *TextArea*, definido com o AddField com a constante efFldTypeTextArea.

Sintaxe

objeto.FieldTextAreaHeight *String*, *Integer*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. Nome do campo já criado utilizando o AddField.
<i>Integer</i>	Requerido. Número de linhas que será fixado apresentação da altura do campo.

Comentários

O código abaixo ilustra o uso do método **FieldTextAreaHeight**.

```
Edit.BeginForm "sncChamaTecnico", "Chamado Técnico", 1, sparEditWhat,
- efValLocClient

Edit.FormUnit
"chtSolicitante|chtSetor,chtDataSolicita,chtHoraSolicita" & _
"-chtDescricao"

Edit.FormTabs "Solicitante|Setor"

Edit.FormList "chtSolicitante,chtSetor,chtDataSolicita"
```

```

Edit.FormFind "chtSolicitante,chtSetor,chtDataSolicita"

Edit.AddReport "chtSetor,#1(l=Nº,gc)", "Por Setor", conListSetor
Edit.AddReport "chtSolicitante,#1(l=Nº,gc)", "Por Usuário",
conListUsuario

Edit.AddField "chtCodigo", "&Código", efFldTypeText, , , efNext
Edit.AddField "chtSolicitante", "&Solicitante"
Edit.AddField "chtSetor", "S&etor"
Edit.AddField "chtDataSolicita", "&Data", efFldTypeText, efValOptDate,
' _
String.LongDate(Now)
Edit.AddField "chtHoraSolicita", "&Horário", efFldTypeText, , , _
String.LeadingZeroes(Hour(Now), 2) & ":" &
String.LeadingZeroes(Minute(Now), 2)

Edit.AddField "chtDescricao", "Descr&icao", efFldTypeTextArea
Edit.FieldTextAreaHeight "chtDescricao", 3
Edit.FieldShowSize "chtDescricao", 40

Edit.FieldInternalLink "chtSolicitante", sparEditWhat, "chtCodigo"

Edit.EndForm

```

Aplica-se

EditForm

Veja também

AddField, “Constantes Field Types”

Método FieldImage

Descrição

Define um campo imagem que servirá para enviar imagens diretamente da tela do formulário e visualizado de forma fácil.

Sintaxe

objeto.FieldImage *String1, String2, String3*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String1</i>	Requerido. Nome do campo já criado utilizando o AddField.
<i>String2</i>	Requerido. Diretório onde serão gravadas as imagens enviadas.
<i>String3</i>	Requerido. Prefixo do nome dos arquivos das imagens. Será acrescido ao nome da imagem o código do registro – a chave primária definida.

Comentários

O código abaixo ilustra o uso do método **FieldImage**.


```

Edit.BeginForm "sncChamaTecnico", "Chamado Técnico", 1, sparEditWhat,
-
efValLocClient

Edit.FormUnit
"chtSolicitante|chtSetor,chtDataSolicita,chtHoraSolicita" & _
"-chtImagem"

Edit.FormTabs "Solicitante|Setor"

Edit.FormList "chtSolicitante,chtSetor,chtDataSolicita"

Edit.FormFind "chtSolicitante,chtSetor,chtDataSolicita"

Edit.AddReport "chtSetor,#1(l=Nº,gc)", "Por Setor", conListSetor
Edit.AddReport "chtSolicitante,#1(l=Nº,gc)", "Por Usuário",
conListUsuario

Edit.AddField "chtCodigo", "&Código", efFldTypeText, , , efNext
Edit.AddField "chtSolicitante", "&Solicitante"
Edit.AddField "chtSetor", "Ssetor"
Edit.AddField "chtDataSolicita", "&Data", efFldTypeText, efValOptDate,
' -
String.LongDate(Now)
Edit.AddField "chtHoraSolicita", "&Horário", efFldTypeText, , , _
String.LeadingZeroes(Hour(Now), 2) & ":" &
String.LeadingZeroes(Minute(Now), 2)

Edit.AddField " chtImagem ", "&Imagem", efFldTypeImage
Edit.FieldImage "chtImagem", "/img", "img"

Edit.FieldInternalLink "chtSolicitante", sparEditWhat, "chtCodigo"

Edit.EndForm

```

Aplica-se

EditForm

Veja também

AddField, DataAddField, FieldFile, “Constantes Field Types”

Método FieldRadioColumns

Descrição

Define o número de colunas de um campo Rádio. Utiliza-se este método para se personalizar a apresentação de *RadioBoxes* no EditForm, pois apresenta-se estes organizados em uma tabela – veja objeto Table.

Sintaxe

objeto.FieldRadioColumns *String*, *Integer*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. Nome do campo já criado utilizando o AddField

	e o tipo <code>efFldTypeRadio</code> .
<i>Integer</i>	Requerido. Quantidade de colunas.

Comentários

O código abaixo ilustra o uso do método **FieldRadioColumns**.

```

Edit.BeginForm "sncChamaTecnico", "Chamado Técnico", 1, sparEditWhat,
-
efValLocClient

Edit.FormUnit
"chtSolicitante|chtSetor,chtDataSolicita,chtHoraSolicita" & _
"-chtTipodeProblema"

Edit.FormTabs "Solicitante|Setor"

Edit.FormList "chtSolicitante,chtSetor,chtDataSolicita"

Edit.FormFind "chtSolicitante,chtSetor,chtDataSolicita"

Edit.AddReport "chtSetor,#1(l=Nº,gc)", "Por Setor", conListSetor
Edit.AddReport "chtSolicitante,#1(l=Nº,gc)", "Por Usuário",
conListUsuario

Edit.AddField "chtCodigo", "&Código", efFldTypeText, , , efNext
Edit.AddField "chtSolicitante", "&Solicitante"
Edit.AddField "chtSetor", "S&setor"
Edit.AddField "chtDataSolicita", "&Data", efFldTypeText, efValOptDate,
' -
String.LongDate(Now)
Edit.AddField "chtHoraSolicita", "&Horário", efFldTypeText, , , _
String.LeadingZeroes(Hour(Now), 2) & ":" &
String.LeadingZeroes(Minute(Now), 2)

Edit.AddField "chtTipodeProblema", "&Tipo de Problema ",
efFldTypeRadio
Edit.FieldAddValue "chtTipodeProblema", 0, "Hardware"
Edit.FieldAddValue "chtTipodeProblema", 1, "Software"
Edit.FieldAddValue "chtTipodeProblema", 2, "Outros"
Edit.FieldRadioColumns "chtTipodeProblema", 3

' fará com que fique um Radio em baixo do outro:

Edit.FieldRadioColumns "chtTipodeProblema", 1

Edit.FieldInternalLink "chtSolicitante", sparEditWhat, "chtCodigo"

Edit.EndForm

```

Aplica-se

EditForm

Veja também

AddField, DataAddField, "Constantes Field Types"

Método FieldCharCase

Descrição

Define a caixa dos caracteres do texto de um determinado campo.

Sintaxe

objeto.FieldCharCase *String*, *Integer*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. Nome do campo já criado utilizando o AddField.
<i>Integer</i>	Requerido. Constante que será utilizada. Utilize a constante do tipo efCharCase....

Comentários

O código abaixo ilustra o uso do método **FieldCharCase**.

```
Edit.BeginForm "sncChamaTecnico", "Chamado Técnico", 1, sparEditWhat,
-
efValLocClient

Edit.FormUnit
"chtSolicitante|chtSetor,chtDataSolicita,chtHoraSolicita"

Edit.FormTabs "Solicitante|Setor"

Edit.FormList "chtSolicitante,chtSetor,chtDataSolicita"

Edit.FormFind "chtSolicitante,chtSetor,chtDataSolicita"

Edit.AddReport "chtSetor,#1(l=Nº,gc)", "Por Setor", conListSetor
Edit.AddReport "chtSolicitante,#1(l=Nº,gc)", "Por Usuário",
conListUsuario

Edit.AddField "chtCodigo", "&Código", efFldTypeText, , , efNext
Edit.AddField "chtSolicitante", "&Solicitante"
Edit. FieldCharCase "chtSolicitante", efCharCaseUpper

Edit.AddField "chtSetor", "S&setor"
Edit.AddField "chtDataSolicita", "&Data", efFldTypeText, efValOptDate,
' -
String.LongDate(Now)
Edit.AddField "chtHoraSolicita", "&Horário", efFldTypeText, , , _
String.LeadingZeroes(Hour(Now), 2) & ":" &
String.LeadingZeroes(Minute(Now), 2)

Edit.FieldInternalLink "chtSolicitante", sparEditWhat, "chtCodigo"

Edit.EndForm
```

Aplica-se

EditForm

Veja também

AddField, DataAddField, "Constantes Character Case"

Método FieldDisableValue

Descrição

Define as formas de desabilitação de um campo. Utiliza-se para fazer com que os campos se habilitem ou desabilitem automaticamente de acordo com estados/valores entrados pelo usuário no formulário.

Sintaxe*objeto*.**FieldDisableValue** *String1, String2, String3, Boolean*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String1</i>	Requerido. Nome do campo já criado utilizando o AddField.
<i>String2</i>	Requerido. Nome do campo desabilitante já criado utilizando o AddField.
<i>String3</i>	Requerido. Valor de desabilitação que o campo desabilitante deve conter para desabilitar o campo especificado em <i>String1</i> .
<i>Boolean</i>	Requerido. Opção de abilitação ou desabilitação para o valor. De acordo com as constantes efCond...

Comentários

O código abaixo ilustra o uso do método **FieldDisableValue**.

```
Edit.BeginForm "sncChamaTecnico", "Chamado Técnico", 1, sparEditWhat,
-
efValLocClient

Edit.FormUnit
"chtSolicitante|chtSetor,chtDataSolicita,chtHoraSolicita"

Edit.FormTabs "Solicitante|Setor"

Edit.FormList "chtSolicitante,chtSetor,chtDataSolicita"

Edit.FormFind "chtSolicitante,chtSetor,chtDataSolicita"

Edit.AddReport "chtSetor,#1(l=Nº,gc)", "Por Setor", conListSetor
Edit.AddReport "chtSolicitante,#1(l=Nº,gc)", "Por Usuário",
conListUsuario

Edit.AddField "chtCodigo", "&Código", efFldTypeText, , , efNext
Edit.AddField "chtSolicitante", "&Solicitante"
Edit.AddField "chtSetor", "Ssetor"
Edit.AddField "chtDataSolicita", "&Data", efFldTypeText, efValOptDate,
' -
String.LongDate(Now)
```

```

Edit.AddField "chtHoraSolicita", "&Horário", efFldTypeText, , , _
String.LeadngZeroes(Hour(Now), 2) & ":" &
String.LeadngZeroes(Minute(Now), 2)

Edit.FieldInternalLink "chtSolicitante", sparEditWhat, "chtCodigo"

Edit.FieldDisableValue "chtDataSolicita", "chtSetor", 21, efCondEnable

'Se o valor de chtSetor for 21 então o campo chtDataSolicita será
abilitado.

Edit.EndForm

```

Aplica-se

EditForm

Vea também

AddField, DataAddField, "Constantes Field Disable Condition"

Método FieldListSize

Descrição

Define a largura máxima de um campo na listagem. Se o campo for mais longo que esta largura máxima então concatena-se reticências (...) ao final do campo, denotando continuidade.

Sintaxe

objeto.FieldListSize *String*, *Integer*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. Nome do campo já criado utilizando o AddField.
<i>Integer</i>	Requerido. Valor máximo para o tamanho do campo. Este valor é baseado em largura de Tabelas.

Comentários

O código abaixo ilustra o uso do método **FieldListSize**.

```

Edit.BeginForm "sncChamaTecnico", "Chamado Técnico", 1, sparEditWhat,
- efValLocClient

Edit.FormUnit
"chtSolicitante|chtSetor,chtDataSolicita,chtHoraSolicita"

Edit.FormTabs "Solicitante|Setor"

Edit.FormList "chtSolicitante,chtSetor,chtDataSolicita"

Edit.FormFind "chtSolicitante,chtSetor,chtDataSolicita"

Edit.AddReport "chtSetor,#1(l=Nº,gc)", "Por Setor", conListSetor

```

```
Edit.AddReport "chtSolicitante,#1(l=N°,gc)", "Por Usuário",
conListUsuario

Edit.AddField "chtCodigo",          "&Código",  efFldTypeText, , , efNext
Edit.AddField "chtSolicitante",    "&Solicitante"
Edit.FieldListSize "chtSolicitante", 30

Edit.AddField "chtSetor",           "S&setor"
Edit.AddField "chtDataSolicita",    "&Data",  efFldTypeText, efValOptDate,
' _
String.LongDate(Now)
Edit.AddField "chtHoraSolicita",    "&Horário", efFldTypeText, , , _
String.LeadingZeroes(Hour(Now), 2) & ":" &
String.LeadingZeroes(Minute(Now), 2)

Edit.FieldInternalLink "chtSolicitante", sparEditWhat, "chtCodigo"

Edit.EndForm
```

Aplica-se

EditForm

Veja também

AddField, DataAddField, FormList, FieldListChars, Table.RowWidth, Table.CellWidth

Método FieldListChars

Descrição

Define a largura máxima em caracteres de um campo na listagem.

Sintaxe*objeto*.FieldListChars *String*, *Integer*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. Nome do campo já criado utilizando o AddField.
<i>Integer</i>	Requerido. Valor máximo em caracteres para o tamanho do campo.

ComentáriosO código abaixo ilustra o uso do método **FieldListChars**.

```
Edit.BeginForm "sncChamaTecnico", "Chamado Técnico", 1, sparEditWhat,
_
efValLocClient

Edit.FormUnit
"chtSolicitante|chtSetor,chtDataSolicita,chtHoraSolicita"

Edit.FormTabs "Solicitante|Setor"

Edit.FormList "chtSolicitante,chtSetor,chtDataSolicita"
```

```

Edit.FormFind "chtSolicitante,chtSetor,chtDataSolicita"

Edit.AddReport "chtSetor,#1(l=Nº,gc)", "Por Setor", conListSetor
Edit.AddReport "chtSolicitante,#1(l=Nº,gc)", "Por Usuário",
conListUsuario

Edit.AddField "chtCodigo", "&Código", efFldTypeText, , , efNext
Edit.AddField "chtSolicitante", "&Solicitante"
Edit.FieldListChars "chtSolicitante", 30

Edit.AddField "chtSetor", "Ssetor"
Edit.AddField "chtDataSolicita", "&Data", efFldTypeText, efValOptDate,
, _
String.LongDate(Now)
Edit.AddField "chtHoraSolicita", "&Horário", efFldTypeText, , , _
String.LeadingZeroes(Hour(Now), 2) & ":" &
String.LeadingZeroes(Minute(Now), 2)

Edit.FieldInternalLink "chtSolicitante", sparEditWhat, "chtCodigo"

Edit.EndForm

```

Aplica-se

EditForm

Veja também

AddField, DataAddField, FormList, FieldListSize

Método FieldMask

Descrição

Define a máscara a ser aplicada em um campo durante as apresentações.

Sintaxe*objeto*.FieldMask *String1*, *String2*, [*String3*]

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String1</i>	Requerido. Nome do campo já criado utilizando o AddField.
<i>String2</i>	Requerido. Máscara que será utilizada na apresentação de um campo específico.
<i>String3</i>	Opcional. Complemento da máscara. É o que será apresentado caso não exista a posição definida – veja os comentários abaixo.

Comentários

A seguinte sintaxe se aplica à máscara:

Parte	Descrição
"0"	Valida esta posição como um número, se a posição estiver vazia ou inválida ele é preenchido com o complemento da máscara.
"A"	Valida esta posição como um caracter alfabético (entre A e Z, ou a e z), se a posição estiver vazia ou inválida ele é preenchido com o complemento da máscara
<i>literal</i>	Preenche esta posição com o que for passado.

O código abaixo ilustra o uso do método **FieldMask**.

```
Edit.AddField "chtCodigo", "&Código", efFldTypeText, , , efNext
Edit.FieldMask "chtCodigo", "00/00", "0"
```

Aplica-se

EditForm

Veja também

AddField, DataAddField, String.Mask

Método FieldShowSize

Descrição

Define o tamanho máximo de um campo a ser apresentado em um formulário – com *Inputs*.

Sintaxe

objeto.FieldShowSize *String*, *Integer*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. Nome do campo já criado utilizando o AddField.
<i>Integer</i>	Requerido. Número de linhas que será fixado apresentação da largura do campo.

Comentários

O código abaixo ilustra o uso do método **FieldShowSize**.

```
Edit.BeginForm "sncChamaTecnico", "Chamado Técnico", 1, sparEditWhat,
-
efValLocClient

Edit.FormUnit
"chtSolicitante|chtSetor,chtDataSolicita,chtHoraSolicita" & _
"-chtDescricao"

Edit.FormTabs "Solicitante|Setor"

Edit.FormList "chtSolicitante,chtSetor,chtDataSolicita"
```



```

Edit.FormFind "chtSolicitante,chtSetor,chtDataSolicita"

Edit.AddReport "chtSetor,#1(l=Nº,gc)", "Por Setor", conListSetor
Edit.AddReport "chtSolicitante,#1(l=Nº,gc)", "Por Usuário",
conListUsuario

Edit.AddField "chtCodigo", "&Código", efFldTypeText, , , efNext
Edit.AddField "chtSolicitante", "&Solicitante"
Edit.AddField "chtSetor", "&Setor"
Edit.AddField "chtDataSolicita", "&Data", efFldTypeText, efValOptDate,
, _
String.LongDate(Now)
Edit.AddField "chtHoraSolicita", "&Horário", efFldTypeText, , , _
String.LeadingZeroes(Hour(Now), 2) & ":" &
String.LeadingZeroes(Minute(Now), 2)

Edit.AddField "chtDescricao", "Descrição", efFldTypeTextArea
Edit.FieldTextAreaHeight "chtDescricao", 3
Edit.FieldShowSize "chtDescricao", 40

Edit.FieldInternalLink "chtSolicitante", sparEditWhat, "chtCodigo"

Edit.EndForm

```

Aplica-se

EditForm

Veja também

AddField, DataAddField, Form.FormInput

Método FieldSepDate

Descrição

Define os anos de início e de fim na apresentação de um campo de data separada –dia, mês, ano em campos separados. O campo deve ser criado com o AddField utilizando o tipo efFldTypeSeparateDate.

Sintaxe

objeto.FieldSepDate *String*, *Integer1*, *Integer2*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. Nome de campo criado utilizando o AddField.
<i>Integer1</i>	Requerido. Ano de início.
<i>Integer2</i>	Requerido. Ano de fim.

Comentários

O código abaixo ilustra o uso do método **FieldSepDate**.

```

Edit.AddField "chtDataSolicita", "&Data", efFldTypeSeparateDate, _
efValOptSepDate, , String.LongDate(Now)

```

Aplica-se

EditForm

Veja também

AddField, DataAddField, "Constantes Field Types"

Método FieldExternalLink

Descrição

Define que o campo especificado seja apresentado com um link "externo", isto é, com link que aponta para qualquer página fora do escopo do EditForm, ou dentro do escopo mais em outro *script*.

Sintaxe*objeto*.**FieldExternalLink** *String1*, *String2*, [*String3*], [*String4*]

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String1</i>	Requerido. Nome do campo criado utilizando o AddField.
<i>String2</i>	Requerido. O Link estático. Aquele que sempre vai se repetir.
<i>String3</i>	Opcional. Parâmetro, de QueryString, que receberá o valor do campo definido em <i>String4</i>
<i>String4</i>	Opcional. Campo que terá o valor apresentado na QueryString – no parâmetro definido em <i>String3</i> . Este campo deve ser criado utilizando o AddField.

Comentários

O código abaixo ilustra o uso do método **FieldExternalLink**.

```
Edit.AddField "chtSetor", "Setor"
Edit.AddField "chtSolicitante", "Solicitante"

'Apresentará um link no campo "Solicitante" para
http://localhost/outro.asp?O=15&T=7?Y=<valor do campo chtSetor naquele
momento>

Edit.FieldExternalLink "chtSolicitante",
"http://localhost/outro.asp?O=15&T=7", "Y", "chtSetor"
```

Aplica-se

EditForm

Veja também

AddField, DataAddField, URL.AddURL, URL.Equal

Método FieldInternalLink

Descrição

Define que o campo especificado seja apresentado com um link “interno”, isto é, com link que aponta para um formulário no escopo do EditForm.

Sintaxe

objeto.FieldInternalLink *String1, String2, String3, [Boolean]*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String1</i>	Requerido. Nome do campo criado utilizando o AddField.
<i>String2</i>	Requerido. <i>What</i> que define qual tabela/módulo o link vai apontar para.
<i>String3</i>	Requerido. Nome do campo que deve aparecer na QueryString criado pelo FieldInternalLink. Este aparece como parâmentor do efQueryStrFind.
<i>Boolean</i>	Opcional. Define se o destino do link será editável.

Comentários

O código abaixo ilustra o uso do método **FieldInternalLink**.

```
Edit.AddField "chtSolicitante", "Solicitante"
Edit.FieldInternalLink "chtSolicitante", Edit.parWhat, "chtCodigo"
```

Aplica-se

EditForm

Veja também

AddField, DataAddField, FieldExternalLink, FieldAddILTarget, “Constantes QueryString Parameters – efQueryStrEditableStr”, URL.Equal

Método FieldAddILTarget

Descrição

Adiciona um campo de ligação para o FieldInternalLink

Sintaxe

objeto.FieldAddILTarget *String1, String2*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String1</i>	Requerido. Nome do campo criado utilizando o AddField.
<i>String2</i>	Requerido. Nome do campo de ligação

Comentários

O código abaixo ilustra o uso do método **FieldAddILTarget**.

```
Edit.AddField "chtSolicitante", "Solicitante"
```

```
Edit.FieldInternalLink "chtSolicitante", Edit.parWhat, "chtCodigo"  
Edit.FieldAddILTarget "chtSolicitante", "chtSetor"
```

Aplica-se

EditForm

Veja também

AddField, DataAddField, FieldInternalLink

Método FieldAddValue

Descrição

Adiciona valores de apresentação a um campo do tipo efFldTypeSelect, efFldTypeRadio ou efFldTypeCheckBox.

Sintaxe

objeto.**FieldAddValue** *String1*, *Variante*, *String2*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String1</i>	Requerido. Nome do campo criado utilizando o AddField.
<i>Variante</i>	Requerido. Valor a ser adicionado. É o que será salvo no Banco de Dados.
<i>String2</i>	Requerido. Rótulo deste valor. Sempre será apresentado, na listagem, relatórios e formulários.

Comentários

O código abaixo ilustra o uso do método **FieldAddValue**.

```
Edit.AddField "chtTipodeProblema", "&Tipo de Problema ",  
efFldTypeRadio  
Edit.FieldAddValue "chtTipodeProblema", 0, "Hardware"  
Edit.FieldAddValue "chtTipodeProblema", 1, "Software"  
Edit.FieldAddValue "chtTipodeProblema", 2, "Outros"  
Edit.FieldRadioColumns "chtTipodeProblema", 3
```

Aplica-se

EditForm

Veja também

AddField, DataAddField, "Constantes Field Types"

Método FieldHint

Descrição

Define que o campo terá uma explicação básica para o seu preenchimento. Utiliza-se a propriedade HTML *TITLE* para se fazer com que quando se passe o mouse em cima do campo salte a explicação definida.

Sintaxe

objeto.**FieldHint** *String1*, *String2*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String1</i>	Requerido. Nome do campo criado utilizando o AddField.
<i>String2</i>	Requerido. Texto explicativo – o <i>hint</i> ou <i>tip</i> .

Comentários

O código abaixo ilustra o uso do método **FieldHint**.

```
Edit.AddField "chtTipodeProblema", "&Tipo de Problema ",
efFldTypeRadio

Edit.FieldAddValue "chtTipodeProblema", 0, "Hardware"
Edit.FieldAddValue "chtTipodeProblema", 1, "Software"
Edit.FieldAddValue "chtTipodeProblema", 2, "Outros"
Edit.FieldRadioColumns "chtTipodeProblema", 3

Edit.FieldHint "chtTipodeProblema", "Escolha aqui o tipo de problema a
ser reportado"
```

Aplica-se

EditForm

Veja também

AddField, DataAddField

Método FieldLookup

Descrição

Define que um campo seja apresentado como uma lista que se relaciona com outra tabela. Normalmente este campo será definido como uma chave estrangeira, com relacionamento de um para muitos. A idéia de *LookUp* é a de que este campo “procura” em outra tabela um campo mais nominal e menos numérico assim deixando o usuário livre da tarefa de ficar memorizando códigos.

Sintaxe

objeto.FieldLookup *String1*, *String2*, *String3*, *String4*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String1</i>	Requerido. <i>String1</i> é nome do campo criado utilizando o AddField.
<i>String2</i>	Requerido. Nome válido de uma tabela que será procurada e listada.
<i>String3</i>	Requerido. Nome do campo da tabela <i>String2</i> que terá o valor comparado com o valor do campo <i>String1</i> .
<i>String4</i>	Requerido. Nome do campo texto ou de qualquer outro tipo que será apresentado na lista ou <i>select</i> .

Comentários

O código abaixo ilustra o uso do método **FieldLookup**.

```
Edit.AddField "chtSetor", "S&etor", efFldTypeLookup  
Edit.FieldLookup "chtSetor", "chaSetores", "steCodigo", "steDescricao"
```

Aplica-se

EditForm

Veja também

AddField, DataAddField, DataAddRelation, DataAddRelationFields, FieldLookupFilter

Método FieldLookupFilter

Descrição

Define um filtro que limitará a lista gerada por um lookup. Este filtro pode ser fixo ou relacionado à outra tabela, definida com o FieldLookup.

Sintaxe

objeto.**FieldLookupFilter** *String1*, *String2*, [*String3*], [*String4*], [*String5*], [*String6*]

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String1</i>	Requerido. Nome do campo criado utilizando AddField.
<i>String2</i>	Requerido. Nome do campo de filtro que será comparado com um valor fixo (<i>String4</i>) ou de outro campo (<i>String3</i>) – opcionalmente de outra tabela.
<i>String3</i>	Opcional. Nome do campo a ser comparado com o campo de filtro (<i>String2</i>). Pode ser um campo da tabela de lookup definida no FieldLookup. Se este parâmetro for passado o próximo parâmetro (<i>String4</i>) é ignorado.
<i>String4</i>	Opcional. Valor a ser comparado ao campo de filtro (<i>String2</i>).
<i>String5</i>	Opcional. Tipo de dado do campo a ser comparado (<i>String3</i>) ou do valor a ser comparado (<i>String4</i>).

Comentários

O código abaixo ilustra o uso do método **FieldLookupFilter**.

```
Edit.AddField "chtSetor", "S&etor", efFldTypeLookup  
Edit.FieldLookup "chtSetor", "chaSetores", "steCodigo", "steDescricao"
```

```
'Aqui ele filtrará o Lookup definido acima, para os Setores que se  
'relacionam com clientes desta mesma tabela, no caso Chamados Técnicos.  
'Veja o exemplo no início.
```

```
Edit.FieldLookupFilter "chtSetor", "chtCliente", "steCliente"  
  
'ou
```

```
'Aqui ele filtrará o mesmo Lookup para os Setores que cliente de código  
' "20"
```

```
Edit.FieldLookupFilter "chtSetor", "chtCliente", , "20"
```

Aplica-se

EditForm

Veja também

FieldLookUp, AddField, DataAddField

Método FieldDoubleValidate

Descrição

Define uma validação que envolva dois campos de datas.

FieldDoubleValidate(ByVal strField As String, ByVal strField2 As String, ByVal strmessage As String)

Sintaxe

objeto.FieldDoubleValidate *String1*, *String2*, *String3*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String1</i>	Requerido. Nome do campo criado utilizando o AddField.
<i>String2</i>	Requerido. Nome do segundo campo também criado utilizando o AddField.
<i>String3</i>	Requerido. Mensagem que deve aparecer quando a validação for inválida.

Comentários

O código abaixo ilustra o uso do método **FieldDoubleValidate**.

```
Edit.AddField "chtDataSolicita", "&Primeira Data", efFldTypeText, _  
efValOptCompareDates, , String.LongDate(Now)  
Edit.AddField "chtDataSolicita2", "&Segunda Data", efFldTypeText, _  
efValOptDate, , String.LongDate(Now)  
Edit.FieldDoubleValidate "chtDataSolicita", "chtDataSolicita2", "Segunda  
Data deve ser menor que Primeira Data"
```

Aplica-se

EditForm

Veja também

AddField, DataAddField, “Constantes Validation Options”

Método FieldIsUnique

Descrição

Define que o campo será um campo de valor único, impedindo a repetição de valores no campo.

Sintaxe

objeto.FieldIsUnique *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. <i>String</i> é o nome do campo criado utilizando o AddField.

Comentários

O código abaixo ilustra o uso do método **FieldIsUnique**.

```
Edit.AddField "chtSolicitante", "Solicitante"  
Edit.FieldIsUnique "chtSolicitante"
```

Aplica-se

EditForm

Veja também

DataAddPrimaryKey, AddField

Método FieldIsUnique

Descrição

Define que o campo será um campo de valor único, impedindo a repetição de valores no campo.

Sintaxe

objeto.FieldIsUnique *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. Nome do campo criado utilizando o AddField.

Comentários

O código abaixo ilustra o uso do método **FieldIsUnique**.

```
Edit.AddField "chtSolicitante", "Solicitante"  
Edit.FieldIsUnique "chtSolicitante"
```

Aplica-se

EditForm

Veja também

DataAddPrimaryKey, AddField

Método CreateTable

Descrição

Cria uma tabela no Banco de Dados se já a estrutura estiver definida. Compatível com SQL Server e Access.

Sintaxe

objeto.**CreateTable** *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. Nome válido de uma tabela que será utilizada pelo EditForm.

Comentários

O código abaixo ilustra o uso do método **CreateTable**.

```

Edit.DataTable "sncChamaTecnico"

    Edit.DataAddField "chtCodigo",          efDataTypeInt,          "4",
efRequired
    Edit.DataAddField "chtSolicitante",      efDataTypeVarChar, "100",
efRequired
    Edit.DataAddField "chtSetor",           efDataTypeVarChar, "50",
efNotRequired
    Edit.DataAddField "chtDataSolicita",     efDataTypeDateTime, "10",
efNotRequired
    Edit.DataAddField "chtHoraSolicita",     efDataTypeVarChar, "5",
efNotRequired
    Edit.DataAddPrimaryKey "chtCodigo"

Edit.CreateTable "sncChamaTecnico"

```

Aplica-se

EditForm

Veja também

DataTable, DropTable, CreateIndex

Método DropTable

Descrição

Exclui uma Tabela do banco de dados. Compatível com Access e SQL Server.

Sintaxe

objeto.**DropTable** *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. Nome válido de uma tabela do Banco de Dados.

Comentários

O código abaixo ilustra o uso do método **DropTable**.

```
Edit.DropTable "sncChamaTecnico"
```

Aplica-se

EditForm

Veja também

CreateTable

Método CreateIndex

Descrição

Cria os índices de uma tabela. Estes já devem estar definidos com o método DataAddIndex.

Sintaxe

objeto.**CreateIndex** *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto EditForm.
<i>String</i>	Requerido. Nome válido de uma tabela do Banco de Dados.

Comentários

O código abaixo ilustra o uso do método **CreateIndex**.

```
Edit.DataTable "sncChamaTecnico"
```

```
    Edit.DataAddField "chtCodigo",          efDataTypeInt,          "4",  
    efRequired  
    Edit.DataAddField "chtSolicitante",    efDataTypeVarChar, "100",  
    efRequired  
    Edit.DataAddField "chtSetor",          efDataTypeVarChar,  "50",  
    efNotRequired  
    Edit.DataAddField "chtDataSolicita",   efDataTypeDateTime, "10",  
    efNotRequired  
    Edit.DataAddField "chtHoraSolicita",   efDataTypeVarChar,  "5",  
    efNotRequired  
    Edit.DataAddPrimaryKey "chtCodigo"  
  
    Edit.DataAddIndex "Solicitante", "chtSolicitante, chtSetor"  
    Edit.DataAddIndex "Setor", "chtSetor"  
  
    Edit.CreateIndex "sncChamaTecnico"
```

Aplica-se

EditForm

Veja também

CreateTable

Objeto File

Objeto File

Objeto utilizado para controle, gerenciamento e manipulação de arquivos. Os arquivos sempre são manipulados no servidor onde a aplicação estiver sendo executada.

Sintaxe

```
Set File = Server.CreateObject("Hiptools30.File")
```

Método Exists

Descrição

Verifica se o arquivo indicado existe.

Sintaxe

objeto. **Exists** *String*

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto File.
<i>String</i>	Requerido. <i>String</i> é nome válido de uma tabela que será utilizada pelo EditForm.

Comentários

O código abaixo ilustra o uso do método **Exists**.

```
Dim blnFileExists  
blnFileExists = File.Exists("C:\Inetpub\wwwroot\arquivo.asp")
```

Aplica-se

File

Veja também

Método Copy

Descrição

Copia um ou mais arquivos de uma localização no para outra.

Sintaxe

objeto. **Copy** *String1*, *String2*[, *Boolean*]

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto File.
<i>String1</i>	Requerido. <i>String1</i> é o path do arquivo origem.
<i>String2</i>	Requerido. <i>String2</i> é o path do arquivo destino.

<i>Boolean</i>	Requerido. <i>Boolean</i> permite Overwrite no arquivo, caso ele já exista.
----------------	---

Comentários

Se não for possível executar a cópia do

O código abaixo ilustra o uso do método **Copy**.

```
On Error Resume Next
```

```
File.Copy("C:\Inetpub\wwwroot\arquivo.asp", _  
         "C:\Inetpub\wwwroot\NovoArquivo.asp", True)
```

```
If Err.Number <> 0 Then  
    Show.MessageError Err.Description  
End If
```

Aplica-se

File

Veja também

Método ExtractFileName

Descrição

Retorna somente o nome do arquivo especificado sem o caminho.

Sintaxe

objeto.**ExtractFileName** *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto File.
<i>String</i>	Requerido. <i>String</i> é o caminho do arquivo.

Comentários

O código abaixo ilustra o uso do método **ExtractFileName**.

```
Dim strFileNameOnly  
strFileNameOnly = File.ExtractFileName("C:\Inetpub\wwwroot\arquivo.asp")
```

Aplica-se

File

Veja também

Método DateCreated

Descrição

Retorna a data de criação do arquivo.

Sintaxe

objeto.**DateCreated** *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto File.
<i>String</i>	Requerido. <i>String</i> é o path do arquivo.

Comentários

O código abaixo ilustra o uso do método **DateCreated**.

```
Dim vntDate  
vntDate = File.DateCreated("C:\Inetpub\wwwroot\arquivo.asp")
```

Aplica-se

File

Veja também

DateLastModified.

Método DateLastModified

Descrição

Retorna a data da última modificação do arquivo.

Sintaxe

objeto.**DateLastModified** *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto File.
<i>String</i>	Requerido. <i>String</i> é o path do arquivo.

Comentários

O código abaixo ilustra o uso do método **DateLastModified**.

```
Dim vntDate  
vntDate = File.DateLastModified("C:\Inetpub\wwwroot\arquivo.asp")
```

Aplica-se

File

Veja também

DateCreated.

Método Size

Descrição

Retorna o tamanho do arquivo.

Sintaxe*objeto*.**Size** *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto File.
<i>String</i>	Requerido. <i>String</i> é o path do arquivo.

ComentáriosO código abaixo ilustra o uso do método **Size**.

```
Dim lngSize
```

```
lngSize = File.Size("C:\Inetpub\wwwroot\arquivo.asp")
```

Aplica-se

File

Veja também

Objeto Font

Objeto Font

Este objeto armazena informações sobre formatação de fonte e possibilita gerar o código HTML para formatação de fonte. Esse objeto também é usado internamente como propriedade de outros objetos.

Sintaxe

```
Set Font = Server.CreateObject("Hiptertools30.Font")
```

Propriedade Bold

Descrição

Especifica se o texto deve ser exibido em negrito.

Sintaxe

objeto. **Bold** [= *Boolean*]

O argumento objeto sempre será o objeto *Font*.

Comentários

O código abaixo ilustra o uso da propriedade **Bold**.

```
Font.Bold = True
```

Aplica-se

Font

Veja também

Propriedade Color

Descrição

Especifica a cor do texto.

Sintaxe

objeto. **Color** [= *String*]

O argumento objeto sempre será o objeto *Font*.

Comentários

Os valores válidos para essa propriedade são os mesmos do HTML, especificado como nome da cor ou valor RGB em hexadecimal.

O código abaixo ilustra o uso da propriedade **Color**.

```
Font.Color = "#000000"
```

Aplica-se

Font

Veja também

Propriedade Face

Descrição

Especifica o nome da fonte com a qual o texto deve ser exibido.

Sintaxe

objeto. **Face** [= *String*]

O argumento *objeto* sempre será o objeto *Font*.

Comentários

O código abaixo ilustra o uso da propriedade **Face**.

```
Font.Face = "Arial, Helvetica"
```

Aplica-se

Font

Veja também

Propriedade Italic

Descrição

Especifica se o texto deve ser exibido em itálico.

Sintaxe

objeto. **Italic** [= *Boolean*]

O argumento *objeto* sempre será o objeto *Font*.

Comentários

O código abaixo ilustra o uso da propriedade **Italic**.

```
Font.Italic = True
```

Aplica-se

Font

Veja também

Propriedade Size

Descrição

Especifica o tamanho da fonte com a qual o texto deve ser exibido.

Sintaxe

objeto. **Size** [= *Integer*]

O argumento *objeto* sempre será o objeto *Font*.

Comentários

O código abaixo ilustra o uso da propriedade **Size**.

```
Font.Size = 3
```

Aplica-se
Font

Veja também

Propriedade Underline

Descrição

Especifica se o texto deve ser exibido sublinhado.

Sintaxe

objeto. **Underline** [= *Boolean*]

O argumento *objeto* sempre será o objeto *Font*.

Comentários

O código abaixo ilustra o uso da propriedade **Underline**.

```
Font.Underline = True
```

Aplica-se
Font

Veja também

Método BeginFont

Descrição

Inicia uma seção de texto que receberá a formatação definida nas propriedades do objeto.

Sintaxe

objeto. **BeginFont**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto Font.

Comentários

O código abaixo ilustra o uso do método **BeginFont**.

```
Font.Face = "Arial, Helvetica"  
Font.Color = "#000000"  
Font.Size = 3  
Font.Bold = True  
Font.Italic = True  
Font.Underline = True  
Font.BeginFont
```

Retorna:

```
<FONT FACE="Arial, Helvetica" COLOR="#000000" SIZE=3><B><I><U>
```

Aplica-se
Font

Veja também

Método EndFont

Descrição

Finaliza a última seção de texto iniciada pelo método *BeginFont*.

Sintaxe

objeto. **EndFont**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto Font.

Comentários

O código abaixo ilustra o uso do método **EndFont**.

```
Font.EndFont
```

Retorna:

```
</B></I></U></FONT>
```

Aplica-se

Font

Veja também

Objeto Form

Objeto Form

Facilita a criação de formulários em páginas.

Sintaxe

```
Set Form = Server.CreateObject("HiperTools30.Form")
```

Propriedade AccessKey

Descrição

Especifica a tecla de acesso para os itens do formulário.

Sintaxe

```
objeto.AccessKey [= String]
```

O argumento *objeto* sempre será o objeto Form.

Comentários

O código abaixo ilustra o uso da propriedade **AccessKey**.

```
Form.AccessKey = "C"
```

Aplica-se

Form

Veja também

Propriedade Alt

Descrição

Especifica o texto alternativo dos itens do formulário.

Sintaxe

```
objeto.Alt [= String]
```

O argumento *objeto* sempre será o objeto Form.

Comentários

O código abaixo ilustra o uso da propriedade **Alt**.

```
Form.Alt = "Teste do ALT"
```

Aplica-se

Form

Veja também

Propriedade Class

Descrição

Especifica a classe de estilo para os itens do formulário.

Sintaxe

`objeto.Class` [= *String*]

O argumento *objeto* sempre será o objeto Form.

Comentários

O código abaixo ilustra o uso da propriedade **Class**.

```
Form.Class = "Classe"
```

Aplica-se

Form

Veja também

Propriedade Enable

Descrição

Especifica se os itens do formulário estarão habilitados para interação com o usuário. Essa propriedade é suportada somente no Internet Explorer 4.0 ou superior.

Sintaxe

`objeto.Enable` [= *Boolean*]

O argumento *objeto* sempre será o objeto Form.

Comentários

O código abaixo ilustra o uso da propriedade **Enable**.

```
Form.Enable = True
```

Aplica-se

Form

Veja também

Propriedade EncType

Descrição

Especifica o tipo de codificação do formulário.

Sintaxe

`objeto.EncType` [= *String*]

O argumento *objeto* sempre será o objeto Form.

Comentários

O código abaixo ilustra o uso da propriedade **EncType**.

```
String.EncType = "text/mime"
```

Aplica-se

Form

Veja também

Propriedade MaxLength

Descrição

Especifica o comprimento máximo, em caracteres, que pode ser digitado em um item do formulário do tipo *Text*.

Sintaxe

objeto.**MaxLength** [= *Integer*]

O argumento *objeto* sempre será o objeto Form.

Comentários

O código abaixo ilustra o uso da propriedade **MaxLength**.

```
Form.MaxLength = 30
```

Aplica-se

Form

Veja também

Propriedade onChange

Descrição

Especifica o código JavaScript a ser executado no evento *onChange* dos itens do formulário.

Sintaxe

objeto.**onChange** [= *String*]

O argumento *objeto* sempre será o objeto Form.

Comentários

O código abaixo ilustra o uso da propriedade **onChange**.

```
String.onChange = "ExecutaNaMudanca()"
```

Aplica-se

Form

Veja também

Propriedade onClick

Descrição

Especifica o código JavaScript a ser executado no evento *onClick* dos itens do formulário.

Sintaxe

objeto.**onClick** [= *String*]

O argumento *objeto* sempre será o objeto Form.

Comentários

O código abaixo ilustra o uso da propriedade **onClick**.

```
String.onClick = "ExecutaNoClique()"
```

Aplica-se

Form

Veja também

Propriedade ReadOnly

Descrição

Especifica se os itens do formulário estarão habilitados para o usuário alterar seu valor. Essa propriedade é suportada somente no Internet Explorer 4.0 ou superior.

Sintaxe

objeto.ReadOnly [= *String*]

O argumento *objeto* sempre será o objeto Form.

Comentários

O código abaixo ilustra o uso da propriedade **ReadOnly**.

```
Form.ReadOnly = True
```

Aplica-se

Form

Veja também

Propriedade Title

Descrição

Especifica o título para os itens do formulário. De acordo com a definição do HTML, esse valor não é exibido, mas pode aparecer como *ToolTip*.

Sintaxe

objeto.Title [= *String*]

O argumento *objeto* sempre será o objeto Form.

Comentários

O código abaixo ilustra o uso da propriedade **Title**.

```
Form.Title = "Teste do TITLE"
```

Aplica-se

Form

Veja também

Método BeginForm

Descrição

Inicia um formulário.

Sintaxe

objeto.**BeginForm** *String1* [, *String2*] [, *String3*] [, *String4*] [, *String5*]

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Form</i> .
<i>String1</i>	Requerido. Endereço para o qual o formulário será submetido.
<i>String2</i>	Opcional. Nome do formulário.
<i>String3</i>	Opcional. Método de submissão do formulário. O método padrão é <i>post</i> .
<i>String4</i>	Opcional. Nome da janela ou frame para o qual submeter o formulário.
<i>String5</i>	Opcional. Código JavaScript a ser executado imediatamente antes do formulário ser submetido.

Comentários

O código abaixo ilustra o uso do método **BeginForm**.

```
Form.BeginForm Initializer.ScriptName, "frmEnviar", , , "return  
ValidateField()"
Retorna: <FORM ACTION="Script.ASP" NAME="frmEnviar" METHOD="POST"  
onSubmit="return ValidateField()">
```

```
Form.Hidden "Campo Escondido", "Valor do campo"
```

```
Form.FormInput "SUBMIT", "bmtEnvia", "Emitir"
```

```
Form.Submit "bmtEnvia", "Emitir"
```

```
Form.Reset "Limpar"
```

```
Form.EndForm
```

Aplica-se

Form

Veja também

Método BeginSelect

Descrição

Inicia uma lista de seleção.

Sintaxe

objeto.**BeginSelect** *String* [, *Integer*, *Boolean*]

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Form</i> .
<i>String</i>	Requerido. Nome da lista de seleção.
<i>Integer</i>	Opcional. Número de itens exibidos na lista. Caso esse parâmetro não seja fornecido, a lista é exibida como <i>combo box</i> .
<i>Boolean</i>	Opcional. Especifica se é possível selecionar mais de um item. Essa propriedade se aplica somente se o número de itens for definido como um valor maior do que 1 (um).

Comentários

O código abaixo ilustra o uso do método **BeginSelect**.

```
Form.BeginForm Initializer.ScriptName, "frmEnviar", , , "return
ValidateField()"
```

```
Form.Hidden "Campo Escondido", "Valor do campo"
```

```
Form.BeginSelect "Select1", 10, True
Retorna: <SELECT NAME="Select1" SIZE=10 MULTIPLE>
```

```
Form.SelectOption "Valor1", "Select 1", True
Form.SelectOption "Valor2", "Select 2"
Form.SelectOption "Valor3", "Select 3"
Form.EndSelect
```

```
Form.Submit "bmtEnvia", "Emitir"
Form.Reset "Limpar"
```

```
Form.EndForm
```

Aplica-se

Form

Veja também

Método CheckBox

Descrição

Exibe uma caixa de marcação.

Sintaxe

objeto.**CheckBox** *String1, String2[, String3, Boolean]*

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Form</i> .
<i>String1</i>	Requerido. Nome da caixa de marcação.
<i>String2</i>	Requerido. Valor da caixa de marcação.
<i>String3</i>	Opcional. Especifica o alinhamento da caixa de marcação.
<i>Integer</i>	Opcional. Especifica se a caixa por padrão estará marcada.

Comentários

O código abaixo ilustra o uso do método **CheckBox**.

```
Form.BeginForm Initializer.ScriptName, "frmEnviar", , , "return  
ValidateField()"
```

```
Form.CheckBox "Nome", "Teste", "Right", True
```

```
Retorna: <INPUT TYPE="CHECKBOX" NAME="Nome" VALUE="Teste" ALIGN="Right"  
CHECKED>
```

```
Form.CheckBox "Nome2", "Teste"
```

```
Retorna: <INPUT TYPE="CHECKBOX" NAME="Nome2" VALUE="Teste">
```

```
Form.Submit "bmtEnvia", "Emitir"
```

```
Form.Reset "Limpar"
```

```
Form.EndForm
```

Aplica-se

Form

Veja também

Método EndForm

Descrição

Finaliza o formulário.

Sintaxe

objeto.**EndForm**

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Form</i> .

Comentários

O código abaixo ilustra o uso do método **EndForm**.

```
Form.BeginForm Initializer.ScriptName, "frmEnviar", , , "return
ValidateField()"
```

```
Form.Hidden "Campo Escondido", "Valor do campo"
```

```
Form.FormInput "SUBMIT", "bmtEnvia", "Emitir"
```

```
Retorna: <INPUT TYPE="SUBMIT" NAME="bmtEnvia" VALUE="Emitir">
```

```
Form.Submit "bmtEnvia", "Emitir"
```

```
Form.Reset "Limpar"
```

```
Form.EndForm
```

```
Retorna: </FORM>
```

Aplica-se

Form

Veja também

Método EndSelect

Descrição

Finaliza a lista de seleção.

Sintaxe

objeto.**EndSelect**

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Form</i> .

Comentários

O código abaixo ilustra o uso do método **EndSelect**.

```
Form.BeginForm Initializer.ScriptName, "frmEnviar", , , "return
ValidateField()"
```

```
Form.Hidden "Campo Escondido", "Valor do campo"
```

Objeto Form

```
Form.BeginSelect "Select1", 10, True
Form.SelectOption "Valor1", "Select 1", True
Form.SelectOption "Valor2", "Select 2"
Form.SelectOption "Valor3", "Select 3"
Form.EndSelect
Retorna: </SELECT>

Form.Submit "bmtEnvia", "Emitir"
Form.Reset "Limpar"

Form.EndForm
```

Aplica-se

Form

Veja também

Método File

Descrição

Exibe um campo para seleção de arquivo, geralmente usado para Upload.

Sintaxe

objeto.**File** *String*, *Integer1* [, *Integer2*]

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Form</i> .
<i>String</i>	Requerido. Nome do campo.
<i>Integer1</i>	Requerido. Comprimento do campo em caracteres.
<i>Integer2</i>	Opcional. Comprimento máximo do valor do campo, em caracteres.

Comentários

O código abaixo ilustra o uso do método **File**.

```
Form.BeginForm Initializer.ScriptName, "frmEnviar", , , "return
ValidateField()"

Form.File "Arquivo", 10, 20
Retorna: <INPUT TYPE="FILE" SIZE=10 MAXLENGHT=20>

Form.Submit "bmtEnvia", "Enviar Arquivo"
Form.Reset "Limpar"
```

`Form.EndForm`**Aplica-se**
Form**Veja também**

Método FormInput

Descrição

Exibe um item de inserção de dados. Por ser um método genérico, algumas propriedades não se aplicam a todos os tipos de itens.

Sintaxe

objeto.**FormInput** *String1* [, *String2*, *Integer1*, *String3*, *Integer2*, *String4*, *Boolean*, *String5*]

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Form</i> .
<i>String1</i>	Requerido. Tipo de item de inserção de dados.
<i>String2</i>	Requerido. Nome do item.
<i>Integer1</i>	Opcional. Tamanho do item.
<i>String3</i>	Opcional. Valor padrão do item.
<i>Integer2</i>	Opcional. Comprimento máximo do valor do item, em caracteres.
<i>String4</i>	Opcional. Alinhamento do item.
<i>Boolean</i>	Opcional. Especifica se o item estará marcado.
<i>String5</i>	Opcional. Arquivo de origem (fonte) do item.

Comentários

O código abaixo ilustra o uso do método **FormInput**.

```
Form.BeginForm Initializer.ScriptName, "frmEnviar", , , "return  
ValidateField()"
```

```
Form.Hidden "Campo Escondido", "Valor do campo"
```

```
Form.FormInput "SUBMIT", "bmtEnvia", "Emitir"
```

```
Retorna: <INPUT TYPE="SUBMIT" NAME="bmtEnvia" VALUE="Emitir">
```

```
Form.Submit "bmtEnvia", "Emitir"
```

```
Form.Reset "Limpar"
```

Objeto Form

Form.EndForm

Aplica-se
Form

Veja também

Método Hidden

Descrição

Inclui um campo escondido no formulário.

Sintaxe

objeto.Hidden *String1* [, *String2*]

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Form</i> .
<i>String1</i>	Requerido. Nome do campo.
<i>String2</i>	Requerido. Valor do campo.

Comentários

O código abaixo ilustra o uso do método **Hidden**.

```
Form.BeginForm Initializer.ScriptName, "frmEnviar", , , "return  
ValidateField()"
```

```
Form.Radio "Nome", "Teste", "Right", 1  
Form.Radio "Nome2", "Teste"
```

```
Form.Hidden "Campo Escondido", "Valor do campo"
```

```
Retorna: <INPUT TYPE="HIDDEN" NAME="Campo Escondido" VALUE="Valor do campo">
```

```
Form.Submit "bmtEnvia", "Emitir"  
Form.Reset "Limpar"
```

```
Form.EndForm
```

Aplica-se
Form

Veja também

Método Image

Descrição

Exibe uma imagem que, ao ser clicada, submete o formulário.

Sintaxe

objeto.**Image** *String1*, *String2* [, *String3*]

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Form</i> .
<i>String1</i>	Requerido. Nome da imagem.
<i>String2</i>	Requerido. Arquivo de origem da imagem.
<i>String3</i>	Opcional. Texto alternativo da imagem.

Comentários

O código abaixo ilustra o uso do método **Image**.

```
Form.BeginForm Initializer.ScriptName, "frmEnviar", , , "return
ValidateField() "
```

```
Form.Hidden "Campo Escondido", "Valor do campo"
```

```
Form.Image "Imagem", "/img/image.gif", "Id da Imagem"
```

```
Retorna: <INPUT TYPE="IMAGE" NAME="Imagem" SRC="/img/image/gif" ALT="Id da
Imagem">
```

```
Form.Submit "bmtEnvia", "Emitir"
```

```
Form.Reset "Limpar"
```

```
Form.EndForm
```

Aplica-se

Form

Veja também

Método Palette

Descrição

Exibe uma paleta de cores, permitindo que o usuário selecione uma cor cujo valor é submetido no formulário no formato RGB (#RRGGBB).

Sintaxe

objeto.**Palette** *String1* [, *String2*]

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Form</i> .
<i>String1</i>	Requerido. Nome do arquivo de imagem que será usado como ícone que, ao ser clicado, exibe a paleta de cores.
<i>String2</i>	Opcional. Cor padrão. Caso não seja especificada, assume o valor #FFFFFF.

Comentários

O código abaixo ilustra o uso do método **Palette**.

```
Dim strValor

strValor = "Valor2"
Form.BeginForm Initializer.ScriptName, "frmEnviar", , , "return
ValidateField()"

Form.Hidden "Campo Escondido", "Valor do campo"

Form.Palette "Palettel.gif", "#EA0096"

Form.Submit "bmtEnvia", "Emitir"
Form.Reset "Limpar"

Form.EndForm
```

Aplica-se
Form**Veja também**

Método Password

Descrição

Exibe um campo para digitação de senha. Os valores digitados não são exibidos na tela.

Sintaxe

objeto.**Password** *String*, *Integer1* [, *Integer2*]

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Form</i> .
<i>String</i>	Requerido. Nome do campo.
<i>Integer1</i>	Requerido. Comprimento do campo em caracteres.

<i>Integer2</i>	Opcional. Comprimento máximo do valor do campo, em caracteres.
-----------------	--

Comentários

O código abaixo ilustra o uso do método **Password**.

```
Form.BeginForm Initializer.ScriptName, "frmEnviar", , , "return
ValidateField()"

Form.Password "Senha", 10, 20
Retorna: <INPUT TYPE="PASSWORD" NAME="Senha" SIZE=10 MAXLENGTH=20>

Form.Submit "bmtEnvia", "Emitir"
Form.Reset "Limpar"

Form.EndForm
```

Aplica-se

Form

Veja também

Método Radio

Descrição

Exibe um item de um grupo de seleção. Apenas um item de um grupo de seleção pode estar selecionado num dado momento. Um grupo é constituído por todos os itens do tipo *Radio* que tenham o mesmo nome.

Sintaxe

objeto.**Radio** *String1*, *String2*[, *String3*, *Boolean*]

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Form</i> .
<i>String1</i>	Requerido. Nome do item do grupo.
<i>String2</i>	Requerido. Valor do item do grupo.
<i>String3</i>	Opcional. Alinhamento do item do grupo.
<i>Boolean</i>	Opcional. Especifica se o item estará selecionado. Apenas um item dentro de um grupo pode ser selecionado.

Comentários

O código abaixo ilustra o uso do método **Radio**.

Objeto Form

```
Form.BeginForm Initializer.ScriptName, "frmEnviar", , , "return  
ValidateField()"
```

```
Form.Radio "Nome", "Teste", "Right", True
```

```
Retorna: <INPUT TYPE="radio" NAME="Nome" VALUE="Teste" ALIGN="Right" CHECKED>
```

```
Form.Radio "Nome2", "Teste"
```

```
Retorna: <INPUT TYPE="radio" NAME="Nome2" VALUE="Teste">
```

```
Form.Submit "bmtEnvia", "Emitir"
```

```
Form.Reset "Limpar"
```

```
Form.EndForm
```

Aplica-se

Form

Veja também

Método Reset

Descrição

Exibe um botão que, ao ser clicado, redefine todos os campos do formulário aos seus valores padrão.

Sintaxe

objeto.**Reset** *String*

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Form</i> .
<i>String</i>	Requerido. Valor do botão. O valor será apresentado como texto no botão.

Comentários

O código abaixo ilustra o uso do método **Reset**.

```
Form.BeginForm Initializer.ScriptName, "frmEnviar", , , "return  
ValidateField()"
```

```
Form.Radio "Nome", "Teste", "Right", 1
```

```
Form.Radio "Nome2", "Teste"
```

```
Form.Submit "bmtEnvia", "Emitir"
```

```
Form.Reset "Limpar"
```

```
Retorna: <INPUT TYPE="RESET" VALUE="Limpar">
```

`Form.EndForm`**Aplica-se**
Form**Veja também**

Método SelectChooseOption

Descrição

Inclui uma opção numa lista de seleção, marcando a opção como padrão caso seu valor seja o mesmo de um parâmetro especificado.

Sintaxe

objeto.**SelectChooseOption** *Variant1* [, *String*, *Variant2*]

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Form</i> .
<i>Variant1</i>	Requerido. Valor da opção.
<i>String</i>	Opcional. Texto exibido na opção.
<i>Variant2</i>	Opcional. Valor a ser selecionado como padrão.

Comentários

O código abaixo ilustra o uso do método **SelectChooseOption**.

```
Dim strValor

strValor = "Valor2"
Form.BeginForm Initializer.ScriptName, "frmEnviar", , , "return
ValidateField()"

Form.Hidden "Campo Escondido", "Valor do campo"

Form.BeginSelect "Select1", 10, True
Form.SelectChooseOption "Valor1", "Select 1", strValor
Retorna: <OPTION VALUE="Valor1">Valor1

Form.SelectChooseOption "Valor2", "Select 2", strValor
Retorna: <OPTION VALUE="Valor2" SELECTED>Valor2

Form.SelectChooseOption "Valor3", "Select 3", strValor
Retorna: <OPTION VALUE="Valor3">Valor3

Form.EndSelect
```

```
Form.Submit "bmtEnvia", "Emitir"  
Form.Reset "Limpar"  
  
Form.EndForm
```

Aplica-se
Form

Veja também

Método SelectCountry

Descrição

Exibe uma lista de seleção contendo vários países do mundo.

Sintaxe

objeto.**SelectState** *String1* [, *Integer*, *Boolean*, *String2*]

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Form</i> .
<i>String1</i>	Requerido. Nome da lista de seleção.
<i>Integer</i>	Opcional. Número de itens exibidos na lista. Caso esse parâmetro não seja fornecido, a lista é exibida como <i>combo box</i> .
<i>Boolean</i>	Opcional. Especifica se é possível selecionar mais de um item. Essa propriedade se aplica somente se o número de itens for definido como um valor maior do que 1 (um).
<i>String2</i>	Opcional. Valor do país a ser selecionado como padrão.

Valor	Resultado
AF	Afeganistão
AL	Albânia
DZ	Argélia
AS	Samoa Americano
AD	Andorra
AO	Angola
AI	Anguilla
AQ	Antártica
AG	Antígua

AR	Argentina
AM	Armênia
AW	Aruba
AU	Austrália
AT	Áustria
AZ	Azerbeidjã
BS	Bahamas
BH	Bahrain
BD	Bangladesh
BB	Barbados
BE	Bélgica
BZ	Belize
BJ	Benin
BM	Ilhas Bermudas
BT	Butão
BO	Bolívia
BA	Bosnia e Herzegovina
BW	Botsuana
BV	Ilha de Bouvet
BR	Brasil
IO	Território Britânico no Oceano Índico
VG	Ilhas de Virgem britânicas
BN	Brunei
BG	Bulgária
BF	Burkina Faso
BU	Birmânia
BI	Burundi
BY	Byelorussian SSR
CA	Canadá
CT	Cantão e Ilhas de Enderbury
CV	Cabo Verde

KY	Ilhas de caimão
CF	República africana central
TD	Chade
CL	Chile
CN	China
CX	Ilha de Natal
CC	Ilhas de Cocos
CO	Colômbia
KM	Comores
CG	Congo
CK	Cozinhe Ilhas
CR	Costa Rica
HR	Croácia (Hrvatska)
CU	Cuba
CY	Chipre
CS	Tchecoslováquia
KH	Campuchea democrático
YD	Iêmen democrático
DK	Dinamarca
DJ	Djibouti
DM	Dominica
DO	República dominicana
NQ	Dronning Maud Terra
TP	Timor oriental
EC	Equador
EG	Egito
SV	El Salvador
GQ	Guiné equatorial
ER	Eritrea
EE	Estônia
ET	Etiópia

FO	Ilhas de Faeroe
FK	Ilhas de Falkland
FJ	Fiji
FI	Finlândia
FR	França
FX	França, Metropolitana
GF	Guiana Francesa
PF	Polinésia Francesa
TF	Territórios Meridionais franceses
GA	Gabão
GM	Gâmbia
GE	Geórgia
DE	Alemanha
GH	Gana
GI	Gibraltar
GR	Grécia
GL	Greenland
GD	Granada
GP	Guadalupe
GU	Guam
GT	Guatemala
GN	Guiné
GW	Guiné-bissau
GY	Guiana
HT	Haiti
HM	Ilhas Heard e Mc Donald
HN	Honduras
HK	Hong Kong
HU	Hungria
IS	Islândia
IN	Índia

ID	Indonésia
IR	Irã
IQ	Iraque
IE	Irlanda
IL	Israel
IT	Itália
CI	Costa do marfim
JM	Jamaica
JP	Japão
JT	Ilha de Johnston
JO	Jordan
KZ	Kazakstan
KE	Quênia
KI	Quiribati
KW	Kuwait
KG	Kyrgyzstan
LA	República Democrática de Lao
LV	Latvia
LB	Líbano
LS	Lesoto
LR	Libéria
LY	Líbia Jamahiria árabe
LI	Liechtenstein
LT	Lituânia
LU	Luxemburgo
MO	Macau
MK	Macedônia
MG	Madagáscar
MY	Malásia
MW	Malauí
MV	Maldivas

ML	Mali
MT	Malta
MQ	Martinica
MR	Mauritânia
MU	Maurício
MX	México
MI	Ilhas Midway
MC	Mônaco
MN	Mongólia
MS	Montserrat
MA	Marrocos
MZ	Moçambique
NA	Namíbia
NP	Nepal
NR	Nauru
NL	Países Baixos
AN	Países Baixos Antilhas
NT	Zona Neutra
NC	Calidônia novo
NZ	Nova Zelândia
NI	Nicarágua
NE	Níger
NG	Nigéria
NU	Niue
NF	Ilha de Norfolk
MP	Ilhas Mariana do norte
NO	Noruega
OM	Omã
PC	Ilhas Pacíficas
PK	Paquistão
PA	Panamá

PG	Papua-Nova Guiné
PY	Paraguai
KP	República da Coreia
PE	Peru
PH	Filipinas
PN	Ilha de Pitcairn
PL	Polónia
PT	Portugal
TW	Província da China Taiwan
PR	Porto Rico
QA	Qatar
KR	República da Coreia
RE	Reunião
RO	Romênia
RU	Federação russa
RW	Ruanda
GS	Ilhas S. Geórgia e S. Intercalle
LC	São Lúcia
VC	São Vincent/Grenadines
WS	Samoa
SM	San Marino
ST	São Tomé e Príncipe
SA	Arábia Saudita
SN	Senegal
SC	Seicheles
SL	Sierra Leone
SG	Cingapura
SB	Ilhas de Salomão
SO	Somália
ZA	África do Sul
ES	Espanha

LK	Sri Lanka
SH	St. Helena
KN	St. Kitts Nevis Anguilla
PM	St. Pierre e Miquelon
SD	Sudão
SR	Suriname
SJ	Ilhas Svalbard e Jan Mayen
SZ	Suazilândia
SE	Suécia
CH	Suíça
SY	Syran República árabe
TH	Tailândia
TG	Togo
TK	Tokelau
TO	Tonga
TT	Trinidad e Tobago
TN	Tunísia
TR	Turquia
TM	Turkmenistan
TC	Turcos e Ilhas de Caicos
TV	Tuvalu
UM	EUA Ilhas Periféricas Secundárias
UG	Uganda
UA	SSR ucraniano
AE	Emirados árabes unidos
GB	Reino Unido
CM	República unida de Cameroon
TZ	República unida de Tanzânia
US	Estados Unidos
PU	Estados Unidos Ilhas do Pacífico
VI	Estados Unidos Ilhas Virgem

HV	Volta superior
UY	Uruguai
UZ	Uzbekistan
VU	Vanuatu
VA	Vaticano
VE	Venezuela
VN	Vietnã
WK	Ilha Wake
WF	Wallis e Ilhas de Futuma
EH	Saara Ocidental
YE	Iêmen
YU	Iugoslávia
ZR	Zaire
ZM	Zâmbia
ZW	Zimbabwewe

Comentários

O código abaixo ilustra o uso do método **SelectCountry**.

```
Dim strValor
```

```
strValor = "Valor2"
```

```
Form.BeginForm Initializer.ScriptName, "frmEnviar", , , "return  
ValidateField()"
```

```
Form.Hidden "Campo Escondido", "Valor do campo"
```

```
Form.SelectCountry "Select1", 10, True, "AL"
```

```
Retorna: <SELECT NAME="Select1" SIZE=10 MULTIPLE>
```

```
<OPTION VALUE="AF">Feganistão
```

```
<OPTION VALUE="AL" SELECTED >Albânia
```

```
<OPTION VALUE="DZ" >Argélia
```

```
.
```

```
.
```

```
.
```

```
</SELECT>
```

```
Form.Submit "bmtEnvia", "Emitir"
```

```
Form.Reset "Limpar"
```

```
Form.EndForm
```

Aplica-se

Form

Veja também

Método SelectMonthName

Descrição

Exibe uma lista de seleção contendo os nomes dos meses por extenso.

Sintaxe

objeto.**SelectMonthName** *String* [, *Integer1*, *Boolean*, *Integer2*]

<i>Parte</i>	Descrição
<i>String</i>	Requerido. Nome da lista de seleção.
<i>Integer1</i>	Opcional. Número de itens exibidos na lista. Caso esse parâmetro não seja fornecido, a lista é exibida como <i>combo box</i> .
<i>Boolean</i>	Opcional. Especifica se é possível selecionar mais de um item. Essa propriedade se aplica somente se o número de itens for definido como um valor maior do que 1 (um).
<i>Integer2</i>	Opcional. Valor do mês a ser selecionado como padrão.

Valor	Resultado
1	Janeiro
2	Fevereiro
3	Março
4	Abril
5	Maio
6	Junho
7	Julho
8	Agosto
9	Setembro
10	Outubro

11	Novembro
12	Dezembro

Comentários

O código abaixo ilustra o uso do método **SelectMonthName**.

```
Dim strValor

strValor = "Valor2"
Form.BeginForm Initializer.ScriptName, "frmEnviar", , , "return
ValidateField()"

Form.Hidden "Campo Escondido", "Valor do campo"

Form.SelectMonthName "Select1", 10, True, 2
Retorna: <SELECT NAME="Select1" SIZE=10 MULTIPLE>
<OPTION VALUE=1>Janeiro
<OPTION VALUE=2 SELECTED>Fevereiro
.
.
.
</SELECT>

Form.Submit "bmtEnvia", "Emitir"
Form.Reset "Limpar"

Form.EndForm
```

Aplica-se

Form

Veja também

Método SelectOption

Descrição

Inclui uma opção numa lista de seleção.

Sintaxe

objeto.**SelectOption** *String1*, *String2* [, *Boolean*]

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Form</i> .

<i>Variant1</i>	Requerido. Valor da opção.
<i>String</i>	Opcional. Texto exibido na opção.
<i>Variant2</i>	Opcional. Especifica se a opção será selecionada como padrão.

Comentários

O código abaixo ilustra o uso do método **SelectOption**.

```
Form.BeginForm Initializer.ScriptName, "frmEnviar", , , "return  
ValidateField()"
```

```
Form.Hidden "Campo Escondido", "Valor do campo"
```

```
Form.BeginSelect "Select1", 10, True  
Form.SelectOption "Valor1", "Select 1", True  
Retorna: <OPTION VALUE="Valor1" SELECTED>Valor1
```

```
Form.SelectOption "Valor2", "Select 2"  
Retorna: <OPTION VALUE="Valor2">Valor2
```

```
Form.SelectOption "Valor3", "Select 3"  
Retorna: <OPTION VALUE="Valor3">Valor3
```

```
Form.EndSelect
```

```
Form.Submit "bmtEnvia", "Emitir"  
Form.Reset "Limpar"
```

```
Form.EndForm
```

Aplica-se

Form

Veja também

Método SelectState

Descrição

Exibe uma lista de seleção contendo todos os estados do Brasil.

Sintaxe*objeto.SelectState String1 [, Integer, Boolean, String2]*

<i>Parte</i>	Descrição
<i>String1</i>	Requerido. Nome da lista de seleção.
<i>Integer</i>	Opcional. Número de itens exibidos na lista. Caso esse parâmetro não seja fornecido, a lista é exibida como <i>combo box</i> .
<i>Boolean</i>	Opcional. Especifica se é possível selecionar mais de um item. Essa propriedade se aplica somente se o número de itens for definido como um valor maior do que 1 (um).
<i>String2</i>	Opcional. Valor do estado a ser selecionado como padrão.

Valor	Resultado
AC	Acre
AP	Amapá
AM	Amazonas
BA	Bahia
CE	Ceará
DF	Distrito Federal
ES	Espírito Santo
GO	Goiás
MA	Maranhão
MT	Mato Grosso
MS	Mato Grosso do Sul
MG	Minas Gerais
PA	Pará
PB	Paraíba
PR	Paraná
PE	Pernambuco
PI	Piauí
RJ	Rio de Janeiro
RN	Rio Grande do Norte

RS	Rio Grande do Sul
RO	Rondônia
RR	Roraima
SC	Santa Catarina
SP	São Paulo
SE	Sergipe
TO	Tocantins

Comentários

O código abaixo ilustra o uso do método **SelectState**.

```
Dim strValor

strValor = "Valor2"
Form.BeginForm Initializer.ScriptName, "frmEnviar", , , "return
ValidateField()"

Form.Hidden "Campo Escondido", "Valor do campo"

Form.SelectState "Select1", 10, True, "AM"
Retorna: <SELECT NAME="Select1" SIZE=10 MULTIPLE>
<OPTION VALUE="AC">Acre
<OPTION VALUE="AP">Amapá
<OPTION VALUE="AM" SELECTED>Amazonas
.
.
.
</SELECT>

Form.Submit "bmtEnvia", "Emitir"
Form.Reset "Limpar"

Form.EndForm
```

Aplica-se

Form

Veja também

Método SelectWeekDay

Descrição

Exibe uma lista de seleção contendo os dias da semana por extenso.

Sintaxe

objeto.**SelectWeekDay** *String* [, *Integer1*, *Boolean*, *Integer2*]

<i>Parte</i>	Descrição
<i>String</i>	Requerido. Nome da lista de seleção.
<i>Integer1</i>	Opcional. Número de itens exibidos na lista. Caso esse parâmetro não seja fornecido, a lista é exibida como <i>combo box</i> .
<i>Boolean</i>	Opcional. Especifica se é possível selecionar mais de um item. Essa propriedade se aplica somente se o número de itens for definido como um valor maior do que 1 (um).
<i>Integer2</i>	Opcional. Valor do dia da semana a ser selecionado como padrão.

Valor	Resultado
1	Domingo
2	Segunda-Feira
3	Terça-Feira
4	Quarta-Feira
5	Quinta-Feira
6	Sexta-Feira
7	Sábado

Comentários

O código abaixo ilustra o uso do método **SelectWeekDay**.

```
Dim strValor
```

```
strValor = "Valor2"
```

```
Form.BeginForm Initializer.ScriptName, "frmEnviar", , , "return  
ValidateField()"
```

```
Form.Hidden "Campo Escondido", "Valor do campo"
```



```

Form.SelectWeekDay "Select1", 10, True, 2
Retorna: <SELECT NAME="Select1" SIZE=10 MULTIPLE>
<OPTION VALUE=1>Domingo
<OPTION VALUE=2 SELECTED>Segunda-Feira
.
.
.
</SELECT>

Form.Submit "bmtEnvia", "Emitir"
Form.Reset "Limpar"

Form.EndForm

```

Aplica-se

Form

Veja também

Método Submit

Descrição

Exibe um botão que, ao ser clicado, submete o formulário.

Sintaxe

objeto.**Submit** *String1*, *String2*

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Form</i> .
<i>String1</i>	Requerido. Nome do botão.
<i>String2</i>	Requerido. Valor do botão. O valor será apresentado como texto no botão.

Comentários

O código abaixo ilustra o uso do método **Submit**.

```

Form.BeginForm Initializer.ScriptName, "frmEnviar", , , "return
ValidateField()"

Form.Radio "Nome", "Teste", "Right", 1
Form.Radio "Nome2", "Teste"

Form.Submit "bmtEnvia", "Emitir"
Retorna: <INPUT TYPE="SUBMIT" NAME="bmtEnvia" VALUE="Emitir">

```

```
Form.Reset "Limpar"
```

```
Form.EndForm
```

Aplica-se
Form

Veja também

Método Text

Descrição

Exibe um campo para digitação de texto em uma única linha.

Sintaxe

objeto.**Text** *String*, *Integer1* [, *Variant*, *Integer2*]

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Form</i> .
<i>String</i>	Requerido. Nome do campo.
<i>Integer1</i>	Requerido. Comprimento do campo em caracteres.
<i>Variant</i>	Opcional. Valor do campo.
<i>Integer2</i>	Opcional. Comprimento máximo do valor do campo, em caracteres.

Comentários

O código abaixo ilustra o uso do método **Text**.

```
Form.BeginForm Initializer.ScriptName, "frmEnviar", , , "return  
ValidateField()"
```

```
Form.Text "Usuario", 10, , 20
```

```
Retorna: <INPUT TYPE="TEXT" NAME="Usuario" SIZE=10 MAXLENGTH=20>
```

```
Form.Text "Usuario", 10, "UsuarioX", 20
```

```
Retorna: <INPUT TYPE="TEXT" NAME="Usuario" VALUE="UsuarioX" SIZE=10  
MAXLENGTH=20>
```

```
Form.Password "Senha", 10, 20
```

```
Form.Submit "bmtEnvia", "Emitir"
```

```
Form.Reset "Limpar"
```

`Form.EndForm`**Aplica-se**

Form

Veja também

Método TextArea

Descrição

Exibe um campo para digitação de texto em várias linhas.

Sintaxe

objeto.**TextArea** *String1* [, *Integer1*, *Integer2*, *String2*]

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Form</i> .
<i>String</i>	Requerido. Nome do campo.
<i>Integer1</i>	Requerido. Comprimento do campo em caracteres.
<i>Integer2</i>	Opcional. Comprimento máximo do valor do campo, em caracteres.
<i>String2</i>	Opcional. Texto do campo.

Comentários

O código abaixo ilustra o uso do método **TextArea**.

```
Form.BeginForm Initializer.ScriptName, "frmEnviar", , , "return  
ValidateField()"
```

```
Form.Hidden "Campo Escondido", "Valor do campo"
```

```
Form.FormInput "SUBMIT", "bmtEnvia", "Emitir"
```

```
Form.TextArea "SUBMIT", "TextAreal", 4, 40, "Conteudo do Text Area"
```

```
Retorna: <TEXTAREA NAME="TextAreal" ROWS=4 COLS=40> Conteudo do Text  
Area</TEXTAREA>
```

```
Form.Submit "bmtEnvia", "Emitir"
```

```
Form.Reset "Limpar"
```

```
Form.EndForm
```

Aplica-se

Form

Objeto Initializer

Objeto_INITIALIZER

Inicializa estruturas de dados utilizadas pelos vários objetos do HiperTools, principalmente o objeto Default, e retorna informações sobre a execução do script atual.

Sintaxe

Set Initializer = Server.CreateObject("HiperTools30.Initializer")

Propriedade ErrorMessage

Descrição

Define ou retorna a cabeçalho das mensagens de erro tratadas pelo HiperTools.

Sintaxe

objeto.ErrorMessage [= String]

O argumento *objeto* sempre será o objeto Initializer. O valor para **ErrorMessage** é o texto do cabeçalho das mensagens de erro.

Comentários

O código abaixo ilustra o uso da propriedade **ErrorMessage**.

```
Response.Write Initializer.ErrorMessage
```

Retorna: "Ocorreu um erro interno do sistema. Entre em contato com a Zevallos Tecnologia em Informação no e-mail info@hipertools.com.br ou no telefone (61) 321-4711."

```
Initializer.ErrorMessage = "Ocorreu um erro interno do sistema. Entre em contato com a XYZ Software no e-mail info@xyz.com.br ou no telefone (99) 123-4567"
```

```
Response.Write Initializer.ErrorMessage
```

Retorna: "Ocorreu um erro interno do sistema. Entre em contato com a XYZ Software no e-mail info@xyz.com.br ou no telefone (99) 123-4567."

Aplica-se

Initializer

Veja também

Propriedade HiperToolsWorkPath

Descrição

Define ou retorna o diretório virtual no IIS do HiperTools no site da aplicação.

Sintaxe

objeto.HiperToolsWorkPath [= String]

O argumento *objeto* sempre será o objeto Initializer. O valor para **HiperToolsWorkPath** é o caminho do diretório virtual.

Comentários

O código abaixo ilustra o uso da propriedade **HiperToolsWorkPath**.

```
Response.Write Initializer.HiperToolsWorkPath  
Retorna: /HiperTools
```

Aplica-se
Initializer

Veja também

Propriedade HTTPHost

Descrição

Define ou retorna o HTTP do Script atual. O mesmo que o `Request.ServerVariables("HTTP_HOST")`.

Sintaxe

objeto.HTTPHost [= String]

O argumento *objeto* sempre será o objeto Initializer. O valor para **HTTPHost** é o HTTP.

Comentários

O código abaixo ilustra o uso da propriedade **HTTPHost**.

```
Response.Write Initializer.HTTPHost  
Retorna: http://www.zevallos.com.br
```

Aplica-se
Initializer

Veja também

Propriedade LastQueryString

Descrição

Recupera a Query String da última execução do script. Essa propriedade é *read-only*.

Sintaxe

[String =] *objeto*.LastQueryString

O argumento *objeto* sempre será o objeto *Initializer*.

Comentários

O código abaixo ilustra o uso do método **LastQueryString**.

```
Response.Write "[" & Initializer.LastQueryString & "]"  
Retorna: [teste.asp?T=2]  
Response.Write "[" & Initializer.QueryString & "]"  
Retorna: [teste.asp?T=3]
```

Aplica-se
Initializer

Veja também

Propriedade QueryString

Descrição

Retorna a QueryString atual. Essa propriedade é *read-only*.

Sintaxe

[String =] **objeto.QueryString**

O argumento *objeto* sempre será o objeto *Initializer*.

Comentários

O código abaixo ilustra o uso da propriedade **QueryString**.

```
Response.Write "[" & Initializer.LastQueryString & "]"  
Retorna: [teste.asp?T=2]  
Response.Write "[" & Initializer.QueryString & "]"  
Retorna: [teste.asp?T=3]
```

Aplica-se

Initializer

Veja também

Propriedade ScriptFull

Descrição

Define ou retorna o endereço no sistema do Script atual. O mesmo que o `Request.ServerVariables("PATH_TRANSLATED")`.

Sintaxe

objeto.ScriptFull [= String]

O argumento *objeto* sempre será o objeto *Initializer*. O valor para **ScriptFull** é o endereço completo do script.

Comentários

O código abaixo ilustra o uso da propriedade **ScriptFull**.

```
Response.Write Initializer.ScriptFull  
Retorna: D:\inetpub\wwwroot\clients\site\teste.asp
```

Aplica-se

Initializer

Veja também

Propriedade ScriptName

Descrição

Define ou retorna o nome do Script atual. O mesmo que o `Request.ServerVariables("SCRIPT_NAME")`.

Sintaxe

objeto.**ScriptName** [= *String*]

O argumento *objeto* sempre será o objeto Initializer. O valor para **ScriptName** é o nome completo.

Comentários

O código abaixo ilustra o uso da propriedade **ScriptName**.

```
Response.Write Initializer.ScriptName  
Retorna: teste.asp
```

Aplica-se

Initializer

Veja também

Propriedade ScriptURL

Descrição

Define ou retorna a URL do Script atual. O mesmo que o `Request.ServerVariables("HTTP_HOST")` e `Request.ServerVariables("SCRIPT_NAME")`.

Sintaxe

objeto.**ScriptURL** [= *String*]

O argumento *objeto* sempre será o objeto Initializer. O valor para **ScriptURL** é a URL completa.

Comentários

O código abaixo ilustra o uso da propriedade **ScriptURL**.

```
Response.Write Initializer.ScriptURL  
Retorna: http://www.zevallos.com.br/teste.asp
```

Aplica-se

Initializer

Veja também

Propriedade SiteRootDir

Descrição

Define ou retorna o diretório raiz do site do Script atual. O mesmo que o `Request.ServerVariables("APPL_PHYSICAL_PATH")`.

Sintaxe

objeto.**SiteRootDir** [= *String*]

O argumento *objeto* sempre será o objeto *Initializer*. O valor para **SiteRootDir** é o endereço completo do diretório raiz do script.

Comentários

O código abaixo ilustra o uso da propriedade **SiteRootDir**.

```
Response.Write Initializer.SiteRootDir  
Retorna: D:\inetpub\wwwroot\clients\site\
```

Aplica-se

Initializer

Veja também

Propriedade WhenStarted

Descrição

Retorna a data e a hora em que um dos objetos do HiperTools foi criado pela primeira vez. Sempre que o IIS for parado, este valor será reiniciado.

Sintaxe

objeto.WhenStarted

O argumento *objeto* sempre será o objeto *Initializer*.

Comentários

O valor retornado pela propriedade **WhenStarted** é a data e hora da criação do primeiro objeto.

O código abaixo ilustra o uso da propriedade **WhenStarted**.

```
Response.Write Initializer.WhenStarted  
Retorna: 02/18/2000 09:58:34 PM
```

Aplica-se

Initializer

Veja também

Método InitDefault

Descrição

Inicializa todos os parametros do objeto com as informações padrão.

Sintaxe

objeto.InitDefault

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Initializer</i>

Comentários

O código abaixo ilustra o uso do método **InitDefault**.

```
Initializer.InitDefault  
Default.HTMLBegin  
Default.HeadAll "Zeus III"  
Default.BodyBegin  
Show.HTML "XYZ Software"  
Default.PageFooterDefault  
Default.BodyEnd  
Default.HTMLEnd
```

Aplica-se
Initializer

Veja também

Objeto Pack

Objeto Pack

Esse objeto consiste em uma *Collection* que suporta a criação, armazenamento e acesso de pares chave/valor na memória. Cada item num objeto Pack é um *PackItem*, que através da propriedade *Value* permite armazenar uma variável do tipo *Variant*. Pode-se adicionar qualquer tipo de dado ao Pack, inclusive outro Pack ou outros objetos. Além disso, cada item de um Pack possui uma propriedade Pack que automaticamente cria outro Pack dentro do Pack instanciado.

Comentários

O código abaixo ilustra como instanciar o *Pack* e utilizar suas propriedades e métodos.

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
Dim Pack, item, InnerPack

Set Pack = Server.CreateObject("HiperTools30.Pack")
Set InnerPack = Server.CreateObject("HiperTools30.Pack")

Pack.Add "Driver", "{SQL Server Driver}"
Pack.Add "Server", "MyServer"
Pack.Add "UserID", "sa"
Pack.Add "PWD", "sqlserver"

InnerPack.Add "Table0", "Pedidos"
InnerPack.Add "Table1", "IPedidos"

REM Adicionando um Pack como valor de um Pack
Pack.Add "Innerpack", InnerPack

Pack("Innerpack").Value.Add "Table2", "Compras"
Pack("Innerpack").Value.Add "ParentPack", Pack

REM Utilizado o Pack do PackItem
Pack("Table2").Pack.Add "Field0", "Codigo"
Pack("Table2").Pack.Add "Field1", "Pedido"
Pack("Table2").Pack.Add "Field2", "Data"

Pack("Table2").Pack("Field1").Pack.Add "Value0", 0
Pack("Table2").Pack("Field1").Pack.Add "Value1", 1

Response.Write "<BR>" & Pack("Table2").Pack.Count
Response.Write "<BR>" & Pack("Table2").Pack("Field1").Pack.Count

For Each item In Pack("Innerpack").Value
    If Not IsObject(item.Value) Then Response.Write "<BR>" & item.Value
Next

Response.Write "<BR>" & Pack("Innerpack")("Table2")

If Pack.Find("Table0") Then
    Response.Write "<BR>" & Pack.Item("Table0").Value
    Response.Write "<BR>" & Pack.Item(1).Value
    Response.Write "<BR>" & Pack.Item(2).Value
Else
```

```
Response.Write "Não existe o item ""Table0"""
End If

Response.Write "<BR>" & Pack("Table1").Value
Response.Write "<BR>" & Pack.Value("Table1")
Response.Write "<BR>" & Pack.Key("Table1")
Response.Write "<BR>" & Pack("Table1").Key

Response.Write "<BR>" & "======"

Pack.Item(2).Value = "PedidosAlt"
Response.Write "<BR>" & Pack.Item(2).Value

Pack(2).Value = "PedidosAlterado"
Response.Write "<BR>" & Pack(2).Value

Pack.Value(2) = "PedidosAltDeNovo"
Response.Write "<BR>" & Pack.Item(2).Value

Pack(2).Value = "PedidosAlteradoDeNovo"
Response.Write "<BR>" & Pack(2)

Pack("Table1") = "Pedidos"
Response.Write "<BR>" & Pack("Table1")

Response.Write "<BR>" & "======"

For Each item In Pack
    Response.Write "<BR>" & item.Key
Next

Pack.Clear

REM Aqui ele destroi todos os pack "filhos"
Response.Write "<BR>" & Pack.Count

Set Pack = Nothing
Set InnerPack = Nothing
%>
```

Propriedade Count

Descrição

Retorna a quantidade de itens adicionados ao Pack.

Sintaxe

objeto.Count

O argumento *objeto* sempre será o objeto Pack.

Comentários

O código abaixo ilustra o uso da propriedade **Count**.

```
Pack.Add "Table0", "Pedidos"
Pack.Add "Table1", "IPedidos"
Pack.Add "Table2", "Produtos"
Pack.Add "Table3", "Requisitos"
Response.Write Pack.Count
```

Objeto Pack

```
Retorna: 4

Pack.Remove 1
Response.Write Pack.Count

Retorna: 3
```

Aplica-se
Pack

Veja também
Remove, Find

Propriedade Item

Descrição

Retorna a referência a um item adicionado ao Pack. Essa é a propriedade padrão do objeto Pack e é *read-only*.

Sintaxe

objeto.Item(*Variant*)

ou

objeto(*Variant*)

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto Pack.
<i>Variant</i>	Requerido. Índice ou chave do item.

Comentário

O código abaixo ilustra o uso da propriedade **Item**.

```
Pack.Add "Table0", "Pedidos"

Pack("Table0") = "PedidosAlt"
Pack.Item(1).Value = "PedidosAlt"

Response.Write "<BR>" & Pack.Item(1).Value

Retorna: <BR>PedidosAlt

Pack(1).Value = "PedidosAlterado"

Response.Write "<BR>" & Pack(1).Value
Response.Write "<BR>" & Pack("Table0")

Retorna: <BR>PedidosAlterado
        <BR>PedidosAlterado
```

Aplica-se
Pack

Veja também
PackItem.Value, PackItem.Pack

Método Add

Descrição

Adiciona um item ao Pack e retorna uma referência a este item.

Sintaxe

objeto.Add *String*, [*Variant*]

ou

Set *objeto1* = *objeto*.Add(*String*, [*Variant*])

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto Pack.
<i>objeto1</i>	Requerido. Objeto que referenciará o item adicionado.
<i>String</i>	Requerido. Chave do item.
<i>Variant</i>	Opcional. Valor do item.

Comentários

O código abaixo ilustra o uso do método **Add**.

```
Dim Pack
Set Pack = Server.CreateObject("HiperTools30.Pack")
```

```
Pack.Add "Table0", "Pedidos"
Pack.Add "Table1", "IPedidos"
Pack.Add "Table2", "Produtos"
Pack.Add "Table3", "Requisitos"
Response.Write Pack.Count
```

Retorna: 4

```
Response.Write Pack(1)
Retorna: Pedidos
```

```
Response.Write Pack("Table3")
Retorna: Requisitos
```

Aplica-se

Pack

Veja também

PackItem.Value, PackItem.Key, PackItem.Position, Remove.

Método Find

Descrição

Verifica se um determinado item existe e permite removê-lo.

Sintaxe

objeto.Find *String*[, *Boolean*]

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto Pack.
<i>String</i>	Requerido. Chave ou índice do item.
<i>Boolean</i>	Opcional. Especifica se o item deve ser removido. O padrão para essa propriedade é <i>falso</i> , não removendo o item.

Comentários

O código abaixo ilustra o uso da propriedade **Find**.

```
Dim Pack
Set Pack = Server.CreateObject("HiperTools30.Pack")

Pack.Add "Table0", "Pedidos"
Pack.Add "Table1", "IPedidos"
Pack.Add "Table2", "Produtos"

If Pack.Find("Table1") Then
    Pack("Table1") = "JPedidos"
End If

If Pack.Find("Table2", True) Then
    'Item apagado
    Response.Write Pack.Count
    Retorna: 2
End If
```

Aplica-se

Pack

Veja também

PackItem.Position, PackItem.Key, Remove.

Método Remove

Descrição

Remove um item do Pack.

Sintaxe

objeto.Remove String

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto Pack.
<i>String</i>	Requerido. Índice ou chave do item a ser removido.

Comentários

O código abaixo ilustra o uso do método **Remove**.

```
Dim Pack
Set Pack = Server.CreateObject("HiperTools30.Pack")

Pack.Add "Table0", "Pedidos"
```

```
Pack.Add "Table1", "IPedidos"  
Pack.Add "Table2", "Produtos"  
Response.Write Pack.Count
```

Retorna: 3

```
Pack.Remove("Table1")  
Pack.Remove("Table2")  
Response.Write Pack.Count
```

Retorna: 1

Aplica-se

Pack

Veja também

PackItem.Key, PackItem.Position, Add, Find

Objeto PackItem

Objeto PackItem

Esse objeto implementa os itens do objeto Pack.

Sintaxe

Esse é um objeto de uso interno do sistema e não deve ser instanciado via código.

Propriedade Key

Descrição

Determina a chave do item do Pack.

Sintaxe

objeto.**Key**

O argumento *objeto* sempre será o objeto *PackItem*.

Comentários

O código abaixo ilustra o uso da propriedade Value.

```
Dim Pack, Item
Set Pack = Server.CreateObject("HiperTools30.Pack")

Pack.Add "Table0", "Pedidos"

Set Item = Pack("Table0") 'Aqui se referencia o PackItem

Pack("Table0") = "PedidosAlt"
Pack.Item(1).Value = "PedidosAlt"

Response.Write "<BR>" & Item.Key

Retorna: <BR>Table0
```

Aplica-se

PackItem

Veja também

Pack.Add, Pack.Remove, Pack.Find

Propriedade Pack

Descrição

Retorna uma nova instância do objeto Pack. Isso permite que cada item seja um outro objeto Pack, como uma lista encadeada.

Sintaxe

objeto.**Pack**

O argumento *objeto* sempre será o objeto *PackItem*.

Comentários

O código abaixo ilustra o uso da propriedade Pack.

```
Dim Pack
Set Pack = Server.CreateObject("HiperTools30.Pack")

Pack.Add "Table2", "Pedidos"

Pack("Table2").Pack.Add "Field0", "Codigo"
```

```
Pack("Table2").Pack.Add "Field1", "Pedido"
Pack("Table2").Pack.Add "Field2", "Data"

Response.Write Pack("Table2").Pack.Count

Retorna: 3

Response.Write Pack("Table2").Pack("Field2")

Retorna: Data
```

Aplica-se
PackItem

Veja também
Pack

Propriedade Position

Descrição

Especifica a posição, ou índice, do um item no Pack. O primeiro item está na posição 1, o segundo na posição 2 e assim por diante.

Sintaxe

objeto.**Position** [= Long]

O argumento *objeto* sempre será o objeto *PackItem*.

Comentários

O código abaixo ilustra o uso da propriedade *Position*.

```
Dim Pack, Item
Set Pack = Server.CreateObject("HiperTools30.Pack")

Pack.Add "Table0", "Pedidos"
Set Item = Pack("Table0") 'Aqui se referencia o PackItem
Response.Write "<BR>" & Item.Position
Retorna: <BR>1

Pack.Item(1).Position = 5
Response.Write "<BR>" & Item.Position
Retorna: <BR>5
```

Aplica-se
PackItem

Veja também

Propriedade Value

Descrição

Especifica o valor do item do Pack.

Sintaxe

objeto.**Value** [=Variant]

O argumento *objeto* sempre será o objeto *PackItem*.

Comentários

O código abaixo ilustra o uso da propriedade **Value**.

```
Dim Pack, Item
Set Pack = Server.CreateObject("HiperTools30.Pack")

Pack.Add "Table0", "Pedidos"

Set Item = Pack("Table0") `Aqui se referencia o PackItem

Pack("Table0") = "PedidosAlt"
Pack.Item(1).Value = "PedidosAlt"

Response.Write "<BR>" & Item.Value

Retorna: <BR>PedidosAlt
```

Aplica-se

PackItem

Veja também

Pack.Item, Pack.Add

Objeto Path

Objeto Path

Objeto para manipulação e conversão de caminhos (path).

Sintaxe

```
Set Path = Server.CreateObject("HiperTools30.Path")
```

Método Drive

Descrição

Retorna somente a letra do drive de um caminho.

Sintaxe

objeto.**Drive** *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto Path.
<i>String</i>	Requerido. <i>String</i> é o caminho de que será extraída a letra do drive.

Comentários

O código abaixo ilustra o uso do método **Drive**.

```
Response.Write Path.Drive "C:\admin\controle.asp"  
Retorna: "C:"
```

Aplica-se

Path

Veja também

Only

Método InvertSlash

Descrição

Inverte as barras do caminho de “\” para “/” e vise-versa.

Sintaxe

objeto.**InvertSlash** *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto Path.
<i>String</i>	Requerido. <i>String</i> é o caminho o qual terá suas barras invertidas.

Comentários

O código abaixo ilustra o uso do método **InvertSlash**.

```
Response.Write Path.InvertSlash "/admin/controle.asp"  
Retorna: "\admin\controle.asp"
```

Aplica-se

Path

Veja também

NormalSlash, URLSlash, SlashType

Método NormalSlash

Descrição

Transforma as barras normais para as barras de invertidas, que são barras “\”.

Sintaxe*objeto*.**NormalSlash** *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto Path.
<i>String</i>	Requerido. <i>String</i> é o caminho que terá suas barras transformadas em barras invertidas.

Comentários

O código abaixo ilustra o uso do método **NormalSlash**.

```
Response.Write Path.NormalSlash "\admin/controle.asp"  
Retorna: "\admin\controle.asp"
```

Aplica-se

Path

Veja também

InvertSlash, URLSlash, SlashType

Método Only

Descrição

Retorna somente o caminho sem a letra do drive e sem o nome do arquivo.

Sintaxe*objeto*.**Only** *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto Path.
<i>String</i>	Requerido. <i>String</i> é o caminho que será utilizado para extrair o caminho.

Comentários

O código abaixo ilustra o uso do método **Only**.

```
Response.Write Path.Only "C:\admin\controle.asp"  
Retorna: "\admin\controle.asp"
```

Aplica-se

Path

Veja também

Drive

Método RemoveLastSlash

Descrição

Exclui a barra a mais direita caso esta exista.

Sintaxe

objeto.RemoveLastSlash String

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto Path.
<i>String</i>	Requerido. <i>String</i> é o caminho que será verificado e excluída a barra do lado direito.

Comentários

O código abaixo ilustra o uso do método **RemoveLastSlash**.

```
Response.Write Path.RemoveLastSlash "\\admin\\"  
Retorna: "\\admin"
```

```
Response.Write Path.RemoveLastSlash "\\admin"  
Retorna: "\\admin"
```

Aplica-se

Path

Veja também

Método SlashType

Descrição

Retorna a barra (Slash) ("/\\")que esta sendo utilizado no caminho informado.

Sintaxe

objeto.SlashType String

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto Path.
<i>String</i>	Requerido. <i>String</i> é o caminho que será retornado a barra

Comentários

O código abaixo ilustra o uso do método **SlashType**.

```
Response.Write Path.SlashType "/admin/controle.asp"  
Retorna: /
```

Aplica-se

Path

Veja também

InvertSlash, NormalSlash, URLSlash

Método URLSlash

Descrição

Transforma as barras invertidas para as barras de uma URL normal, que são barras "/".

Sintaxe*objeto.URLSlash String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto Path.
<i>String</i>	Requerido. <i>String</i> é o caminho que transformado em URL.

Comentários

O código abaixo ilustra o uso do método **URLSlash**.

```
Response.Write Path.URLSlash "\admin/controle.asp"
```

```
Retorna: "/admin/controle.asp"
```

Aplica-se

Path

Veja também

InvertSlash, NormalSlash, SlashType

Objeto Registry

Objeto Registry

Objeto de acesso ao registro do Windows.

Sintaxe

```
Set Registry = Server.CreateObject("HiperTools30.Registry")
```

Propriedade AutoCreate

Descrição

Determina se as chaves e valores devem ser automaticamente criados caso sejam acessados e não existam.

Sintaxe

objeto.**AutoCreate** [= *Boolean*]

objeto será sempre uma referência a um objeto *Registry*.

Comentários

O código abaixo ilustra o uso da propriedade **AutoCreate**.

```
Registry.AutoCreate = True
```

Aplica-se

Registry

Veja também

Propriedade HKEY

Descrição

Determina a entrada HKEY atual. Essa propriedade é *write-only*.

Sintaxe

objeto.**HKEY** [= *Long*]

objeto será sempre uma referência a um objeto *Registry*.

Comentários

A essa propriedade podem ser atribuídos os valores das constantes a seguir:

Constante	Valor	Descrição
<code>rgHKEYClassesRoot</code>	<code>&H80000000</code>	<code>HKEY_CLASSES_ROOT</code>
<code>rgHKEYCurrentUser</code>	<code>&H80000001</code>	<code>HKEY_CURRENT_USER</code>
<code>rgHKEYLocalMachine</code>	<code>&H80000002</code>	<code>HKEY_LOCAL_MACHINE</code>
<code>rgHKEYUsers</code>	<code>&H80000003</code>	<code>HKEY_USERS</code>
<code>rgHKEYPerformanceData</code>	<code>&H80000004</code>	<code>HKEY_PERFORMANCE_DATA</code>
<code>rgHKEYCurrentConfig</code>	<code>&H80000005</code>	<code>HKEY_CURRENT_CONFIG</code>
<code>rgHKEYDynData</code>	<code>&H80000006</code>	<code>HKEY_DYN_DATA</code>

O código abaixo ilustra o uso da propriedade **HKEY**.

```
Registry.HKEY = rgHKEYLocalMachine
```

Aplica-se

Registry

Veja também

Propriedade HKEYName

Descrição

Retorna uma *String* representando a entrada HKEY atual. Essa propriedade é read-only.

Sintaxe

objeto.**HKEYName**

objeto será sempre uma referência a um objeto *Registry*.

Comentários

O código abaixo ilustra o uso da propriedade **HKEYName**.

```
Show.HTMLCR Registry.HKEYName
```

Aplica-se

Registry

Veja também

Propriedade Key

Descrição

Determina a chave a ser acessada.

Sintaxe

objeto.**Key** [= *String*]

objeto será sempre uma referência a um objeto *Registry*.

Comentários

O código abaixo ilustra o uso da propriedade **Key**.

```
Registry.Key = "SOFTWARE\Microsoft\INetStp\Teste"
```

Aplica-se

Registry

Veja também

Propriedade SubKeys

Descrição

Retorna um objeto *RegistrySubKeys* contendo informações sobre as subchaves da chave atual.

Sintaxe

objeto.**SubKeys**

objeto será sempre uma referência a um objeto *Registry*.

Comentários

O código abaixo ilustra o uso da propriedade **SubKeys**.

```
For intIdx = 1 To Registry.SubKeys.Count  
    Show.HTMLCR Registry.SubKeys(intIdx).Name & "<BR>"  
Next
```

Aplica-se

Registry

Veja também

RegistrySubKeys

Propriedade Values

Descrição

Retorna um objeto *RegistryValues* contendo informações sobre os valores da chave atual.

Sintaxe

objeto.**Values**

objeto será sempre uma referência a um objeto *Registry*.

Comentários

O código abaixo ilustra o uso da propriedade **Values**.

```
For Each objValue In Registry.Values  
    Show.HTMLCR objValue.Name & "<BR>"  
Next
```

Aplica-se

Registry

Veja também

RegistryValues

Método CreateKey

Descrição

Cria a chave definida na propriedade *Key*.

Sintaxe

objeto.**CreateKey**

Parte	Descrição
<i>objeto</i>	Será sempre uma referência a um objeto <i>Registry</i> .

Comentários

O código abaixo ilustra o uso do método **CreateKey**.

```
Registry.HKEY = rgHKEYLocalMachine
Registry.Key = "SOFTWARE\Microsoft\INetStp\Teste"
Registry.CreateKey
```

Aplica-se

Registry

Veja também

Método CreateValue

Descrição

Cria um valor dentro da chave definida na propriedade *Key*.

Sintaxe

objeto.CreateValue *String*, [*Variant*] , [*Long*]

Parte	Descrição
<i>Objeto</i>	Será sempre uma referência a um objeto <i>Registry</i> .
<i>String</i>	Nome do valor a ser criado.
<i>Variant</i>	Opcional. Dados a serem armazenados no valor.
<i>Long</i>	Opcional. Tipo do valor. Deve ser uma das constantes de tipo de valor. Caso o parâmetro não seja fornecido, o valor é considerado REG_SZ.

Comentários

Os possíveis valores para o parâmetro *Long* são os seguintes:

Constante	Valor	Descrição
RgSz	1	<i>String</i> Unicode
RgExpandSz	2	<i>String</i> com variáveis de ambiente (%valor%)
RgBinary	3	valor binário genérico (matriz de Bytes)
RgDword	4	double word (Long)

O código abaixo ilustra o uso do método **CreateValue**.

```
Registry.CreateValue "MyString", "HiperTools"  
Registry.CreateValue "MyDWord", 1999, rgDword
```

Aplica-se
Registry

Veja também

Método DeleteKey

Descrição

Deleta a chave definida na propriedade *Key*.

Sintaxe

objeto.DeleteKey

Parte	Descrição
<i>Objeto</i>	Será sempre uma referência a um objeto <i>Registry</i> .

Comentários

O código abaixo ilustra o uso do método **DeleteKey**.

```
Registry.HKEY = rgHKEYCurrentUser  
Registry.Key = "SOFTWARE\Microsoft\INetStp\Teste"  
Registry.DeleteKey
```

Aplica-se
Registry

Veja também

Método DeleteValue

Descrição

Deleta um valor da chave atual.

Sintaxe

objeto.DeleteValue *String*

Parte	Descrição
<i>objeto</i>	Será sempre uma referência a um objeto <i>Registry</i> .
<i>String</i>	Nome do valor a ser deletado.

Comentários

O código abaixo ilustra o uso do método **DeleteValue**.

```
Registry.DeleteValue "MyString"
```

Aplica-se
Registry

Veja também

Método ValueAsBinary

Descrição

Lê ou grava um valor do tipo REG_BINARY, como uma matriz de *Bytes*. Se AutoCreate for *verdadeiro* o valor é criado caso não exista.

Sintaxe

objeto.ValueAsBinary *String*

Parte	Descrição
<i>objeto</i>	Será sempre uma referência a um objeto <i>Registry</i> .
<i>String</i>	Nome do valor a ser acessado.

Comentários

O código abaixo ilustra o uso do método **ValueAsBinary**.

```
arrValue = Registry.ValueAsBinary("MyBinary")
For intIdx = 0 To Ubound(arrValue)
    strValue = strValue & Chr(arrValue(intIdx))
End If
```

Aplica-se
Registry

Veja também

Método ValueAsBoolean

Descrição

Lê ou grava um valor do tipo REG_DWORD, efetuando a conversão de tipo *Boolean*. Se AutoCreate for *verdadeiro* o valor é criado caso não exista.

Sintaxe

objeto.ValueAsBoolean *String*

Parte	Descrição
<i>objeto</i>	Será sempre uma referência a um objeto <i>Registry</i> .
<i>String</i>	Nome do valor a ser acessado.

Comentários

O código abaixo ilustra o uso do método **ValueAsBoolean**.

```
Registry.ValueAsBoolean("MyBoolean") = True
If Registry.ValueAsBoolean("MyBoolean") Then
    Show.HTMLCR "O valor é verdadeiro."
End If
```

Aplica-se
Registry

Veja também

Método ValueAsLong

Descrição

Lê ou grava um valor do tipo REG_DWORD. Se AutoCreate for verdadeiro o valor é criado caso não exista.

Sintaxe

objeto.ValueAsLong String

Parte	Descrição
<i>objeto</i>	Será sempre uma referência a um objeto <i>Registry</i> .
<i>String</i>	Nome do valor a ser acessado.

Comentários

O código abaixo ilustra o uso do método **ValueAsLong**.

```
Registry.ValueAsLong("MyDWord") = 1999
Show.HTMLCR Registry.ValueAsLong("MyDWord")
```

Aplica-se
Registry

Veja também

Método ValueAsString

Descrição

Lê ou grava um valor do tipo REG_SZ. Se AutoCreate for *verdadeiro* o valor é criado caso não exista.

Sintaxe

objeto.ValueAsString String

Parte	Descrição
<i>objeto</i>	Será sempre uma referência a um objeto <i>Registry</i> .
<i>String</i>	Nome do valor a ser acessado.

Comentários

O código abaixo ilustra o uso do método **ValueAsString**.

```
Registry.ValueAsString("MyString") = "Texto exemplo"  
Show.HTMLCR Registry.ValueAsString("MyString")
```

Aplica-se

Registry

Veja também

Objeto RegistrySubKey

Objeto RegistrySubKey

Objeto que armazena os dados referentes a uma subchave no registro.

Sintaxe

Esse é um objeto de uso interno do sistema e não deve ser instanciado via código.

Propriedade LastWrite

Descrição

Retorna um *Date* representando a data e hora da última gravação feita na subchave. Essa propriedade é *read-only*.

Sintaxe

objeto.LastWrite

objeto será sempre uma referência a um objeto *RegistrySubKey*.

Comentários

O código abaixo ilustra o uso da propriedade **LastWrite**.

```
Show.HTMLCR "Último acesso à subchave: "  
Show.HTMLCR Registry.SubKeys("chave").LastWrite
```

Aplica-se

RegistrySubKey

Veja também

Propriedade Name

Descrição

Retorna uma *String* representando o nome da subchave. Essa propriedade é *read-only*.

Sintaxe

objeto.Name

objeto será sempre uma referência a um objeto *RegistrySubKey*.

Comentários

O código abaixo ilustra o uso da propriedade **Name**.

```
Show.HTMLCR Registry.SubKeys(2).Name
```

Aplica-se

RegistrySubKey

Veja também

Objeto RegistrySubKeys

Objeto RegistrySubKeys

Objeto que implementa uma *Collection* de objetos *RegistrySubKey*.

Sintaxe

Esse é um objeto de uso interno do sistema e não deve ser instanciado via código.

Propriedade Count

Descrição

Retorna o número de itens no objeto *RegistrySubKeys*. Essa propriedade é *read-only*.

Sintaxe

objeto.Count

objeto será sempre uma referência a um objeto *RegistrySubKeys*.

Comentários

O código abaixo ilustra o uso da propriedade **Count**.

```
Show.HTMLCR "Há " & Registry.SubKeys.Count & " subchaves."
```

Aplica-se

RegistrySubKeys

Veja também

Propriedade Item

Descrição

Retorna o objeto *RegistrySubKey* associado ao item especificado. Essa propriedade é *read-only*.

Sintaxe

objeto.Item(*Variant*)

Parte	Descrição
<i>objeto</i>	Será sempre uma referência a um objeto <i>RegistrySubKeys</i> .
<i>Variant</i>	<i>String</i> representando o nome da subchave ou <i>Integer</i> indicando o índice.

Comentários

O código abaixo ilustra o uso da propriedade **Item**.

```
For intIdx = 1 To Registry.SubKeys.Count  
Show.HTMLCR Registry.SubKeys(intIdx).Name & "<BR>"  
Next
```

Aplica-se
RegistrySubKeys

Veja também

Objeto RegistryValue

Objeto RegistryValue

Objeto que armazena os dados referentes a um valor no registro.

Sintaxe

Esse é um objeto de uso interno do sistema e não deve ser instanciado via código.

Propriedade DataType

Descrição

Retorna um *Long* representando o tipo do valor. Essa propriedade é *read-only*.

Sintaxe

objeto.**DataType**

objeto será sempre uma referência a um objeto *RegistryValue*.

Comentários

O código abaixo ilustra o uso da propriedade **DataType**.

```
With Registry
.CreateValue "dword2", "", .Values("dword1").DataType
End With
```

Aplica-se

RegistryValue

Veja também

Propriedade Name

Descrição

Retorna uma *String* representando o nome do valor. Essa propriedade é *read-only*.

Sintaxe

objeto.**Name**

objeto será sempre uma referência a um objeto *RegistryValue*.

Comentários

O código abaixo ilustra o uso da propriedade **Name**.

```
Show.HTMLCR Registry.Values(2).Name
```

Aplica-se

RegistryValue

Veja também

Objeto RegistryValues

Objeto RegistryValues

Objeto que implementa uma *Collection* de objetos *RegistryValue*.

Sintaxe

Esse é um objeto de uso interno do sistema e não deve ser instanciado via código.

Propriedade Count

Descrição

Retorna o número de itens no objeto *RegistryValues*. Essa propriedade é *read-only*.

Sintaxe

objeto.Count

objeto será sempre uma referência a um objeto *RegistryValues*.

Comentários

O código abaixo ilustra o uso da propriedade **Count**.

```
Show.HTMLCR "Há " & Registry.Values.Count & " valores."
```

Aplica-se

RegistryValues

Veja também

Propriedade Item

Descrição

Retorna o objeto *RegistryValue* associado ao item especificado. Essa propriedade é *read-only*.

Sintaxe

objeto.Item(*Variant*)

Parte	Descrição
<i>objeto</i>	Será sempre uma referência a um objeto <i>RegistryValues</i> .
<i>Variant</i>	<i>String</i> representando o nome do valor ou <i>Integer</i> indicando o índice.

Comentários

O código abaixo ilustra o uso da propriedade **Item**.

```
With Registry
  If .Values(intIdx).Name = .Values("acesso").Name Then
    Show.HTMLCR "O índice de acesso é " & intIdx
  End If
End With
```

Aplica-se
RegistryValues

Veja também

Objeto Show

Objeto Show

Objeto de visa facilitar a apresentação de informação, substituindo o Response.Write e implementando outras funcionalidades.

Propriedade AutoWrite

Descrição

Especifica se o método *HTML* escreverá o código no documento, além de retornar a *string* correspondente ao código.

Sintaxe

objeto.**AutoWrite** [= *Boolean*]

O argumento *objeto* sempre será o objeto *Show*. Como os demais métodos do objeto *Show* utilizam internamente o método *HTML*, o valor dessa propriedade afeta também os demais métodos.

Comentários

O código abaixo ilustra o uso da propriedade **AutoWrite**.

```
Show.AutoWrite = False  
Response.Write>Show.HTML("Código retornado e escrito pelo Response.Write.")  
Show.AutoWrite = True  
Show.HTML "Código escrito diretamente."
```

Aplica-se

Show

Veja também

HTML

Propriedade CrLf

Descrição

Especifica se as quebras de linha (Cr e Lf) serão inseridas no código HTML.

Sintaxe

objeto.**CrLf** [= *Boolean*]

O argumento *objeto* sempre será o objeto *Show*. Caso o valor dessa propriedade seja *falso*, os métodos *HTMLCR*, *CRHTML* e *CRHTMLCR* se comportarão de maneira idêntica ao método *HTML*.

Comentários

O código abaixo ilustra o uso da propriedade **CrLf**.

```
Show.CrLf = False
```

Aplica-se

Show

Veja também

CrLfBefore, CRLFAfter

Propriedade CrLfAfter

Descrição

Especifica se uma quebra de linha (Cr e Lf) será acrescentada ao início de cada *string* escrita através do método *HTML*.

Sintaxe

objeto.CrLfAfter [= *Boolean*]

O argumento *objeto* sempre será o objeto *Show*. O valor dessa propriedade é ignorado caso o valor da propriedade *CrLf* seja *falso*.

Comentários

O código abaixo ilustra o uso da propriedade **CrLfAfter**.

```
Show.CrLfAfter = False
```

Aplica-se

Show

Veja também

CrLfBefore, CRLF

Propriedade CrLfBefore

Descrição

Especifica se uma quebra de linha (Cr e Lf) será acrescentada ao término de cada *string* escrita através do método *HTML*.

Sintaxe

objeto.CrLfBefore [= *Boolean*]

O argumento *objeto* sempre será o objeto *Show*. O valor dessa propriedade é ignorado caso o valor da propriedade *CrLf* seja *falso*.

Comentários

O código abaixo ilustra o uso da propriedade **CrLfBefore**.

```
Show.CrLfBefore = True
```

Aplica-se

Show

Veja também

CrLfAfter, CRLF

Propriedade ShowComments

Descrição

Especifica se os comentários serão incluídos no código HTML.

Sintaxe

objeto.ShowComments [= *Boolean*]

Objeto Show

O argumento *objeto* sempre será o objeto *Show*. Caso o valor dessa propriedade seja *falso*, a inclusão de comentários no código HTML será ignorada.

Comentários

O código abaixo ilustra o uso da propriedade **ShowComments**.

```
Show.ShowComments = False
```

Aplica-se

Show

Veja também

Método AllServerVariables

Descrição

Retorna a todas as variáveis do Request.ServerVariables.

Sintaxe

objeto.AllServerVariables

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .

Comentários

O código abaixo ilustra o uso do método **AllServerVariables**.

```
Show.AllServerVariables
```

Aplica-se

Show

Veja também

Método BeginItemList

Descrição

Inicia uma lista de itens a serem exibidos.

Sintaxe

objeto.BeginItemList *Integer*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .
<i>Integer</i>	Opcional. Uma das constantes que especificam o tipo da lista. Caso esse valor não seja especificado, o valor assumido é <i>shListTypeDisc</i> .

Comentários

Ao parâmetro podem ser atribuídos os valores das constantes a seguir:

Constante	Valor	Descrição
shListTypeDisc	1	Círculos (cheios)
shListTypeCircle	2	Circunferências
shListTypeSquare	3	Quadrados
shListTypeNumber	4	Números ordinais (1, 2, 3...)
shListTypeUChar	5	Letras maiúsculas (A, B, C...)
shListTypeLChar	6	Letras minúsculas (a, b, c...)
shListTypeURoman	7	Números romanos em maiúsculas (I, II, III...)
shListTypeLRoman	8	Números romanos em minúsculas (i, ii, iii...)

O código abaixo ilustra o uso do método **BeginItemList**.

```
Show.BeginItemList
```

```
Retorna: <OL TYPE="disc">
```

Aplica-se

Show

Veja também

Método Bold

Descrição

Marca o início de uma seção de texto com a formatação em negrito.

Sintaxe

objeto.**Bold**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .

Comentários

O código abaixo ilustra o uso do método **Bold**.

```
Show.Bold
```

Retorna:

Aplica-se
Show

Veja também

Método BoldEnd

Descrição

Finaliza a última formatação de texto iniciada com o método *Bold*.

Sintaxe

objeto.**BoldEnd**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .

Comentários

O código abaixo ilustra o uso do método **BoldEnd**.

```
Show.BoldEnd  
Retorna: </B>
```

Aplica-se
Show

Veja também

Método BoldText

Descrição

Apresenta um texto em negrito.

Sintaxe

objeto.**BoldText** *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .
<i>String</i>	Requerido. <i>String</i> , é que será apresentada em Bold

Comentários

O código abaixo ilustra o uso do método **BoldText**.

```
Show.BoldText "Teste do Text"
```

Retorna: Teste do Text

Aplica-se
Show

Veja também

Método BR

Descrição

Insere uma ou mais quebras de linha no documento.

Sintaxe

objeto.**BR** *Integer*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .
<i>Integer</i>	Opcional. Especifica quantas quebras de linha serão inseridas no documento. O valor padrão para esse parâmetro é 1 (um).

Comentários

O código abaixo ilustra o uso do método **BR**.

```
Show.BR  
Retorna: <BR>
```

Aplica-se
Show

Veja também

Método Center

Descrição

Marca o início de uma seção na qual os elementos do documento são centralizados.

Sintaxe

objeto.**Center**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .

Comentários

O código abaixo ilustra o uso do método **Center**.

Show.**Center**

Retorna: <CENTER>

Aplica-se

Show

Veja também

Método CenterEnd

Descrição

Finaliza a última centralização iniciada com o método *Center*.

Sintaxe

objeto.CenterEnd

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .

Comentários

O código abaixo ilustra o uso do método **CenterEnd**.

Show.**CenterEnd**

Retorna: </CENTER>

Aplica-se

Show

Veja também

Método CRHTML

Descrição

Apresenta uma quebra de linha (Cr e Lf) no código HTML, caso o valor da propriedade CrLf seja *verdadeiro*, em seguida incluindo no documento o texto especificado. Esse método ignora os valores atuais das propriedades *CrLfBefore* e *CrLfAfter*.

Sintaxe

objeto.CRHTML String

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .
<i>String</i>	Requerido. <i>String</i> que será apresentada.

Comentários

O código abaixo ilustra o uso do método **CRHTML**.

```
Show.CRHTML "<B>Mostra um HTML1</B>"
Show.CRHTML "<B>Mostra um HTML2</B>"
Retorna:
<B>Mostra um HTML1</B>
<B>Mostra um HTML2</B>
```

Aplica-se

Show

Veja também

Método CRHTMLCR

Descrição

Apresenta no documento um texto especificado, incluindo, antes e depois do texto, uma quebra de linha (Cr e Lf) no código HTML, caso o valor da propriedade *CrLf* seja *verdadeiro*. Esse método ignora os valores atuais das propriedades *CrLfBefore* e *CrLfAfter*.

Sintaxe

objeto.**CRHTMLCR** *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .
<i>String</i>	Requerido. <i>String</i> que será apresentada.

Comentários

O código abaixo ilustra o uso do método **CRHTMLCR**.

```
Show.CRHTMLCR "<B>Mostra um HTML1</B>"
Show.CRHTMLCR "<B>Mostra um HTML2</B>"
Retorna:
<B>Mostra um HTML1</B>

<B>Mostra um HTML2</B>
```

Aplica-se

Show

Veja também

Método EndItemList

Descrição

Finaliza uma lista iniciada com *BeginItemList*.

Sintaxe

objeto.**EndItemList**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .

Comentários

O código abaixo ilustra o uso do método **EndItemList**.

```
Show.EndItemList  
Retorna: </UL>
```

Aplica-se

Show

Veja também

Método HTML

Descrição

Apresenta um texto especificado no documento. Caso as propriedades *CrLfAfter* ou *CrLfBefore* tenham o valor *verdadeiro*, as quebras de linha correspondentes são acrescentadas.

Sintaxe

objeto.**HTML** *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .
<i>String</i>	Requerido. <i>String</i> que será apresentada.

Comentários

O código abaixo ilustra o uso do método **HTML**.

```
Show.HTML "<B>Mostra um HTML1</B>"  
Show.HTML "<B>Mostra um HTML2</B>"  
Retorna: <B>Mostra um HTML1</B><B>Mostra um HTML2</B>
```

Aplica-se

Show

Veja também

AutoWrite, CrLfBefore, CrLfAfter

Método HTMLComment

Descrição

Insere um comentário no código HTML, caso o valor da propriedade ShowComments seja *verdadeiro*. Os comentários não são exibidos no documento.

Sintaxe*objeto.HTMLComment String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .
<i>String</i>	Requerido. <i>String</i> que será apresentada como comentário HTML.

Comentários

O código abaixo ilustra o uso do método **HTMLComment**.

```
Show.HTMLComment "comentário no código"
```

```
Retorna: <!-- comentário no código -->
```

Aplica-se

Show

Veja também

Método HTMLCR

Descrição

Apresenta no documento um texto especificado, em seguida incluindo uma quebra de linha (Cr e Lf) no código HTML, caso o valor da propriedade CrLf seja *verdadeiro*. Esse método ignora os valores atuais das propriedades *CrLfBefore* e *CrLfAfter*.

Sintaxe*objeto.HTMLCR String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .
<i>String</i>	Requerido. <i>String</i> que será apresentada.

Comentários

O código abaixo ilustra o uso do método **HTMLCR**.

Objeto Show

```
Show.HTMLCR "<B>Mostra um HTML1</B>"
Show.HTMLCR "<B>Mostra um HTML2</B>"
Retorna: <B>Mostra um HTML1</B>
<B>Mostra um HTML2</B>
```

Aplica-se
Show

Veja também

Método Image

Descrição

Exibe uma imagem no documento.

Sintaxe

objeto.**Image** *String1, String2, String3, String4, String5, String6, String7*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .
<i>String1</i>	Requerido. URL, absoluta ou relativa, especificando a localização do arquivo da imagem.
<i>String2</i>	Opcional. ID da imagem. Esse atributo é geralmente utilizado com código DHTML.
<i>String3</i>	Opcional. Texto alternativo da imagem. Esse texto é exibido quando a imagem não for carregada pelo browser ou ao posicionar o cursor do mouse sobre a mesma.
<i>String4</i>	Opcional. Link da imagem. Especifica uma URL a ser aberta caso a imagem seja clicada.
<i>String5</i>	Opcional. Alinhamento horizontal da imagem em relação à página.
<i>String6</i>	Opcional. Largura da imagem.
<i>String7</i>	Opcional. Altura da imagem.

Comentários

O código abaixo ilustra o uso do método **Image**.

```
Show.Image "../mail.gif", , "Email", "mailto:webmaster@zyc.com.br"
Retorna: <A HREF="mailto:webmaster@zyc.com.br"><IMG SRC="../mail.gif"
ALT="Email" BORDER="0"></A>
```

Aplica-se
Show

Veja também

Método *Italic*

Descrição

Marca o início de uma seção de texto com a formatação em itálico.

Sintaxe

objeto.Italic

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .

Comentários

O código abaixo ilustra o uso do método **Italic**.

```
Show.Italic
```

Retorna: <I>

Aplica-se

Show

Veja também

Método *ItalicEnd*

Descrição

Finaliza a última formatação de texto iniciada com o método *Italic*.

Sintaxe

objeto.ItalicEnd

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .

Comentários

O código abaixo ilustra o uso do método **ItalicEnd**.

```
Show.ItalicEnd
```

Retorna: </I>

Aplica-se

Show

Veja também

Método ItalicText

Descrição

Apresenta um texto em itálico.

Sintaxe

objeto.ItalicText String

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .
<i>String</i>	Requerido. <i>String</i> , é que será apresentada em Italic

Comentários

O código abaixo ilustra o uso do método **ItalicText**.

```
Show.ItalicText "Teste do Text"
Retorna: <I>Teste do Text</I>
```

Aplica-se

Show

Veja também

Método ItemList

Descrição

Exibe um item em uma lista iniciada com *BeginItemList*.

Sintaxe

objeto.ItemList String1, String2

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .
<i>String1</i>	Requerido. Determina o título do item, o qual será exibido em negrito.
<i>String2</i>	Requerido. Determina o conteúdo do item, exibido à direita do título.

Comentários

O código abaixo ilustra o uso do método **ItemList**.

```
Show.ItemList "Aviso:", "valor indefinido"
```

Retorna: <TR><TD NOWRAP=NOWRAP>Aviso:</TD><TD>valor indefinido</TD></TR>

Aplica-se
Show

Veja também

Método ItemListError

Descrição

Exibe um item em uma lista iniciada com *BeginItemList*. Esse método é idêntico a *ItemList*, exceto por exibir o título na cor vermelha.

Sintaxe

objeto.ItemListError String1, String2

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .
<i>String1</i>	Requerido. Determina o título do item, o qual será exibido em negrito e vermelho.
<i>String2</i>	Requerido. Determina o conteúdo do item, exibido à direita do título.

Comentários

O código abaixo ilustra o uso do método **ItemListError**.

```
Show.ItemListError "Erro:", "arquivo não encontrado"
```

Retorna: <TR><TD NOWRAP=NOWRAP><FONT
COLOR=RED>Erro:</TD><TD>arquivo não encontrado</TD></TR>

Aplica-se
Show

Veja também

Método JSAlert

Descrição

Retorna uma janela com uma mensagem de alerta em JavaScript.

Sintaxe

objeto.JSAlert String

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .
<i>String</i>	Requerido. <i>String</i> que será apresentada a mensagem.

Comentários

O código abaixo ilustra o uso do método **JSAlert**.

```
Show.JSAlert "Mostra um HTML1"
```

```
Retorna: <SCRIPT LANGUAGE=JAVASCRIPT>alert("Mostra um HTML1");</SCRIPT>
```

Aplica-se

Show

Veja também

Método Link

Descrição

Inclui uma *tag* LINK, para inclusão de um arquivo externo no código atual.

Sintaxe

objeto.**Link** *String1*, *String2*, *String3*, *String4*, *String5*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .
<i>String1</i>	Requerido. <i>String1</i> que será <i>tag</i> do LINK.
<i>String2</i>	Requerido. <i>String2</i> que é o nome da <i>tag</i> LINK.
<i>String3</i>	Requerido. <i>String3</i> que é o HREF da <i>tag</i> LINK.
<i>String4</i>	Requerido. <i>String4</i> que é o TITLE da <i>tag</i> LINK.
<i>String5</i>	Requerido. <i>String5</i> que é o TYPE da <i>tag</i> LINK.

Comentários

O código abaixo ilustra o uso do método **Link**.

```
Show.Link "REL", "STYLESHEET", strHref, "", "text/css"
```

```
Retorna: <LINK REL="STYLESHEET" HREF="StyleSheet.css" TYPE="text/css">
```

Aplica-se

Show

Veja também

Método LinkRel

Descrição

Inclui uma *tag* LINK REL, para inclusão de um arquivo externo no código atual.

Sintaxe

objeto.LinkRel String1, String2, String3

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .
<i>String1</i>	Requerido. <i>String1</i> que é o nome da <i>tag</i> LINK.
<i>String2</i>	Requerido. <i>String2</i> que é o HREF da <i>tag</i> LINK.
<i>String3</i>	Requerido. <i>String3</i> que é o TITLE da <i>tag</i> LINK.

Comentários

O código abaixo ilustra o uso do método **LinkRev**.

```
Show.LinkRel "STYLESHEET", http://www.ztitools.com.br, "Título do REL"
Retorna: <LINK REL="STYLESHEET" HREF="http://www.ztitools.com.br"
TITLE="Título do REL" TYPE="text/css">
```

Aplica-se

Show

Veja também

Método LinkRelStyleSheet

Descrição

Inserir no HTML uma inclusão a um arquivo externo contendo especificação de formatação CSS.

Sintaxe

objeto.LinkRelStyleSheet String

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .
<i>String</i>	Requerido. URL, absoluta ou relativa, especificando o arquivo CCS a ser incluído.

Comentários

O código abaixo ilustra o uso do método **LinkRelStyleSheet**.

Show.**LinkRelStyleSheet** "site.css"

Retorna: <LINK REL="STYLESHEET" HREF="site.css" TYPE="text/css">

Aplica-se
Show

Veja também

Método LinkRev

Descrição

Inclui uma *tag* LINK REV, para referência externa do arquivo atual.

Sintaxe

objeto.**LinkRev** *String1*, *String2*, *String3*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .
<i>String1</i>	Requerido. <i>String1</i> que é o nome da <i>tag</i> LINK.
<i>String2</i>	Requerido. <i>String2</i> que é o HREF da <i>tag</i> LINK.
<i>String3</i>	Requerido. <i>String3</i> que é o TITLE da <i>tag</i> LINK.

Comentários

O código abaixo ilustra o uso do método **LinkRev**.

```
Show.LinkRev "STYLESHEET", http://www.ztitoools.com.br, "Título do REV"
```

Retorna: <LINK REV="STYLESHEET" HREF="http://www.ztitoools.com.br"
TITLE="Título do REV" TYPE="text/css">

Aplica-se
Show

Veja também

Método Message

Descrição

Retorna uma mensagem em HTML.

Sintaxe

objeto.**Message** *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .

<i>String</i>	Requerido. <i>String</i> que será apresentada a mensagem.
---------------	---

Comentários

O código abaixo ilustra o uso do método **Message**.

```
Show.Message "Mostra um HTML1"  
Retorna: <BR>Mostra um HTML1
```

Aplica-se

Show

Veja também

Método MessageAlert

Descrição

Retorna uma mensagem de alerta em HTML.

Sintaxe

objeto.MessageAlert String

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .
<i>String</i>	Requerido. <i>String</i> que será apresentada a mensagem.

Comentários

O código abaixo ilustra o uso do método **MessageAlert**.

```
Show.MessageAlert "Mostra um HTML1"  
Retorna: <BR><font color=orange><B>Mostra um HTML1</B></font>
```

Aplica-se

Show

Veja também

Método MessageError

Descrição

Retorna uma mensagem de erro em HTML.

Sintaxe

objeto.MessageError String

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .
<i>String</i>	Requerido. <i>String</i> que será apresentada a mensagem.

Comentários

O código abaixo ilustra o uso do método **MessageError**.

```
Show.MessageError "Mostra um HTML1"  
Retorna: <BR><font color=red><B>Mostra um HTML1</B></font>
```

Aplica-se

Show

Veja também

Método Meta

Descrição

Inclui uma *tag* META no código atual.

Sintaxe

objeto.Meta *String1*, *String2*, *String3*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .
<i>String1</i>	Requerido. <i>String1</i> que é o Field da <i>tag</i> META.
<i>String2</i>	Requerido. <i>String2</i> que é o Name da <i>tag</i> META.
<i>String3</i>	Requerido. <i>String3</i> que é o conteúdo da <i>tag</i> META.

Comentários

O código abaixo ilustra o uso do método **Meta**.

```
Show.Meta "HTTP-EQUIV", "Name", "http://www.ztitoools.com.br"  
Retorna: <META HTTP-EQUIV="Name", CONTENT="http://www.ztitoools.com.br">
```

Aplica-se

Show

Veja também

Método MetaHTTPEquiv

Descrição

Inclui uma *tag* META HTTP-EQUIV no código atual.

Sintaxe

objeto.**MetaHTTPEquiv** *String1*, *String2*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .
<i>String1</i>	Requerido. <i>String1</i> que é o Name da <i>tag</i> META.
<i>String2</i>	Requerido. <i>String2</i> que é o conteúdo da <i>tag</i> META.

Comentários

O código abaixo ilustra o uso do método **MetaHTTPEquiv**.

```
Show.MetaHTTPEquiv "Name", "http://www.ztitoools.com.br"
```

```
Retorna: <META HTTP-EQUIV="Name", CONTENT="http://www.ztitoools.com.br">
```

Aplica-se

Show

Veja também

Método MetaName

Descrição

Inclui uma *tag* META NAME no código atual.

Sintaxe

objeto.**MetaName** *String1*, *String2*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .
<i>String1</i>	Requerido. <i>String1</i> que é o Name da <i>tag</i> META.
<i>String2</i>	Requerido. <i>String2</i> que é o conteúdo da <i>tag</i> META.

Comentários

O código abaixo ilustra o uso do método **MetaName**.

```
Show.MetaName "Name", "http://www.ztitoools.com.br"
```

```
Retorna: <META NAME="Name", CONTENT="http://www.ztitoools.com.br">
```


Aplica-se
Show

Veja também

Método Underline

Descrição

Marca o início de uma seção de texto com a formatação sublinhado.

Sintaxe

objeto.**Underline**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .

Comentários

O código abaixo ilustra o uso do método **Underline**.

```
Show.Underline  
Retorna: <U>
```

Aplica-se
Show

Veja também

Método UnderlineEnd

Descrição

Finaliza a última formatação de texto iniciada com o método *Underline*.

Sintaxe

objeto.**UnderlineEnd**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .

Comentários

O código abaixo ilustra o uso do método **UnderlineEnd**.

```
Show.UnderlineEnd  
Retorna: </U>
```

Aplica-se
Show

Veja também

Método UnderlineText

Descrição

Apresenta um texto sublinhado.

Sintaxe

objeto.**UnderlineText***String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Show</i> .
<i>String</i>	Requerido. <i>String</i> , é que será apresentada em Underline

Comentários

O código abaixo ilustra o uso do método **UnderlineText**.

```
Show.UnderlineText "Teste do Text"
```

```
Retorna: <U>Teste do Text</U>
```

Aplica-se
Show

Veja também

Objeto String

Objeto String

O objeto String é constituído de um conjunto de métodos para formatação e manipulação de textos e seqüências de caracteres diversas.

Sintaxe

```
Set String = Server.CreateObject("HiperTools30.String")
```

Propriedade DateSeparator

Descrição

Especifica o separador usado pelas funções de manipulação de data.

Sintaxe

objeto.**DateSeparator** [= *String*]

O argumento *objeto* sempre será o objeto *String*.

Comentários

O código abaixo ilustra o uso da propriedade **DateSeparator**.

```
String.DateSeparator = "/"
```

Aplica-se

String

Veja também

Propriedade Piece

Descrição

Define a *String* que será acessada por partes. As funções de acesso a partes de *Strings* geralmente são usadas para importação de arquivos texto.

Sintaxe

objeto.**SetPiece** [= *String*]

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>String</i>	Requerido. <i>String</i> que será utilizada pelos os métodos de manipulação de pedaços.

Comentários

O código abaixo ilustra o uso da propriedade **Piece**.

```
Set objFSO = CreateObject("Scripting.FileSystemObject")
Set objFile = objFSO.OpenTextFile("c:\import.txt", ForReading, TristateFalse)
String.Piece = objFile.ReadLine
```

Aplica-se
String

Veja também

Método BreakString

Descrição

Retorna uma *String* até um determinado comprimento, acrescentando reticências caso a *String* seja truncada.

Sintaxe

objeto.**BreakString** *String*, *Integer*

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>String</i>	Requerido. Texto a ser truncado.
<i>Integer</i>	Requerido. Tamanho máximo para o texto.

Comentários

O código abaixo ilustra o uso do método **BreakString**.

```
strValorA = String.BreakString("Teste de quebra", 5)  
Retorna: Teste...
```

Aplica-se
String

Veja também

Método DayOfWeek

Descrição

Retorna o nome do dia.

Sintaxe

objeto.**DayOfWeek** *Date*, [*Integer*, [*Boolean*]]

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>Date</i>	Requerido. Data de referência para determinação do dia.
<i>Integer</i>	Opcional. Número de caracteres a ser retornado.

<i>Boolean</i>	Opcional. Valor que especifica se o nome do dia será retornado com o sufixo "feira". Esse parâmetro é ignorado caso o número de caracteres a ser retornado seja especificado.
----------------	---

Comentários

O código abaixo ilustra o uso do método **DayOfWeek**.

```
strValor = String.DayOfWeek(Now(),,True)
Retorna: Segunda-feira

strValor = String.DayOfWeek(Now(), 3, True)
Retorna: Seg
```

Aplica-se

String

Veja também

Método FormatCGC

Descrição

Retorna uma string formatada como CGG (999.999.999/9999-99).

Sintaxe

objeto.**FormatCGC** String

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>String</i>	Requerido. <i>String</i> que será formatada como CGC.

Comentários

O código abaixo ilustra o uso do método **FormatCGC**.

```
strValor = "123456789000123"

strValorA = String.FormatCGC(strValor)
Retorna: 123.456.789/0001-23
```

Aplica-se

String

Veja também

Método FormatCPF

Descrição

Retorna uma string formatada como CPF (999.999.999-99).

Sintaxe

objeto.FormatCPF String

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>String</i>	Requerido. <i>String</i> , que será formatada como CPF.

Comentários

O código abaixo ilustra o uso do método **FormatCPF**.

```
strValor = "12345678923"
```

```
strValorA = String.FormatCPF(strValor)
```

```
Retorna: 123.456.789-23
```

Aplica-se

String

Veja também

Método FormatDigit

Descrição

Retorna uma string formatada como valor incluindo dígito verificador (999.999-9).

Sintaxe

objeto.FormatDigit String, [Integer]

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>Integer</i>	Opcional. Especifica quantos dígitos estarão após o traço. O valor padrão para esse parâmetro é 1 (um).
<i>String</i>	Requerido. <i>String</i> , que será formatada com dígito verificador.

Comentários

O código abaixo ilustra o uso do método **FormatDigit**.

```
strValor = "12345678"

strValorA = String.FormatDigit(strValor)
Retorna: 1.234.567-8
```

Aplica-se

String

Veja também

Método FormatInt

Descrição

Retorna o número apresentando-o com formatação de grupos, sem casas decimais (9.999).

Sintaxe

objeto.**FormatInt** *Long*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>Number</i>	Requerido. Número a ser formatado.

Comentários

O código abaixo ilustra o uso do método **FormatInt**.

```
intValor = 32123

Response.Write String.FormatInt(intValor)
Retorna: 32.123
```

Aplica-se

String

Veja também

Método FormatMoney

Descrição

Retorna o número apresentando-o formatado como moeda (R\$ 9.999,99).

Sintaxe

objeto.**FormatMoney** *Currency*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>Currency</i>	Requerido. Número a ser formatado.

Comentários

O código abaixo ilustra o uso do método **FormatMoney**.

```
intValor = 32123.45
```

```
Response.Write String.FormatMoney(intValor)
```

```
Retorna: R$ 32.123,45
```

Aplica-se

String

Veja também

Método FormatNumber

Descrição

Retorna o número apresentando-o com formatação de grupos e casas decimais (9.999,99).

Sintaxe

objeto.FormatNumber Double

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>Double</i>	Requerido. Número a ser formatado.

Comentários

O código abaixo ilustra o uso do método **FormatNumber**.

```
intValor = 32123.45
```

```
Response.Write String.FormatNumber(intValor)
```

```
Retorna: 32.123,45
```

Aplica-se

String

Veja também

Método FormatPhone

Descrição

Retorna uma *String* formatada como telefone. O formato dependerá da quantidade de caracteres do parâmetro.

Sintaxe

objeto.**FormatPhone** *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>String</i>	Requerido. <i>String</i> , que será formatada como telefone.

Tamanho	Formato
7	###-####
8	####-####
9	(0_##) ###-####
10	(###) ###-####
11	(###) ####-####

Comentários

O código abaixo ilustra o uso do método **FormatPhone**.

```
strValorA = String.FormatPhone("123 1234")  
Retorna: 123-1234
```

```
strValorA = String.FormatPhone("1231 2345")  
Retorna: 1231-2345
```

```
strValorA = String.FormatPhone("(12) 3123456")  
Retorna: (0_12) 312-3456
```

```
strValorA = String.FormatPhone("(123) 1234567")  
Retorna: (123) 123-4567
```

```
strValorA = String.FormatPhone("123 1234 5678")  
Retorna: (123) 1234-5678
```

Aplica-se

String

Veja também

Método GetPiece

Descrição

Retorna um determinado número de caracteres a partir da posição atual e reposiciona a posição após o último caractere lido.

Sintaxe

objeto.**GetPiece** *Integer*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>Integer</i>	Requerido. Número de caracteres a serem lidos.

Comentários

O código abaixo ilustra o uso do método **GetPiece**.

```
String.Piece = "Zevallos Tecnologia em Informação"
```

```
Response.Write String.GetPiece(10)
```

```
Retorna: Zevallos T
```

```
Response.Write String.GetPiece(10)
```

```
Retorna: ecnologia
```

Aplica-se

String

Veja também

Método GetPieceAsDate

Descrição

Retorna um determinado número de caracteres a partir da posição atual e reposiciona a posição após o último caractere lido. O resultado é convertido para o tipo *Date*.

Sintaxe

objeto.**GetPieceAsDate** *Integer*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>Integer</i>	Requerido. Número de caracteres a serem lidos como <i>Date</i> .

Comentários

O código abaixo ilustra o uso do método **GetPieceAsDate**.

```
String.Piece = "Zevallos Tecnologia em Informação-10/10/99"
```

```
Response.Write String.GetPieceAsStr(45)  
Retorna: Zevallos Tecnologia em Informação-
```

```
Response.Write String.GetPieceAsDate(8)  
Retorna: 10/10/99
```

Aplica-se

String

Veja também

Método GetPieceAsStr

Descrição

Retorna um determinado número de caracteres a partir da posição atual e reposiciona a posição após o último caractere lido. A *String* resultante é delimitada por aspas simples.

Sintaxe

objeto.**GetPieceAsStr** *Integer*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>Integer</i>	Requerido. Número de caracteres a serem lidos.

Comentários

O código abaixo ilustra o uso do método **GetPieceAsStr**.

```
String.Piece = "Zevallos Tecnologia em Informação"
```

```
Response.Write String.GetPieceAsStr(10)  
Retorna: 'Zevallos T'  
Response.Write String.GetPieceAsStr(10)  
Retorna: 'ecnologia'
```

Aplica-se

String

Veja também

Método GetPieceRest

Descrição

Retorna a porção da *String* definida na propriedade *Piece* que ainda não foi recuperada.

Sintaxe

objeto.**GetPieceRest**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .

Comentários

O código abaixo ilustra o uso do método **GetPieceRest**.

```
String.Piece = "Zevallos Tecnologia em Informação"
String.GetPiece(9)
strPiece = String.GetPieceRest
```

Retorna: Tecnologia em Informação

Aplica-se

String

Veja também

Método GetWord

Descrição

Retorna uma palavra de acordo com sua posição num texto.

Sintaxe

objeto.**GetWord** *String*, *Integer*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>String</i>	Requerido. Frase de onde determinar a palavra.
<i>Integer</i>	Requerido. Índice da palavra. A primeira palavra tem o índice 1, a segunda tem índice 2 e assim por diante.

Comentários

O código abaixo ilustra o uso do método **GetWord**.

```
String.SetPiece "Zevallos Tecnologia em Informação - 10/10/99"
```

```
Response.Write String.GetWord(2)  
Retorna: Tecnologia
```

Aplica-se

String

Veja também

Método ImportDate

Descrição

Converte uma *String* no formato especificado para uma variável do tipo *Date*.

Sintaxe

objeto.ImportDate String , Integer

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>String</i>	Requerido. <i>String</i> com os dados a serem convertidos.
<i>Integer</i>	Requerido. Formato dos dados.

Comentários

O formato deve ser uma das constantes a seguir:

Constante	Valor	Descrição
conDateTypeAAMMDD	0	Formato AAMMDD
conDateTypeAAAAMMDD	1	Formato AAAAMMDD
conDateTypeMMDDAA	2	Formato MMDDAA
conDateTypeMMDDAAAA	3	Formato MMDDAAAA
conDateTypeDDMMAA	4	Formato DDMMAA
conDateTypeDDMMAAAA	5	Formato DDMMAAAA

O código abaixo ilustra o uso do método **ImportDate**.

```
strData = "660823"  
  
Response.Write String.ImportDate(strData, conDateTypeAAMMDD)  
Retorna: 08/23/66
```


Aplica-se
String

Veja também

Método ImportMoney

Descrição

Converte uma *String* no formato especificado para uma variável do tipo *Currency*.

Sintaxe

objeto.ImportMoney String, Integer

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>String</i>	Requerido. <i>String</i> com os dados a serem convertidos.
<i>Integer</i>	Requerido. Formato dos dados.

Comentários

O formato deve ser uma das constantes a seguir:

Constante	Valor	Descrição
<code>conMoneyType9999D99</code>	0	Formato 9999,99
<code>conMoneyType9D999D99</code>	1	Formato 9.999,99
<code>conMoneyType9999</code>	2	Formato 9999

O código abaixo ilustra o uso do método **ImportMoney**.

```
strValor = "12345.54"
```

```
Response.Write String.ImportMoney(strValor, conMoneyType9999D99)  
Retorna: 12345.54
```

Aplica-se
String

Veja também

Método ImportInt

Descrição

Converte uma *String* para uma variável do tipo *Int*.

Sintaxe

objeto.ImportInt String

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>String</i>	Requerido. <i>String</i> com os dados a serem convertidos.

Comentários

O código abaixo ilustra o uso do método **ImportInt**.

```
strValor = "123.45"

Response.Write String.ImportInt(strValor)
Retorna: 123
```

Aplica-se

String

Veja também

Método ImportLong

Descrição

Converte uma *String* para uma variável do tipo *Long*.

Sintaxe

objeto.ImportLong String

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>String</i>	Requerido. <i>String</i> é o número que será convertido.

Comentários

O código abaixo ilustra o uso do método **ImportLong**.

```
strValor = "123.45"

Response.Write String.ImportLong(strValor)
Retorna: 123
```

Aplica-se
String

Veja também

Método LeadingZeroes

Descrição

Acrescenta zeros à esquerda de um número.

Sintaxe

objeto.**LeadingZeroes** *String, Integer*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>String</i>	Requerido. <i>String</i> é o número ao qual serão acrescentados zeros.
<i>Integer</i>	Requerido. <i>Integer</i> é o tamanho mínimo do número resultante.

Comentários

O código abaixo ilustra o uso do método **LeadingZeroes**.

```
strValor = "123"

Response.Write String.LeadingZeroes(strValor, 6)
Retorna: 000123
```

Aplica-se
String

Veja também

Método LongDate

Descrição

Retorna uma *String* como uma data no formato 99/99/9999.

Sintaxe

objeto.**LongDate** *Variant*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .

<i>Variant</i>	Requerido. Valor representando a data a ser formatada.
----------------	--

Comentários

O código abaixo ilustra o uso do método **LongDate**.

```
datValor = Now()  
  
Response.Write String.LongDate(datValor)  
Retorna: 23/08/1966
```

Aplica-se

String

Veja também

Método LongDateTime

Descrição

Retorna uma *String* como uma data/hora no formato 99/99/9999 99:99:99.

Sintaxe

objeto.LongDateTime *Variant*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>Variant</i>	Requerido. Valor representando a data/hora a ser formatada.

Comentários

O código abaixo ilustra o uso do método **LongDateTime**.

```
datValor = Now()  
  
Response.Write String.LongDateTime(datValor)  
Retorna: 23/08/1966 13:00:00
```

Aplica-se

String

Veja também

Método LongTime

Descrição

Retorna uma *String* como uma hora no formato HH:MM:SS

Sintaxe

objeto.LongTime Variant

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>Variant</i>	Requerido. Valor representando a hora a ser formatada.

Comentários

O código abaixo ilustra o uso do método **LongTime**.

```
datValor = Now()  
  
Response.Write String.LongTime(datValor)  
Retorna: 13:00:00
```

Aplica-se

String

Veja também

Método LTrimZeros

Descrição

Retira os zeros à esquerda de uma *String* representando um número.

Sintaxe

objeto.LTrimZeros String

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>String</i>	Requerido. <i>String</i> é o número que serão retirados os zeros.

Comentários

O código abaixo ilustra o uso do método **LTrimZeros**.

```
strValor = "123"  
  
strValor = String.LTrimZeros(strValor, 6)  
Retorna: 000123
```

```
Response.Write String.LtrimZeroes(strValor)  
Retorna: 123
```

Aplica-se

String

Veja também

Método Mask

Descrição

Retorna uma *String* formatada de acordo com uma máscara especificada.

Sintaxe

objeto.Mask String1, String2, String3

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>String1</i>	Requerido. Valor a ser formatado.
<i>String2</i>	Requerido. Máscara a ser aplicada.
<i>String3</i>	Requerido. Complemento da máscara.

Comentários

A máscara deve ser composta de caracteres "0" (zero) para representar números e caracteres "A" para representar letras. Os demais caracteres são colocados na máscara como forem digitados.

O código abaixo ilustra o uso do método **Mask**.

```
strValorA = String.Mask("123456789000155", "000.000.000/0000-00", "0")  
Retorna: 123.456.789/0001-55
```

Aplica-se

String

Veja também

Método MonthName

Descrição

Retorna o nome de um mês.

Sintaxe

objeto.MonthName Integer

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>Integer</i>	Requerido. Valor especificando o índice do mês, conforme a tabela abaixo.

Valor	Descrição
1	Janeiro
2	Fevereiro
3	Março
4	Abril
5	Maio
6	Junho
7	Julho
8	Agosto
9	Setembro
10	Outubro
11	Novembro
12	Dezembro

Comentários

O código abaixo ilustra o uso do método **MonthName**.

```
strValorA = String.MonthName(1)  
Retorna: Janeiro
```

```
strValorA = String.MonthName(Month(Now()))  
Retorna: Março
```

Aplica-se

String

Veja também

Método MonthSize

Descrição

Retorna o número de dias de um mês no ano especificado.

Sintaxe

objeto.**MonthSize** *Integer1*, *Integer2*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>Integer1</i>	Requerido. Valor especificando o índice do mês.
<i>Integer2</i>	Opcional. Ano usado como referência para determinação do número de dias no mês. Por padrão esse parâmetro assume o ano atual.

Comentários

O código abaixo ilustra o uso do método **MonthSize**.

```
strValorA = String.MonthSize(1)
Retorna: 30

strValorA = String.MonthSize(2, 1999)
Retorna: 28

strValorA = String.MonthSize(2, 2000)
Retorna: 29
```

Aplica-se

String

Veja também

Método NormalizeAccents

Descrição

Retorna uma string removendo a acentuação.

Sintaxe

objeto.**NormalizeAccents**(*String*)

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>String</i>	Requerido. <i>String</i> , que será convertida

Comentários

O código abaixo ilustra o uso do método **NormalizeAccents**.


```
strValorA = String.NormalizeAccents("Teste ãöüçÇ")  
Retorna: Teste aooucC
```

Aplica-se
String

Veja também

Método NormalizeNumber

Descrição

Retorna uma string representando um número válido, removendo caracteres especiais ou alfabéticos.

Sintaxe

objeto.NormalizeNumber String

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>String</i>	Requerido. <i>String</i> , que será normalizado.

Comentários

O código abaixo ilustra o uso do método **TotalNormalizeString**.

```
strValor = "String para o SQL:??<>1234"  
  
strValorA = String.NormalizeNumber(strValor)  
Retorna: 1234
```

Aplica-se
String

Veja também

Método NormalizeString

Descrição

Retorna uma string normalizada, contendo somente caracteres alfanuméricos.

Sintaxe

objeto.NormalizeString String

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>String</i>	Requerido. <i>String</i> , que será normalizado.

Comentários

O código abaixo ilustra o uso do método **NormalizeString**.

```
strValor = "String para o SQL:><>1234"

strValorA = String.NormalizeString(strValor)
Retorna: StringparaoSQL1234
```

Aplica-se

String

Veja também

Método Reverse

Descrição

Retorna uma seqüência de caracteres invertida.

Sintaxe

objeto.**Reverse** *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>String</i>	Requerido. <i>String</i> , que será invertida.

Comentários

O código abaixo ilustra o uso do método **Reverse**.

```
strValorA = String.Reverse("123456")
Retorna: 654321

strValorA = String.Reverse("Valor que sera invertido")
Retorna: oditrevni ares euq rolaV
```

Aplica-se

String

Veja também

Método SecondsAsTime

Descrição

Converte um valor especificando a quantidade de segundos transcorridos a partir da zero hora em uma variável do tipo *Date*.

Sintaxe

objeto.SecondsAsTime Integer

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>Integer</i>	Requerido. Número de segundos transcorridos.

Comentários

O código abaixo ilustra o uso do método **SecondsAsTime**.

```
datValor = Now()  
  
intValor = String.TimeAsSeconds(datValor)  
Retorna: 46800  
Response.Write String.SecondsAsTime(intValor)  
Retorna: 13:00:00
```

Aplica-se

String

Veja também

Método ShortTime

Descrição

Retorna uma *String* como uma hora no formato HH:MM.

Sintaxe

objeto.ShortTime Variant

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>Variant</i>	Requerido. Valor representando a hora a ser formatada.

Comentários

O código abaixo ilustra o uso do método **ShortTime**.

```
datValor = Now()
```

```
Response.Write String.ShortTime(datValor)
Retorna: 13:00
```

Aplica-se

String

Veja também

Método ShowPiece

Descrição

Apresenta a porção da *String* definida na propriedade *Piece* que ainda não foi recuperada. Esse método geralmente é usado para debug.

Sintaxe

objeto.**ShowPieceRest**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto String.

Comentários

O código abaixo ilustra o uso do método **ShowPieceRest**.

```
String.Piece = "Zevallos Tecnologia em Informação"
String.GetPiece(9)
String.ShowPieceRest
```

Retorna: Tecnologia em Informação

Aplica-se

String

Veja também

Método SkipPiece

Descrição

Avança a posição do próximo *GetPiece*.

Sintaxe

objeto.**SkipPiece** *Integer*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .

<i>Integer</i>	Requerido. Número de caracteres a serem avançados.
----------------	--

Comentários

O código abaixo ilustra o uso do método **SkipPiece**.

```
String.SetPiece("Zevallos Tecnologia em Informação-10/10/99")
Response.Write String.SkipPiece(45)
Response.Write String.GetPieceAsDate(8)
Retorna: 10/10/99
```

Aplica-se

String

Veja também

Método SQLLongDate

Descrição

Retorna uma data no formato MMDDAAAA e delimitada por aspas simples.

Sintaxe

objeto.**SQLLongDate** *Variant*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>Variant</i>	Requerido. Data a ser formatada.

Comentários

O código abaixo ilustra o uso do método **SQLLongDate**.

```
datValor = Now()

intValor = String.SQLLongDate(datValor)
Retorna: 08/23/1966
```

Aplica-se

String

Veja também

Método SQLLongDateBR

Descrição

Retorna uma data no formato DDMMAAAA e delimitada por aspas simples.

Sintaxe

objeto.SQLLongDateBR Variant

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>Variant</i>	Requerido. Data a ser formatada.

Comentários

O código abaixo ilustra o uso do método **SQLLongDateBR**.

```
datValor = "23/08/66"  
intValor = String.SQLLongDateBR(datValor)  
Retorna: 08/23/1966
```

Aplica-se

String

Veja também

Método SQLDateTime

Descrição

Retorna uma data/hora no formato MMDDAAAA HH:MM:SS e delimitada por aspas simples.

Sintaxe

objeto.SQLDateTime Variant

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>Variant</i>	Requerido. Data a ser formatada.

Comentários

O código abaixo ilustra o uso do método **SQLDateTime**.

```
datValor = Now()  
  
intValor = String.SQLDateTime(datValor)  
Retorna: '08/23/1966 13:00:00'
```

Aplica-se

String

Veja também

Método SQLLong

Descrição

Efetua a conversão de um valor para ser usado como um *Long* em uma consulta SQL.

Sintaxe

objeto.**SQLLong** *Variant*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>Variant</i>	Requerido. Valor a ser convertido como <i>Long</i> .

Comentários

O código abaixo ilustra o uso do método **SQLLong**.

```
lgnValor = "123454"
```

```
intValor = String.SQLLong(lgnValor)
```

```
Retorna: 123454
```

Aplica-se

String

Veja também

Método SQLString

Descrição

Remove os espaços em branco do início e do fim de uma *String*, delimitando a mesma com aspas simples.

Sintaxe

objeto.**SQLString** *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>String</i>	Requerido. Valor que será usado como <i>String</i> numa consulta SQL.

Comentários

O código abaixo ilustra o uso do método **SQLString**.

```
strValor = "String para o SQL"

strValorA = String.SQLString(strValor)
Retorna: 'String para o SQL'
```

Aplica-se

String

Veja também

Método TimeAsSeconds

Descrição

Retorna o número de segundos transcorridos entre a zero hora e a hora especificada num valor do tipo *Date*.

Sintaxe

objeto.**TimeAsSeconds** *Date*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>Date</i>	Requerido. Data de referência.

Comentários

O código abaixo ilustra o uso do método **TimeAsSeconds**.

```
datValor = Now()

Response.Write String.TimeAsSeconds(datValor)
Retorna: 46800
```

Aplica-se

String

Veja também

Método ToString

Descrição

Converte um valor compatível com o tipo *Variant* para *String*.

Sintaxe

objeto.**ToString** *Variant*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>Variant</i>	Requerido. <i>Valor</i> que convertido para <i>String</i> .

Comentários

O código abaixo ilustra o uso do método **ToString**.

```
strValorA = String.ToString(12345)
Retorna: "12345"
```

Aplica-se

String

Veja também

Método USADate

Descrição

Retorna uma data no formato MMDDAAAA.

Sintaxe

objeto.**USADate** *Variant*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>String</i> .
<i>Variant</i>	Requerido. Data a ser formatada.

Comentários

O código abaixo ilustra o uso do método **USADate**.

```
datValor = Now()

Response.Write String.USADate(datValor)
Retorna: 08/23/1966
```

Aplica-se

String

Veja também

Objeto Table

Objeto Table

Esse objeto permite a montagem de tabelas, fornecendo várias opções de apresentação e formatação dos dados.

Sintaxe

```
Set Table = Server.CreateObject("HiperTools30.Table")
```

Propriedade CellAlign

Descrição

Especifica o alinhamento do texto de todas as células da tabela posteriormente definidas.

Sintaxe

```
objeto.CellAlign [= String]
```

O argumento *objeto* sempre será o objeto *Table*.

Comentários

O código abaixo ilustra o uso da propriedade **CellAlign**.

```
Table.CellAlign = "center"  
Table.Cell "Centralizado"  
Table.Cell "Tambem centralizado"  
Table.CellAlign = ""  
Table.Cell "Padrao"
```

Aplica-se

Table

Veja também

Propriedade CellColSpan

Descrição

Combina horizontalmente todas as células da tabela posteriormente definidas. As células ocupam a largura correspondente ao número de colunas especificado.

Sintaxe

```
objeto.CellColSpan [= String]
```

O argumento *objeto* sempre será o objeto *Table*.

Comentários

O código abaixo ilustra o uso da propriedade **CellColSpan**.

```
Table.CellColSpan = "2"  
Table.BeginRow  
Table.Cell "Celula ocupando duas colunas"  
Table.EndRow  
Table.CellColSpan = ""  
Table.BeginRow
```

```
Table.Cell "Celula ocupando uma coluna"
Table.Cell "Celula ocupando a outra coluna"
Table.EndRow
```

Aplica-se
Table

Veja também

Propriedade CellHeight

Descrição

Especifica a altura de todas as células da tabela posteriormente definidas.

Sintaxe

objeto.**CellHeight** [= *String*]

O argumento *objeto* sempre será o objeto *Table*.

Comentários

Os valores válidos para essa propriedade são os mesmos do HTML, especificado em pixels ou porcentagem.

O código abaixo ilustra o uso da propriedade **CellHeight**.

```
Table.CellHeight = "30%"
Table.Cell "Em porcentagem"
Table.CellHeight = "55"
Table.Cell "Fixo em pixels"
Table.CellHeight = ""
Table.Cell "Padrao"
```

Aplica-se
Table

Veja também

Propriedade CellNoWrap

Descrição

Especifica se o texto de todas as células da tabela posteriormente definidas não poderá ser quebrado em várias linhas caso o espaço ocupado pelo texto seja maior que a largura da célula.

Sintaxe

objeto.**CellNoWrap** [= *Boolean*]

O argumento *objeto* sempre será o objeto *Table*.

Comentários

O código abaixo ilustra o uso da propriedade **CellNoWrap**.

```
Table.CellNoWrap = True
```

Aplica-se
Table

Veja também

Propriedade CellRowSpan

Descrição

Combina verticalmente todas as células da tabela posteriormente definidas. As células ocupam a altura correspondente ao número de linhas especificado.

Sintaxe

```
objeto.CellRowSpan [= String]
```

O argumento *objeto* sempre será o objeto *Table*.

Comentários

O código abaixo ilustra o uso da propriedade **CellRowSpan**.

```
Table.CellRowSpan = "2"  
Table.BeginRow  
Table.Cell "Celula ocupando duas linhas"  
Table.CellRowSpan = ""  
Table.Cell "Celula ocupando uma linha"  
Table.Cell "Celula ocupando a outra linha"  
Table.EndRow
```

Aplica-se
Table

Veja também

Propriedade CellVAlign

Descrição

Especifica o alinhamento vertical do texto de todas as células da tabela posteriormente definidas.

Sintaxe

```
objeto.CellVAlign [= String]
```

O argumento *objeto* sempre será o objeto *Table*.

Comentários

O código abaixo ilustra o uso da propriedade **CellVAlign**.

```
Table.CellVAlign = "top"  
Table.Cell "Texto acima"  
Table.CellVAlign = ""  
Table.Cell "Padrao"
```

Aplica-se

Table

Veja também

Propriedade CellWidth

Descrição

Especifica a largura de todas as células da tabela posteriormente definidas.

Sintaxe

objeto.**CellWidth** [= *String*]

O argumento *objeto* sempre será o objeto *Table*.

Comentários

Os valores válidos para essa propriedade são os mesmos do HTML, especificado em pixels ou porcentagem.

O código abaixo ilustra o uso da propriedade **CellWidth**.

```
Table.CellWidth = "30%"  
Table.Cell "Em porcentagem"  
Table.CellWidth = "55"  
Table.Cell "Fixo em pixels"  
Table.CellWidth = ""  
Table.Cell "Padrao"
```

Aplica-se

Table

Veja também

Propriedade GraphMax

Descrição

Especifica o valor máximo que ocorre em gráficos de barra. Essa propriedade é útil caso se deseje mostrar gráficos sem legenda.

Sintaxe

objeto.**GraphMax** [= *Double*]

O argumento *objeto* sempre será o objeto *Table*.

Comentários

O código abaixo ilustra o uso da propriedade **GraphMax**.

```
Table.GraphMax = 16000
```

Aplica-se
Table

Veja também
LegendInfo

Propriedade GraphTotal

Descrição

Especifica o valor total, resultado da soma dos valores de todos os gráficos de barra apresentados. Essa propriedade é útil caso se deseje mostrar gráficos sem legenda.

Sintaxe

objeto.**GraphTotal** [= *Double*]

O argumento *objeto* sempre será o objeto *Table*.

Comentários

O código abaixo ilustra o uso da propriedade **GraphTotal**.

```
Table.GraphTotal = 6000
```

Aplica-se
Table

Veja também
LegendInfo

Propriedade Padding

Descrição

Especifica o espaço interno entre os textos das células e o limite das mesmas.

Sintaxe

objeto.**Padding** [= *String*]

O argumento *objeto* sempre será o objeto *Table*.

Comentários

Essa é uma propriedade que se aplica a todas as células da tabela, sendo que alterações no seu valor surtem efeito apenas nas tabelas subsequentes.

O código abaixo ilustra o uso da propriedade **Padding**.

```
Table.Padding = "2"
```

Aplica-se
Table

Veja também

Propriedade ShowBorder

Descrição

Especifica se a borda definida para a tabela será exibida ou não.

Sintaxe

objeto.**ShowBorder** [= *Boolean*]

O argumento *objeto* sempre será o objeto *Table*.

Comentários

O código abaixo ilustra o uso da propriedade **ShowBorder**.

```
Table.ShowBorder = True
```

Aplica-se

Table

Veja também

Propriedade Spacing

Descrição

Especifica o espaço externo entre as células.

Sintaxe

objeto.**Spacing** [= *String*]

O argumento *objeto* sempre será o objeto *Table*.

Comentários

O código abaixo ilustra o uso da propriedade **Spacing**.

```
Table.Spacing = "3"
```

Aplica-se

Table

Veja também

Propriedade Style

Descrição

Essa propriedade é um objeto *TableStyle*, usado para modificar a formatação de vários itens da tabela.

Sintaxe

objeto.**Style** [= *TableStyle*]

O argumento *objeto* sempre será o objeto *Table*.

Comentários

O código abaixo ilustra o uso da propriedade **Style**.

```
Table.Style.TitleFont.Bold = True
```

Aplica-se
Table

Veja também
TableStyle

Método BeginCell

Descrição

Inicia uma célula da tabela.

Sintaxe

objeto.**BeginCell**

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Table</i> .

Comentários

O código abaixo ilustra o uso do método **BeginCell**.

```
Table.Padding = 4
Table.Spacing = 0
Table.BeginTable "65%", ,2 ,True

Table.CellColSpan = 2
Table.BeginRow 3, True
Table.Cell "Painel de Controle"
Table.EndRow

Table.BeginRow 2
Table.BeginCell
URL.BeginURL Initializer.ScriptName

URL.Add efQueryStrAction, efQSAActionEditor
URL.Add efQueryStrWhat, conWhatChamaTecnico
URL.Show "Chamado Técnico", efQueryStrEditableStr
Show.BR
URL.EndURL
Table.EndCell
Table.EndRow
Table.EndTable
```

Aplica-se
Table

Veja também

Método BeginRow

Descrição

Inicia uma linha da tabela, com a possibilidade de definir o tamanho do fonte e se o a linha será um cabeçalho.

Sintaxe

objeto.**BeginRow** [*Integer*], [*Boolean*]

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Table</i> .
<i>Integer</i>	Opcional. Tamanho da fonte na linha. Os valores válidos para essa propriedade são os mesmos do HTML. O padrão para essa propriedade é 0 (zero).
<i>Boolean</i>	Opcional. Especifica se a linha será um cabeçalho. Como padrão as linhas não são exibidas como cabeçalho.

Comentários

O código abaixo ilustra o uso do método **BeginRow**.

```
Table.Padding = 4
Table.Spacing = 0
Table.BeginTable "65%", ,2 ,True

Table.CellColSpan = 2
Table.BeginRow 3, True
Table.Cell "Painel de Controle"
Table.EndRow

Table.BeginRow 2
Table.BeginCell
URL.BeginURL Initializer.ScriptName

URL.Add efQueryStrAction, efQSActionEditor
URL.Add efQueryStrWhat, conWhatChamaTecnico
URL.Show "Chamado Técnico", efQueryStrEditableStr
Show.BR
URL.EndURL
Table.EndCell
Table.EndRow
Table.EndTable
```

Aplica-se

Table

Veja também

Método BeginTable

Descrição

Inicia a apresentação de uma tabela.

Sintaxe

objeto.**BeginTable** [*String1*], [*String2*], [*Integer*], [*Boolean*], [*String3*]

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Table</i> .
<i>String1</i>	Opcional. Largura mínima da tabela. Os valores válidos para essa propriedade são os mesmos do HTML, especificado em porcentagem ou pixels.
<i>String2</i>	Opcional. Título a ser exibido na tabela.
<i>Integer</i>	Opcional. Número de colunas que a tabela terá. O valor padrão para essa propriedade é 1 (um).
<i>Boolean</i>	Opcional. Especifica se a tabela terá borda. Como padrão não há borda.
<i>String3</i>	Opcional. Altura da mínima da tabela. Os valores válidos para essa propriedade são os mesmos do HTML, especificado em porcentagem ou pixels.

Comentários

O código abaixo ilustra o uso do método **BeginTable**.

```
Table.Padding = 4
Table.Spacing = 0
Table.BeginTable "65%", ,2 ,True

Table.CellColSpan = 2
Table.BeginRow 3, True
Table.Cell "Painel de Controle"
Table.EndRow

Table.BeginRow 2
Table.BeginCell
URL.BeginURL Initializer.ScriptName

URL.Add efQueryStrAction, efQSAActionEditor
URL.Add efQueryStrWhat, conWhatChamaTecnico
URL.Show "Chamado Técnico", efQueryStrEditableStr
Show.BR
```

```
URL.EndURL  
Table.EndCell  
Table.EndRow  
Table.EndTable
```

Aplica-se

Table

Veja também

Método Cell

Descrição

Inicia uma célula, exibe seu conteúdo e a finaliza.

Sintaxe

objeto.**Cell** *String*

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Table</i> .
<i>String</i>	Requerido. Conteúdo da célula.

Comentários

O código abaixo ilustra o uso do método **Cell**.

```
Table.Padding = 4  
Table.Spacing = 0  
Table.BeginTable "65%", ,2 ,True  
  
Table.CellColSpan = 2  
Table.BeginRow 3, True  
Table.Cell "Painel de Controle"  
Table.EndRow  
  
Table.BeginRow 2  
Table.BeginCell  
URL.BeginURL Initializer.ScriptName  
  
URL.Add efQueryStrAction, efQSActionEditor  
URL.Add efQueryStrWhat, conWhatChamaTecnico  
URL.Show "Chamado Técnico", efQueryStrEditableStr  
Show.BR  
URL.EndURL  
Table.EndCell
```

Table.EndRow
Table.EndTable

Aplica-se
Table

Veja também

Método CellGraph

Descrição

Apresenta uma célula cujo conteúdo é um gráfico.

Sintaxe

objeto.**CellGraph** *Double*, [*String*]

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Table</i> .
<i>Double</i>	Requerido. Valor do percentual do gráfico, em relação ao valor máximo.
<i>String</i>	Opcional. Texto apresentado ao lado do gráfico para identificá-lo. Como padrão é exibido o valor percentual.

Comentários

O código abaixo ilustra o uso do método **CellGraph**.

```
Table.BeginTable "75%", "Exemplo de grafico", 2, True  
Table.BeginRow 1, True  
Table.Cell "Nomes"
```

```
Table.GraphLegend dblMax, dblTotal  
Table.EndRow  
Do Until subjRS.EOF  
    Table.BeginRow 1  
    Table.Cell subjRS("Nome")  
    Table.CellGraph subjRS("Valor")  
    Table.EndRow  
    subjRS.MoveNext  
Loop  
Table.BeginRow 1, True  
Table.Cell "Total Geral"  
Table.Cell String.FormatMoney(dblTotal)  
Table.EndRow  
Table.EndTable
```

Aplica-se
Table

Veja também

Método EndCell

Descrição

Finaliza uma célula da tabela.

Sintaxe

objeto.**EndCell**

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Table</i> .

Comentários

O código abaixo ilustra o uso do método **EndCell**.

```
Table.Padding = 4
Table.Spacing = 0
Table.BeginTable "65%", ,2 ,True

Table.CellColSpan = 2
Table.BeginRow 3, True
Table.Cell "Painel de Controle"
Table.EndRow

Table.BeginRow 2
Table.BeginCell
URL.BeginURL Initializer.ScriptName

URL.Add efQueryStrAction, efQSActionEditor
URL.Add efQueryStrWhat, conWhatChamaTecnico
URL.Show "Chamado Técnico", efQueryStrEditableStr
Show.BR
URL.EndURL
Table.EndCell
Table.EndRow
Table.EndTable
```

Aplica-se
Table

Veja também

Método EndRow

Descrição

Finaliza uma linha da tabela.

Sintaxe

objeto.**EndRow**

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Table</i> .

Comentários

O código abaixo ilustra o uso do método **EndRow**.

```
Table.Padding = 4
Table.Spacing = 0
Table.BeginTable "65%", ,2 ,True

Table.CellColSpan = 2
Table.BeginRow 3, True
Table.Cell "Painel de Controle"
Table.EndRow

Table.BeginRow 2
Table.BeginCell
URL.BeginURL Initializer.ScriptName

URL.Add efQueryStrAction, efQSAActionEditor
URL.Add efQueryStrWhat, conWhatChamaTecnico
URL.Show "Chamado Técnico", efQueryStrEditableStr
Show.BR
URL.EndURL
Table.EndCell
Table.EndRow
Table.EndTable
```

Aplica-se

Table

Veja também

Método EndTable

Descrição

Finaliza a tabela.

Sintaxe

objeto.**EndTable**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Table</i> .

Comentários

O código abaixo ilustra o uso do método **EndTable**.

```
Table.Padding = 4
Table.Spacing = 0
Table.BeginTable "65%", ,2 ,True

Table.CellColSpan = 2
Table.BeginRow 3, True
Table.Cell "Painel de Controle"
Table.EndRow

Table.BeginRow 2
Table.BeginCell
URL.BeginURL Initializer.ScriptName

URL.Add efQueryStrAction, efQSActionEditor
URL.Add efQueryStrWhat, conWhatChamaTecnico
URL.Show "Chamado Técnico", efQueryStrEditableStr
Show.BR
URL.EndURL
Table.EndCell
Table.EndRow
Table.EndTable
```

Aplica-se

Table

Veja também

Método GraphLegend

Descrição

Define o valor das propriedades *GraphMax* e *GraphTotal* e exibe a legenda, indicando o valor médio e máximo dos gráficos de barra.

Sintaxe*objeto.GraphLegend Double1, Double2*

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Table</i> .
<i>Double1</i>	Requerido. Valor máximo que ocorre nos gráficos.
<i>Double2</i>	Requerido. Valor total, resultado da soma dos valores de todos os gráficos.

Comentários

O código abaixo ilustra o uso do método **GraphLegend**.

```
Table.BeginTable "75%", "Exemplo de grafico", 2, True
    Table.BeginRow 1, True
    Table.Cell "Nomes"

    Table.GraphLegend dblMax, dblTotal
    Table.EndRow
Do Until subjRS.EOF
    Table.BeginRow 1
    Table.Cell subjRS("Nome")
    Table.CellGraph subjRS("Valor")
    Table.EndRow
    subjRS.MoveNext
Loop
Table.BeginRow 1, True
Table.Cell "Total Geral"
Table.Cell String.FormatMoney(dblTotal)
Table.EndRow
Table.EndTable
```

Aplica-se

Table

Veja também

Objeto TableStyle

Objeto TableStyle

Objeto que implementa um conjunto de propriedades utilizadas na formatação de tabelas geradas com o objeto Table.

Sintaxe

```
Set TableStyle = Server.CreateObject("HiperTools30.TableStyle")
```

Propriedade AlternateColor

Descrição

XxXxXx.

Sintaxe

objeto.**AlternateColor** [= *String*]

O argumento objeto sempre será o objeto *TableStyle*.

Comentários

Os valores válidos para essa propriedade são os mesmos do HTML, especificado como nome da cor ou valor RGB em hexadecimal.

O código abaixo ilustra o uso da propriedade **AlternateColor**.

```
TableStyle.AlternateColor = "#FFFFFF"  
TableStyle.AlternateColor = "white"
```

Aplica-se

TableStyle

Veja também

Propriedade BackgroundFormat

Descrição

XxXxXx.

Sintaxe

objeto.**BackgroundFormat** [= *Integer*]

O argumento objeto sempre será o objeto *TableStyle*.

Comentários

O código abaixo ilustra o uso da propriedade **BackgroundFormat**.

Exemplo

Aplica-se

TableStyle

Veja também

Propriedade `BaseColor`

Descrição

XxXxXx.

Sintaxe

`objeto.BaseColor` [= *String*]

O argumento objeto sempre será o objeto *TableStyle*.

Comentários

Os valores válidos para essa propriedade são os mesmos do HTML, especificado como nome da cor ou valor RGB em hexadecimal.

O código abaixo ilustra o uso da propriedade **`BaseColor`**.

```
TableStyle.BaseColor = "#FFFFFF"
```

```
TableStyle.BaseColor = "white"
```

Aplica-se

`TableStyle`

Veja também

Propriedade `BorderColor`

Descrição

Especifica a cor das bordas externa, interna e do cabeçalho que será aplicada à tabela.

Sintaxe

`objeto.BorderColor` [= *String*]

O argumento objeto sempre será o objeto *TableStyle*.

Comentários

Os valores válidos para essa propriedade são os mesmos do HTML, especificado como nome da cor ou valor RGB em hexadecimal.

O código abaixo ilustra o uso da propriedade **`BorderColor`**.

```
TableStyle.BorderColor = "#FFFFFF"
```

```
TableStyle.BorderColor = "white"
```

Aplica-se

`TableStyle`

Veja também

Propriedade `BorderFormat`

Descrição

XxXxXx.

Sintaxe

objeto.**BorderFormat** [= *Integer*]

O argumento *objeto* sempre será o objeto *TableStyle*.

Comentários

O código abaixo ilustra o uso da propriedade **BorderFormat**.

Exemplo

Aplica-se

`TableStyle`

Veja também

Propriedade `BorderWidth`

Descrição

Especifica a espessura, em pixels, das bordas externa, interna e do cabeçalho que será aplicada à tabela.

Sintaxe

objeto.**BorderWidth** [= *Byte*]

O argumento *objeto* sempre será o objeto *TableStyle*.

Comentários

O código abaixo ilustra o uso da propriedade **BorderWidth**.

`TableStyle.BorderWidth = 1`

Aplica-se

`TableStyle`

Veja também

Propriedade `CellBackground`

Descrição

Especifica a imagem a ser usada como fundo das células da tabela.

Sintaxe

objeto.**CellBackground** [= *String*]

O argumento *objeto* sempre será o objeto *TableStyle*.

Comentários

O código abaixo ilustra o uso da propriedade **CellBackground**.

`TableStyle.CellBackground = "cimento.gif"`

Aplica-se
`TableStyle`

Veja também

Propriedade `ColorFormat`

Descrição

`XxXxXx`.

Sintaxe

`objeto.ColorFormat [= Integer]`

O argumento `objeto` sempre será o objeto `TableStyle`.

Comentários

O código abaixo ilustra o uso da propriedade `ColorFormat`.

Exemplo

Aplica-se

`TableStyle`

Veja também

Propriedade `ExternalBorder`

Descrição

Retorna um objeto `Border` representando as opções de apresentação da borda externa da tabela.

Sintaxe

`objeto.ExternalBorder`

O argumento `objeto` sempre será o objeto `TableStyle`.

Comentários

O código abaixo ilustra o uso da propriedade `ExternalBorder`.

```
TableStyle.ExternalBorder.Color = "gray"
```

```
TableStyle.ExternalBorder.Width = 2
```

Aplica-se

`TableStyle`

Veja também

Propriedade `FirstAltBackground`

Descrição

XxXxXx.

Sintaxe

`objeto.FirstAltBackground [= String]`

O argumento `objeto` sempre será o objeto `TableStyle`.

Comentários

O código abaixo ilustra o uso da propriedade **FirstAltBackground**.

Exemplo

Aplica-se

`TableStyle`

Veja também

Propriedade `FirstAltColor`

Descrição

XxXxXx.

Sintaxe

`objeto.FirstAltColor [= String]`

O argumento `objeto` sempre será o objeto `TableStyle`.

Comentários

Os valores válidos para essa propriedade são os mesmos do HTML, especificado como nome da cor ou valor RGB em hexadecimal.

O código abaixo ilustra o uso da propriedade **FirstAltColor**.

```
TableStyle.FirstAltColor = "#FFFFFF"
```

```
TableStyle.FirstAltColor = "white"
```

Aplica-se

`TableStyle`

Veja também

Propriedade `FontColor`

Descrição

Define a cor da fonte interna e do cabeçalho da tabela.

Sintaxe

`objeto.FontColor [= String]`

O argumento `objeto` sempre será o objeto `TableStyle`.

Comentários

Os valores válidos para essa propriedade são os mesmos do HTML, especificado como nome da cor ou valor RGB em hexadecimal.

O código abaixo ilustra o uso da propriedade **FontColor**.

```
TableStyle.FontColor = "#FFFFFF"  
TableStyle.FontColor = "white"
```

Aplica-se

`TableStyle`

Veja também

Propriedade `FontSize`

Descrição

Define o tamanho da fonte interna e do cabeçalho da tabela.

Sintaxe

objeto.**FontSize** [= *Integer*]

O argumento *objeto* sempre será o objeto *TableStyle*.

Comentários

Os valores válidos para essa propriedade são os mesmos do HTML.

O código abaixo ilustra o uso da propriedade **FontSize**.

```
TableStyle.FontSize = -1
```

Aplica-se

`TableStyle`

Veja também

Propriedade `FontBold`

Descrição

Define se a fonte interna e do cabeçalho da tabela será exibida como negrito.

Sintaxe

objeto.**FontBold** [= *Boolean*]

O argumento *objeto* sempre será o objeto *TableStyle*.

Comentários

O código abaixo ilustra o uso da propriedade **FontBold**.

```
TableStyle.FontBold = True
```

Aplica-se
TableStyle

Veja também

Propriedade FontItalic

Descrição

Define se a fonte interna e do cabeçalho da tabela será exibida como itálico.

Sintaxe

objeto.**FontItalic** [= *Boolean*]

O argumento objeto sempre será o objeto *TableStyle*.

Comentários

O código abaixo ilustra o uso da propriedade **FontItalic**.

```
TableStyle.FontItalic = True
```

Aplica-se
TableStyle

Veja também

Propriedade FontUnderline

Descrição

Define se a fonte interna e do cabeçalho da tabela será exibida como sublinhado.

Sintaxe

objeto.**FontUnderline** [= *Boolean*]

O argumento objeto sempre será o objeto *TableStyle*.

Comentários

O código abaixo ilustra o uso da propriedade **FontUnderline**.

```
TableStyle.FontUnderline = True
```

Aplica-se
TableStyle

Veja também

Propriedade HeaderBorder

Descrição

Retorna um objeto *Border* representando as opções de apresentação da borda apresentada no cabeçalho da tabela.

Sintaxe

`objeto.HeaderBorder`

O argumento *objeto* sempre será o objeto *TableStyle*.

Comentários

O código abaixo ilustra o uso da propriedade **HeaderBorder**.

```
TableStyle.HeaderBorder.Color = "gray"
TableStyle.HeaderBorder.Width = 2
```

Aplica-se

`TableStyle`

Veja também

Propriedade `HeaderCellBackground`

Descrição

Especifica a imagem a ser usada como fundo das células do cabeçalho da tabela.

Sintaxe

`objeto.HeaderCellBackground [= String]`

O argumento *objeto* sempre será o objeto *TableStyle*.

Comentários

O código abaixo ilustra o uso da propriedade **HeaderCellBackground**.

```
TableStyle.HeaderCellBackground = "cimento.gif"
```

Aplica-se

`TableStyle`

Veja também

Propriedade `HeaderColor`

Descrição

Especifica a cor de fundo das células de cabeçalho da tabela.

Sintaxe

`objeto.HeaderColor [= String]`

O argumento *objeto* sempre será o objeto *TableStyle*.

Comentários

Os valores válidos para essa propriedade são os mesmos do HTML, especificado como nome da cor ou valor RGB em hexadecimal.

O código abaixo ilustra o uso da propriedade **HeaderColor**.

```
TableStyle.HeaderColor = "blue"
```

```
TableStyle.HeaderColor = "#0000FF"
```

Aplica-se

TableStyle

Veja também

HeaderBorder, HeaderFont

Propriedade HeaderFont

Descrição

Retorna um objeto *Font* representando a formatação da fonte do cabeçalho da tabela.

Sintaxe

objeto.HeaderFont

O argumento *objeto* sempre será o objeto *TableStyle*.

Comentários

O código abaixo ilustra o uso da propriedade **HeaderFont**.

```
TableStyle.HeaderFont.Color = "#0089FD"  
TableStyle.HeaderFont.Size = 1
```

Aplica-se

TableStyle

Veja também

InternalFont, TitleFont

Propriedade HeaderRowBackground

Descrição

Especifica a imagem a ser usada como fundo das linhas do cabeçalho da tabela.

Sintaxe

objeto.HeaderRowBackground [= *String*]

O argumento *objeto* sempre será o objeto *TableStyle*.

Comentários

O código abaixo ilustra o uso da propriedade **HeaderRowBackground**.

```
TableStyle.HeaderRowBackground = "cimento.gif"
```

Aplica-se

TableStyle

Veja também

Propriedade InternalBorder

Descrição

Retorna um objeto *Border* representando as opções de apresentação da borda interna da tabela.

Sintaxe

objeto.**InternalBorder**

O argumento *objeto* sempre será o objeto *TableStyle*.

Comentários

O código abaixo ilustra o uso da propriedade **InternalBorder**.

```
TableStyle.InternalBorder.Color = "gray"  
TableStyle.InternalBorder.Width = 2
```

Aplica-se

TableStyle

Veja também

Propriedade `InternalFont`

Descrição

Retorna um objeto *Font* representando a formatação da fonte interna da tabela.

Sintaxe

objeto.**InternalFont**

O argumento *objeto* sempre será o objeto *TableStyle*.

Comentários

O código abaixo ilustra o uso da propriedade **InternalFont**.

```
TableStyle.InternalFont.Color = "#0089FD"  
TableStyle.InternalFont.Size = 1
```

Aplica-se

TableStyle

Veja também

HeaderFont, *TitleFont*

Propriedade `LastColor`

Descrição

XxXxXx.

Sintaxe

objeto.**LastColor** [= *String*]

O argumento *objeto* sempre será o objeto *TableStyle*.

Comentários

Os valores válidos para essa propriedade são os mesmos do HTML, especificado como nome da cor ou valor RGB em hexadecimal.

O código abaixo ilustra o uso da propriedade **LastColor**.

```
TableStyle.LastColor = "#FFFFFF"  
TableStyle.LastColor = "white"
```

Aplica-se

TableStyle

Veja também

Propriedade NeedTwoTables

Descrição

Retorna se será necessário colocar uma tabela dentro de outra, para exibição das bordas. O HiperTools internamente cria bordas com o uso de duas tabelas. Essa propriedade é read-only.

Sintaxe

[*Boolean* =] *objeto*.**NeedTwoTables**

O argumento *objeto* sempre será o objeto *TableStyle*.

Comentários

O código abaixo ilustra o uso da propriedade **TitleColor**.

```
TableStyle.BorderFormat = 2  
  
If TableStyle.NeedTwoTables Then  
    Show.HTML "Serão utilizadas duas tabelas."  
End If
```

Aplica-se

TableStyle

Veja também

Propriedade RowBackground

Descrição

Especifica a imagem a ser usada como fundo das linhas da tabela.

Sintaxe

objeto.**RowBackground** [= *String*]

O argumento *objeto* sempre será o objeto *TableStyle*.

Comentários

O código abaixo ilustra o uso da propriedade **RowBackground**.

```
TableStyle.RowBackground = "cimento.gif"
```

Aplica-se`TableStyle`**Veja também**

Propriedade `SecondAltBackground`

Descrição`XxXxXx.`**Sintaxe**`objeto.SecondAltBackground [= String]`

O argumento *objeto* sempre será o objeto *TableStyle*.

Comentários

O código abaixo ilustra o uso da propriedade **`SecondAltBackground`**.

Exemplo

Aplica-se`TableStyle`**Veja também**

Propriedade `TableBackground`

Descrição

Especifica a imagem a ser usada como fundo da tabela.

Sintaxe`objeto.TableBackground [= String]`

O argumento *objeto* sempre será o objeto *TableStyle*.

Comentários

O código abaixo ilustra o uso da propriedade **`TableBackground`**.

```
TableStyle.TableBackground = "cimento.gif"
```

Aplica-se`TableStyle`**Veja também**

Propriedade `TitleColor`

Descrição

Define a cor do título da tabela.

Sintaxe

objeto.**TitleColor** [= *String*]

O argumento *objeto* sempre será o objeto *TableStyle*.

Comentários

Os valores válidos para essa propriedade são os mesmos do HTML, especificado como nome da cor ou valor RGB em hexadecimal.

O código abaixo ilustra o uso da propriedade **TitleColor**.

```
TableStyle.TitleColor = "#FFFFFF"
```

```
TableStyle.TitleColor = "white"
```

Aplica-se

`TableStyle`

Veja também

Propriedade `TitleFont`

Descrição

Retorna um objeto *Font* representando a formatação da fonte do título da tabela.

Sintaxe

objeto.**TitleFont**

O argumento *objeto* sempre será o objeto *TableStyle*.

Comentários

O código abaixo ilustra o uso da propriedade **TitleFont**.

```
TableStyle.TitleFont.Color = "#0089FD"
```

```
TableStyle.TitleFont.Size = 1
```

Aplica-se

`TableStyle`

Veja também

`InternalFont`, `TitleFont`

Objeto Tip

Objeto Tip

Este objeto cria seqüências de itens que podem exibir um texto ou outros elementos, previamente ocultos, ao serem clicados. Esse recurso usa DHTML e é plenamente suportado apenas em Internet Explorer 4.0 ou superior.

Sintaxe

```
Set Tip = Server.CreateObject("HiperTools30.Tip")
```

Propriedade ContentClass

Descrição

Define a classe CSS dos conteúdos dos tips.

Sintaxe

```
objeto.ContentClass [= String]
```

O argumento *objeto* sempre será o objeto *Tip*.

Comentários

O código abaixo ilustra o uso do método **ContentClass**.

```
Tip.ContentClass = "texto"
```

Aplica-se

Tip

Veja também

Propriedade OutColor

Descrição

Define a cor títulos dos tips enquanto o cursor do mouse não estiver sobre eles.

Sintaxe

```
objeto.OutColor [= String]
```

O argumento *objeto* sempre será o objeto *Tip*.

Comentários

O código abaixo ilustra o uso do método **OutColor**.

```
Tip.OutColor = "yellow"
```

Aplica-se

Tip

Veja também

Propriedade OverColor

Descrição

Define a cor títulos dos tips enquanto o cursor do mouse passar sobre eles.

Sintaxe

objeto.**OverColor** [= *String*]

O argumento *objeto* sempre será o objeto *Tip*.

Comentários

O código abaixo ilustra o uso do método **OverColor**.

```
Tip.OverColor = "yellow"
```

Aplica-se

Tip

Veja também

Propriedade TitleClass

Descrição

Define a classe CSS dos títulos dos tips.

Sintaxe

objeto.**TitleClass** [= *String*]

O argumento *objeto* sempre será o objeto *Tip*.

Comentários

O código abaixo ilustra o uso do método **TitleClass**.

```
Tip.TitleClass = "titulo"
```

Aplica-se

Tip

Veja também

Método AddTip

Descrição

Exibe um novo tip.

Sintaxe

objeto.**AddTip** *String1*, *String2*

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Tip</i> .
<i>String1</i>	Requerido. Texto que aparecerá no link do tip.
<i>String2</i>	Requerido. Conteúdo que será mostrado ao clique no link do tip.

Comentários

O código abaixo ilustra o uso do método **AddTip**.

```
Tip.AddTip "Título", "Este conteúdo aparecerá ao clique."
```

Aplica-se

Tip

Veja também

Método AddTipBegin

Descrição

Exibe um novo tip. Este método é semelhante ao AddTip, com a vantagem de permitir tips em vários níveis, um dentro do outro.

Sintaxe

objeto.**AddTipBegin** *String1*, *String2*

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Tip</i> .
<i>String1</i>	Requerido. Texto que aparecerá no link do tip.
<i>String2</i>	Requerido. Conteúdo que será mostrado ao clique no link do tip.

Comentários

O código abaixo ilustra o uso do método **AddTipBegin**.

```
Tip.AddTipBegin "Título", "Este conteúdo aparecerá ao clique."
```

```
Tip.AddTip "Interno", "Este tip está dentro do tip anterior."
```

`Tip.AddTipEnd`**Aplica-se**

Tip

Veja também

Método AddTipEnd

Descrição

Finaliza um tip iniciado pelo método *AddTipBegin*, retornando ao nível imediatamente superior.

Sintaxe*objeto*.AddTipEnd

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Tip</i> .

Comentários

O código abaixo ilustra o uso do método **AddTipEnd**.

```
Tip.AddTipBegin "Título", "Este conteúdo aparecerá ao clique."  
Tip.AddTip "Interno", "Este tip está dentro do tip anterior."  
Tip.AddTipEnd
```

Aplica-se

Tip

Veja também

Método BeginTip

Descrição

Inclui no documento o código necessário para a exibição dos tips. As propriedades do objeto *Tip* devem ser definidas antes da chamada a esse método.

Sintaxe*objeto*.BeginTip

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Tip</i> .

Comentários

O código abaixo ilustra o uso do método **BeginTip**.

```
Tip.BeginTip
```

Aplica-se

Tip

Veja também

Método EndTip

Descrição

Finaliza a exibição dos tips no documento atual.

Sintaxe

```
objeto.EndTip
```

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>Tip</i> .

Comentários

O código abaixo ilustra o uso do método **EndTip**.

```
Tip.EndTip
```

Aplica-se

Tip

Veja também

Objeto TreeView

Objeto TreeView

Objeto é utilizado para apresentar uma lista hierarquica de itens, como um indice de um livro ou a estrutura de diretórios de um disco.

Sintaxe

```
Set TreeView = Server.CreateObject("HiperTools30.TreeView")
```

Comentários

O código abaixo ilustra o uso do objeto **TreeView**.

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#INCLUDE VIRTUAL="/hipertools30site/Objects.inc"-->
<!--#INCLUDE VIRTUAL="/hipertools30site/Constants.inc"-->
<%
REM =====
REM /TreeView.asp
REM -----
REM Nome      : Sistema de teste do TreeView
REM Autor     : José da Silva
REM Versao    : 1.0.0
REM Local     : Brasília - DF
REM Companhia: XYZ Software
REM -----

Main

REM =====
REM Rotina Principal do Sistema
REM -----
Private Sub Main

    Server.ScriptTimeout = conScriptTimeout
    Session.Timeout      = conSessionTimeout

MainBody

End Sub
REM -----
REM Fim Main
REM =====

REM =====
REM Monta o Menu em Tree-View
REM -----
Sub ShowTreeViewMenu

    Session("htDefaultHiperToolsWorkPath") = "/ZTITools"

    strSA = "ScriptAction.asp"
    TreeView.DefaultImage = "Folder"
    TreeView.TreeStyleLink = "font-style: normal; text-decoration: none;" _
        " color:blue"
    TreeView.TreeStyleActive = "font-style: normal; text-decoration: none;" _
        " color:red"
    TreeView.TreeStyleVisited = "font-style: normal; text-decoration: none;" _
```



```

TreeView.TreeStyleHover      = " color:blue"
TreeView.TreeStyleHover      = "font-style: normal; text-decoration: " _
                              " underline;color:red"
TreeView.ItemTarget          = "Body"
TreeView.ItemMaxSize         = 15
TreeView.ItemNoWrap          = True

TreeView.AddImage "Folder", "/HiperTools/Img/Folder/"
TreeView.AddImage "User", "/HiperTools/Img/Folder/", "aberto", "fechado"

TreeView.BeginTree
    TreeView.AddItem "1º item", "ScriptAction.asp?Action=1", "Ação 1", "User"

    TreeView.BeginNode "1º nó", "ScriptAction.asp?Action=2", , , True
        TreeView.AddItem "1º item do 1º nó", "ScriptAction.asp?Action=3"
        TreeView.AddItem "2º item do 1º nó", "ScriptAction.asp?Action=4"
        TreeView.AddSubCaption "1º título auxiliar do 2º item do 1º nó", _
                                "ScriptAction.asp?Action=5", , _
                                TreeView.ItemTarget
    TreeView.EndNode

    TreeView.BeginNode "2º nó", "ScriptAction.asp?Action=6", _
                        "Este é o valor da propriedade 'Title'"
        TreeView.AddItem "1º item do 2º nó"
    TreeView.EndNode

    TreeView.AddItem "2º item", "ScriptAction.asp?Action=6"
TreeView.EndTree

End Sub
REM -----
REM Fim ShowTreeViewMenu
REM =====

REM =====
REM Corpo Principal do sistema
REM -----
Private Sub MainBody

    Default.HTMLBegin
    Show.HTML "<TITLE>Menu do ZEUS III v1.0</TITLE>"
    Show.HTML "<BODY>"
    Show.HTML "<BASEFONT FACE=""Arial, Helvetica, Sans-Serif"">"

    'Chama o Menu no Tree View
    ShowTreeViewMenu

    Show.HTML "</BODY>"
    Default.HTMLEnd

End Sub
REM -----
REM Fim MainBody
REM =====

REM -----
REM Fim do TreeView.asp

```

```
REM =====  
%>
```

Propriedade DefaultImage

Descrição

Especifica qual é a imagem padrão a ser utilizada quando o usuário não informar o nome da imagem ao adicionar um item ou um nó.

Sintaxe

objeto.DefaultImage [= String]

O argumento *objeto* sempre será o objeto TreeView. O valor para **DefaultImage** será.

Comentários

O código abaixo ilustra o uso da propriedade **DefaultImage**.

```
TreeView.DefaultImage = "Folder"  
TreeView.AddImage "Folder", "/HiperTools/Img/Folder/"  
TreeView.AddImage "User", "/HiperTools/Img/Folder/", "aberto", "fechado"  
TreeView.BeginTree  
    TreeView.AddItem "1º item", "ScriptAction.asp?Action=1", "Ação 1", "User"  
    TreeView.AddItem "2º item", "ScriptAction.asp?Action=1", "Com Imagem padrão"  
TreeView.EndTree
```

Aplica-se

TreeView.

Veja também

Propriedade ItemBGColor

Descrição

Especifica qual é a cor de fundo para os próximos itens adicionados.

Sintaxe

objeto.ItemBGColor [= String]

O argumento *objeto* sempre será o objeto TreeView. O valor para **ItemBGColor** deve ser um valor aceitável no padrão HTML.

Comentários

O código abaixo ilustra o uso da propriedade **ItemBGColor**.

```
TreeView.ItemBGColor = "red"  
TreeView.TreeBGColor = "blue"  
TreeView.BeginTree  
    TreeView.AddItem "1º item", "ScriptAction.asp?Action=1", "Ação 1", "User"  
    TreeView.AddItem "2º item", "ScriptAction.asp?Action=1", "Title"  
TreeView.EndTree
```

Aplica-se

TreeView.

Veja também

Propriedade ItemFontColor

Descrição

Especifica qual é a cor da fonte para os próximos itens adicionados.

Sintaxe

objeto.ItemFontColor [= String]

O argumento *objeto* sempre será o objeto TreeView. O valor para **ItemFontColor** deve ser um valor aceitável no padrão HTML.

Comentários

O código abaixo ilustra o uso da propriedade **ItemFontColor**.

```
TreeView.ItemFontColor = "blue"
TreeView.ItemFontFace = "Arial"
TreeView.ItemFontSize = "15"
TreeView.ItemBold = True
TreeView.ItemItalic = True
TreeView.ItemUnderline = False
TreeView.BeginTree
    TreeView.AddItem "1º item", "ScriptAction.asp?Action=1", "Ação 1", "User"
    TreeView.AddItem "2º item", "ScriptAction.asp?Action=1", "Title"
TreeView.EndTree
```

Aplica-se

TreeView.

Veja também

Propriedade ItemFontFace

Descrição

Especifica qual é o nome da fonte para os próximos itens adicionados.

Sintaxe

objeto.ItemFontFace [= String]

O argumento *objeto* sempre será o objeto TreeView. O valor para **ItemFontFace** deve ser um nome de fonte disponível para utilização no texto do item.

Comentários

O código abaixo ilustra o uso da propriedade **ItemFontFace**.

```
TreeView.ItemFontColor = "blue"
TreeView.ItemFontFace = "Arial"
TreeView.ItemFontSize = "15"
TreeView.ItemBold = True
TreeView.ItemItalic = True
TreeView.ItemUnderline = False
TreeView.BeginTree
    TreeView.AddItem "1º item", "ScriptAction.asp?Action=1", "Ação 1", "User"
    TreeView.AddItem "2º item", "ScriptAction.asp?Action=1", "Title"
TreeView.EndTree
```

Aplica-se

TreeView.

Veja também

Propriedade ItemFontSize

Descrição

Especifica o tamanho da fonte que será utilizada no texto do item.

Sintaxe

objeto.**ItemFontSize** [= *String*]

O argumento *objeto* sempre será o objeto TreeView. O valor para **ItemFontSize** indica o tamanho da fonte que será utilizada no texto do item.

Comentários

O código abaixo ilustra o uso da propriedade **ItemFontSize**.

```
TreeView.ItemFontColor = "blue"
TreeView.ItemFontFace = "Arial"
TreeView.ItemFontSize = "15"
TreeView.ItemBold = True
TreeView.ItemItalic = True
TreeView.ItemUnderline = False
TreeView.BeginTree
    TreeView.AddItem "1º item", "ScriptAction.asp?Action=1", "Ação 1", "User"
    TreeView.AddItem "2º item", "ScriptAction.asp?Action=1", "Title"
TreeView.EndTree
```

Aplica-se

TreeView.

Veja também

Propriedade ItemBold

Descrição

Especifica se os textos dos próximos itens adicionados aparecerão com negrito.

Sintaxe

objeto.**ItemBold** [= *Boolean*]

O argumento *objeto* sempre será o objeto TreeView. O valor para **ItemBold** indica se os textos dos próximos itens adicionados aparecerão com negrito.

Comentários

O código abaixo ilustra o uso da propriedade **ItemBold**.

```
TreeView.ItemFontColor = "blue"
TreeView.ItemFontFace = "Arial"
TreeView.ItemFontSize = "15"
TreeView.ItemBold = True
TreeView.ItemItalic = True
TreeView.ItemUnderline = False
TreeView.BeginTree
    TreeView.AddItem "1º item", "ScriptAction.asp?Action=1", "Ação 1", "User"
    TreeView.AddItem "2º item", "ScriptAction.asp?Action=1", "Title"
TreeView.EndTree
```

Aplica-se

TreeView.

Veja também

Propriedade ItemItalic

Descrição

Especifica se os textos dos próximos itens adicionados aparecerão com itálico.

Sintaxe

objeto.ItemItalic [= Boolean]

O argumento *objeto* sempre será o objeto TreeView. O valor para **ItemItalic** indica se os textos dos próximos itens adicionados aparecerão com itálico.

Comentários

O código abaixo ilustra o uso da propriedade **ItemItalic**.

```
TreeView.ItemFontColor = "blue"
TreeView.ItemFontFace = "Arial"
TreeView.ItemFontSize = "15"
TreeView.ItemBold = True
TreeView.ItemItalic = True
TreeView.ItemUnderline = False
TreeView.BeginTree
    TreeView.AddItem "1º item", "ScriptAction.asp?Action=1", "Ação 1", "User"
    TreeView.AddItem "2º item", "ScriptAction.asp?Action=1", "Title"
TreeView.EndTree
```

Aplica-se

TreeView.

Veja também

Propriedade ItemUnderline

Descrição

Especifica se os textos dos próximos itens adicionados aparecerão com subscrito.

Sintaxe

objeto.ItemUnderline [= Boolean]

O argumento *objeto* sempre será o objeto TreeView. O valor para **ItemUnderline** indica se os textos dos próximos itens adicionados aparecerão com subscrito.

Comentários

O código abaixo ilustra o uso da propriedade **ItemUnderline**.

```
TreeView.ItemFontColor = "blue"
TreeView.ItemFontFace = "Arial"
TreeView.ItemFontSize = "15"
TreeView.ItemBold = True
TreeView.ItemItalic = True
TreeView.ItemUnderline = False
TreeView.BeginTree
```

Objeto TreeView

```
TreeView.AddItem "1º item", "ScriptAction.asp?Action=1", "Ação 1", "User"
TreeView.AddItem "2º item", "ScriptAction.asp?Action=1", "Title"
TreeView.EndTree
```

Aplica-se

TreeView.

Veja também

Propriedade ItemNoWrap

Descrição

Especifica se o texto do item deverá ou não aparecer quebrado em múltiplas linhas.

Sintaxe

objeto.ItemNoWrap [= Boolean]

O argumento *objeto* sempre será o objeto TreeView. O valor para **ItemNoWrap** indica se o texto será quebrado e ou não.

Comentários

O código abaixo ilustra o uso da propriedade **ItemNoWrap**.

```
TreeView.ItemNoWrap = True
TreeView.BeginTree
TreeView.AddItem "Um texto bem longo que deverá ser " _
                "mostrado na mesma linha sem quebrar", _
                "ScriptAction.asp?Action=1", "Ação 1", "User"
TreeView.AddItem "2º item", "ScriptAction.asp?Action=1", "Title"
TreeView.EndTree
```

Aplica-se

TreeView.

Veja também

Propriedade ItemHAlign

Descrição

Especifica qual é o alinhamento horizontal para os próximos itens adicionados.

Sintaxe

objeto.ItemHAlign [= String]

O argumento *objeto* sempre será o objeto TreeView. O valor para **ItemHAlign** indica o tipo de alinhamento utilizado no texto do item adicionado.

Comentários

O código abaixo ilustra o uso da propriedade **ItemHAlign**.

```
TreeView.ItemHAlign = "Right"
TreeView.ItemWidth = 150
TreeView.ItemVAlign = "Bottom"
TreeView.ItemHeight = 50
TreeView.BeginTree
TreeView.AddItem "1º item", "ScriptAction.asp?Action=1", "Ação 1", "User"
TreeView.AddItem "2º item", "ScriptAction.asp?Action=1", "Title"
```

```
TreeView.EndTree
```

Aplica-se

TreeView.

Veja também

Propriedade ItemVAlign

Descrição

Especifica qual é o alinhamento vertical para os próximos itens adicionados.

Sintaxe

objeto.ItemVAlign [= String]

O argumento *objeto* sempre será o objeto TreeView. O valor para **ItemVAlign** indica o tipo de alinhamento utilizado na vertical no texto dos próximos itens adicionados.

Comentários

O código abaixo ilustra o uso da propriedade **ItemVAlign**.

```
TreeView.ItemHAlign = "Right"
TreeView.ItemWidth = 150
TreeView.ItemVAlign = "Bottom"
TreeView.ItemHeight = 50
TreeView.BeginTree
    TreeView.AddItem "1º item", "ScriptAction.asp?Action=1", "Ação 1", "User"
    TreeView.AddItem "2º item", "ScriptAction.asp?Action=1", "Title"
TreeView.EndTree
```

Aplica-se

TreeView

Veja também

Propriedade ItemTarget

Descrição

Especifica qual é o alvo (target) para os próximos itens adicionados.

Sintaxe

objeto.ItemTarget [= String]

O argumento *objeto* sempre será o objeto TreeView. O valor para **ItemTarget** indica o alvo para os próximos itens adicionados.

Comentários

O código abaixo ilustra o uso da propriedade **ItemTarget**.

```
TreeView.BeginTree
    TreeView.ItemTarget = "FrameBotom"
    TreeView.AddItem "1º item no (Bottom)", "ScriptAction.asp?Action=1"
    TreeView.ItemTarget = "FrameTop"
    TreeView.AddItem "2º item no (Top)", "ScriptAction.asp?Action=1"
TreeView.EndTree
```

Aplica-se
TreeView.

Veja também

Propriedade ItemWidth

Descrição

Especifica qual é a largura para os próximos itens adicionados.

Sintaxe

objeto.ItemWidth [= String]

O argumento *objeto* sempre será o objeto TreeView. O valor para **ItemWidth** indica qual é a largura.

Comentários

O código abaixo ilustra o uso da propriedade **ItemWidth**.

```
TreeView.ItemHAlign = "Right"
TreeView.ItemWidth = 150
TreeView.ItemVAlign = "Bottom"
TreeView.ItemHeight = 50
TreeView.BeginTree
    TreeView.AddItem "1º item", "ScriptAction.asp?Action=1", "Ação 1", "User"
    TreeView.AddItem "2º item", "ScriptAction.asp?Action=1", "Title"
TreeView.EndTree
```

Aplica-se
TreeView.

Veja também

Propriedade ItemHeight

Descrição

Especifica qual é a altura para os próximos itens adicionados.

Sintaxe

objeto.ItemHeight [= Tipo]

O argumento *objeto* sempre será o objeto TreeView. O valor para **ItemHeight** indica qual é a altura.

Comentários

O código abaixo ilustra o uso da propriedade **ItemHeight**.

```
TreeView.ItemHAlign = "Right"
TreeView.ItemWidth = 150
TreeView.ItemVAlign = "Bottom"
TreeView.ItemHeight = 50
TreeView.BeginTree
    TreeView.AddItem "1º item", "ScriptAction.asp?Action=1", "Ação 1", "User"
    TreeView.AddItem "2º item", "ScriptAction.asp?Action=1", "Title"
TreeView.EndTree
```


Aplica-se

TreeView.

Veja também

Propriedade TreeBGColor

Descrição

Especifica qual é a cor de fundo a ser utilizada na árvore.

Sintaxe

objeto.TreeBGColor [= String]

O argumento *objeto* sempre será o objeto TreeView. O valor para **TreeBGColor** é a cor de fundo da árvore.

Comentários

O código abaixo ilustra o uso da propriedade **TreeBGColor**.

```
TreeView.ItemBGColor = "red"
TreeView.TreeBGColor = "blue"
TreeView.BeginTree
    TreeView.AddItem "1º item", "ScriptAction.asp?Action=1", "Ação 1", "User"
    TreeView.AddItem "2º item", "ScriptAction.asp?Action=1", "Title"
TreeView.EndTree
```

Aplica-se

TreeView.

Veja também

Propriedade ItemMaxSize

Descrição

Especifica qual é o tamanho máximo de apresentação dos textos em número de caracteres para os próximos itens adicionados.

Sintaxe

objeto.ItemMaxSize [= Integer]

O argumento *objeto* sempre será o objeto TreeView. O valor para **ItemMaxSize** indica o número de caracteres.

Comentários

O código abaixo ilustra o uso da propriedade **ItemMaxSize**.

```
TreeView.ItemNoWrap = False
TreeView.ItemMaxSize = 15
TreeView.BeginTree
    TreeView.AddItem "Só vai aparecer os 12 primeiros caracteres + '...' ", _
        "ScriptAction.asp?Action=1", "Ação 1", "User"
    TreeView.AddItem "2º item", "ScriptAction.asp?Action=1", "Title"
TreeView.EndTree
```

Aplica-se
TreeView.

Veja também

Propriedade ItemStyle

Descrição

Especifica qual é o estilo (CSS) que deverá ser utilizado para os próximos itens adicionados.

Sintaxe

objeto.ItemStyle [= String]

O argumento *objeto* sempre será o objeto TreeView. O valor para **ItemStyle** indica o estilo a ser utilizado.

Comentários

O código abaixo ilustra o uso da propriedade **ItemStyle**.

```
TreeView.ItemStyle = "color:red"
TreeView.BeginTree
    TreeView.AddItem "1º item", "ScriptAction.asp?Action=1"
    TreeView.AddItem "2º item", "ScriptAction.asp?Action=2"
TreeView.EndTree
```

Aplica-se
TreeView.

Veja também

Propriedade TreeStyleLink

Descrição

Especifica qual é o estilo (CSS) do "Link" que deverá ser utilizado para todos os itens adicionados.

Sintaxe

objeto.TreeStyleLink [= String]

O argumento *objeto* sempre será o objeto TreeView. O valor para **TreeStyleLink** indica o estilo a ser utilizado.

Comentários

O código abaixo ilustra o uso da propriedade **TreeStyleLink**.

```
TreeView.TreeStyleLink = "font-style: normal; text-decoration: none;" _
    " color:blue"
TreeView.TreeStyleActive = "font-style: normal; text-decoration: none;" _
    " color:red"
TreeView.TreeStyleVisited = "font-style: normal; text-decoration: none;" _
    " color:blue"
TreeView.TreeStyleHover = "font-style: normal; text-decoration: " _
    " underline;color:red"
TreeView.BeginTree
    TreeView.AddItem "1º item", "ScriptAction.asp?Action=1"
    TreeView.AddItem "2º item", "ScriptAction.asp?Action=2"
TreeView.EndTree
```

Aplica-se

TreeView.

Veja também

Propriedade TreeStyleActive

Descrição

Especifica qual é o estilo (CSS) do "Active" que deverá ser utilizado para todos os itens adicionados.

Sintaxe

objeto. **TreeStyleActive** [= *String*]

O argumento *objeto* sempre será o objeto TreeView. O valor para **TreeStyleActive** indica o estilo a ser utilizado.

Comentários

O código abaixo ilustra o uso da propriedade **TreeStyleActive**.

```
TreeView.TreeStyleLink      = "font-style: normal; text-decoration: none;" _  
                             " color:blue"  
TreeView.TreeStyleActive   = "font-style: normal; text-decoration: none;" _  
                             " color:red"  
TreeView.TreeStyleVisited  = "font-style: normal; text-decoration: none;" _  
                             " color:blue"  
TreeView.TreeStyleHover    = "font-style: normal; text-decoration: " _  
                             " underline;color:red"  
TreeView.BeginTree  
    TreeView.AddItem "1º item", "ScriptAction.asp?Action=1"  
    TreeView.AddItem "2º item", "ScriptAction.asp?Action=2"  
TreeView.EndTree
```

Aplica-se

TreeView.

Veja também

Propriedade TreeStyleVisited

Descrição

Especifica qual é o estilo (CSS) do "Visited" que deverá ser utilizado para todos os itens adicionados.

Sintaxe

objeto. **TreeStyleVisited** [= *String*]

O argumento *objeto* sempre será o objeto TreeView. O valor para **TreeStyleVisited** indica o estilo a ser utilizado.

Comentários

O código abaixo ilustra o uso da propriedade **TreeStyleVisited**.

```
TreeView.TreeStyleLink      = "font-style: normal; text-decoration: none;" _
```

```
TreeView.TreeStyleActive = "font-style: normal; text-decoration: none;" _
    " color:blue"
TreeView.TreeStyleVisited = "font-style: normal; text-decoration: none;" _
    " color:red"
TreeView.TreeStyleHover = "font-style: normal; text-decoration: " _
    " underline;color:red"

TreeView.BeginTree
    TreeView.AddItem "1º item", "ScriptAction.asp?Action=1"
    TreeView.AddItem "2º item", "ScriptAction.asp?Action=2"
TreeView.EndTree
```

Aplica-se

TreeView.

Veja também

Propriedade TreeStyleHover

Descrição

Especifica qual é o estilo (CSS) do "Hover" que deverá ser utilizado para todos os itens adicionados.

Sintaxe

objeto.TreeStyleHover [= String]

O argumento *objeto* sempre será o objeto TreeView. O valor para **TreeStyleHover** indica o estilo a ser utilizado.

Comentários

O código abaixo ilustra o uso da propriedade **TreeStyleHover**.

```
TreeView.TreeStyleLink = "font-style: normal; text-decoration: none;" _
    " color:blue"
TreeView.TreeStyleActive = "font-style: normal; text-decoration: none;" _
    " color:red"
TreeView.TreeStyleVisited = "font-style: normal; text-decoration: none;" _
    " color:blue"
TreeView.TreeStyleHover = "font-style: normal; text-decoration: " _
    " underline;color:red"

TreeView.BeginTree
    TreeView.AddItem "1º item", "ScriptAction.asp?Action=1"
    TreeView.AddItem "2º item", "ScriptAction.asp?Action=2"
TreeView.EndTree
```

Aplica-se

TreeView.

Veja também

Propriedade CurrentChild

Descrição

Retorna a posição atual dos níveis de acordo com o nível de indentação em que estou na árvore.

Sintaxe

[*Result* =] *objeto*.**CurrentChild**

O argumento *objeto* sempre será o objeto TreeView. O valor para *Result* será **CurrentChild**, o valor de nó atual.

Comentários

O código abaixo ilustra o uso da propriedade **CurrentChild**.

```
TreeView.BeginTree
    TreeView.BeginNode "Identação" & TreeView.CurrentField, _
        "ScriptAction.asp?Action=0"
Retorna: 0
    TreeView.BeginNode "Identação" & TreeView.CurrentField, _
        "ScriptAction.asp?Action=1"
Retorna: 1
    TreeView.BeginNode "Identação" & TreeView.CurrentField, _
        "ScriptAction.asp?Action=2"
Retorna: 2
    TreeView.EndNode
    TreeView.EndNode
    TreeView.EndNode
TreeView.EndTree
```

Aplica-se

TreeView.

Veja também

Propriedade SetLevelID

Descrição

Define um valor de identificação deste para o próximo item apenas. É utilizada para posterior identificação por parte do usuário do item clicado.

Sintaxe

objeto.**SetLevelID** [= *String*]

O argumento *objeto* sempre será o objeto TreeView. O valor para **SetLevelID** é o identificador deste nível.

Comentários**Aplica-se**

TreeView.

Veja também

Propriedade GetLevelID

Descrição

Retorna o valor de identificação do item clicado.

Sintaxe

[*Result* =] *objeto*.**GetLevelID**

O argumento *objeto* sempre será o objeto TreeView. O valor para *Result* é o **GetLevelID**, o valor para **GetLevelID** é o identificador de nível do item clicado.

Comentários

Aplica-se

TreeView.

Veja também

Propriedade IsLevelID

Descrição

Retorna verdadeiro se este nível for o nível clicado.

Sintaxe

[*Result* =] *objeto*.**IsLevelID**

O argumento *objeto* sempre será o objeto TreeView. O valor para *Result* é o **IsLevelID**, o valor de **IsLevelID** indica se este é o nível clicado.

Comentários

Aplica-se

TreeView.

Veja também

Propriedade ItemHaveMore

Descrição

Especifica que os próximos nós adicionados deverão ser mostrados com o sinal de mais independente que ter mais itens filhos ou não.

Sintaxe

objeto.**ItemHaveMore** [= *Boolean*]

O argumento *objeto* sempre será o objeto TreeView. O valor para **ItemHaveMore** indica a obrigatoriedade de ser mostrado o sinal de mais incondicional.

Comentários

O código abaixo ilustra o uso da propriedade **ItemHaveMore**.

```
TreeView.BeginTree
  TreeView.BeginNode "Ação 1", "ScriptAction.asp?Action=1"
    TreeView.BeginNode "Ação 2", "ScriptAction.asp?Action=2"
      TreeView.ItemHaveMore = True
      TreeView.BeginNode "Ação 3", "ScriptAction.asp?Action=3"
      TreeView.EndNode
    TreeView.EndNode
  TreeView.EndNode
TreeView.EndTree
```

Aplica-se
TreeView.

Veja também

Propriedade ItemState

Descrição

Retorna qual é o estado do item clicado. Os valores possíveis são "O" para abrindo um item e "C" para fechando um item.

Sintaxe

[*Result* =] *objeto*.ItemState

O argumento *objeto* sempre será o objeto TreeView. O valor para *Result* é o **ItemState**, o valor de **ItemState** indica qual o estado do item clicado.

Comentários

Aplica-se
TreeView.

Veja também

Método BeginTree

Descrição

Inicializa a construção da árvore (TreeView).

Sintaxe

objeto.BeginTree

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>TreeView</i> .

Comentários

O código abaixo ilustra o uso do método **BeginTree**.

```
TreeView.DefaultImage = "Folder"
TreeView.AddImage "Folder", "/ZTITools/Img/Folder/"
TreeView.AddImage "User", "/HiperTools/Img/Folder/", "aberto", "fechado"
TreeView.BeginTree
    TreeView.AddItem "1º item", "ScriptAction.asp?Action=1", , "User"
    TreeView.BeginNode "1º nó", "ScriptAction.asp?Action=2"
        TreeView.AddItem "1º item", "ScriptAction.asp?Action=3"
        TreeView.AddSubCaption "SubTítulo", "ScriptAction.asp?Action=4"
    TreeView.EndNode
TreeView.EndTree
```

Aplica-se
TreeView.

Veja também

Método EndTree

Descrição

Finaliza e constrói a árvore (TreeView).

Sintaxe

objeto.**EndTree**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>TreeView</i> .

Comentários

O código abaixo ilustra o uso do método **EndTree**.

```
TreeView.DefaultImage = "Folder"
TreeView.AddImage "Folder", "/HiperTools/Img/Folder/"
TreeView.AddImage "User", "/HiperTools/Img/Folder/", "aberto", "fechado"
TreeView.BeginTree
    TreeView.AddItem "1º item", "ScriptAction.asp?Action=1", , "User"
    TreeView.BeginNode "1º nó", "ScriptAction.asp?Action=2"
        TreeView.AddItem "1º item", "ScriptAction.asp?Action=3"
        TreeView.AddSubCaption "SubTítulo", "ScriptAction.asp?Action=4"
    TreeView.EndNode
TreeView.EndTree
```

Aplica-se

TreeView.

Veja também

Método BeginNode

Descrição

Adiciona um novo nó. Todo os itens e nós adicionados após este método e antes do EndNode respectivo serão tratados como filhos deste nó.

Sintaxe

objeto.**BeginNode** *String1*[, *String2*, *String3*, *String4*, *Boolean*]

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>TreeView</i> .
<i>String1</i>	Requerido. Texto que será mostrado na descrição do nó.
<i>String2</i>	Opcional. URL a ser executada quando o texto do nó for clicado.
<i>String3</i>	Opcional. Título que aparecerá quando o usuário passa com o mouse sobre o texto do nó.

<i>String4</i>	Opcional. Nome da image adicionada para ser utilizada como ícone do nó.
<i>Boolean</i>	Opcional. Indica se o nó já deverá ser mostrado aberto.

Comentários

O código abaixo ilustra o uso do método **BeginNode**.

```
TreeView.DefaultImage = "Folder"
TreeView.AddImage "Folder", "/HiperTools/Img/Folder/"
TreeView.AddImage "User", "/HiperTools/Img/Folder/", "aberto", "fechado"
TreeView.BeginTree
    TreeView.AddItem "1º item", "ScriptAction.asp?Action=1", , "User"
    TreeView.BeginNode "1º nó", "ScriptAction.asp?Action=2"
        TreeView.AddItem "1º item", "ScriptAction.asp?Action=3"
        TreeView.AddSubCaption "SubTítulo", "ScriptAction.asp?Action=4"
    TreeView.EndNode
TreeView.EndTree
```

Aplica-se

TreeView.

Veja também

Método EndNode

Descrição

Finaliza a definição de um nó e indica que os próximo itens adicionados não farão mais parte deste nó.

Sintaxe

objeto.**EndNode**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>TreeView</i> .

Comentários

O código abaixo ilustra o uso do método **EndNode**.

```
TreeView.DefaultImage = "Folder"
TreeView.AddImage "Folder", "/HiperTools/Img/Folder/"
TreeView.AddImage "User", "/HiperTools/Img/Folder/", "aberto", "fechado"
TreeView.BeginTree
    TreeView.AddItem "1º item", "ScriptAction.asp?Action=1", , "User"
    TreeView.BeginNode "1º nó", "ScriptAction.asp?Action=2"
        TreeView.AddItem "1º item", "ScriptAction.asp?Action=3"
        TreeView.AddSubCaption "SubTítulo", "ScriptAction.asp?Action=4"
    TreeView.EndNode
TreeView.EndTree
```

Aplica-se
TreeView.

Veja também

Método AddItem

Descrição

Adiciona um item. Um item não pode conter filhos.

Sintaxe

objeto.AddItem String1[, String2, String3, String4]

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>TreeView</i> .
<i>String1</i>	Requerido. Texto que será mostrado na descrição do item.
<i>String2</i>	Opcional. URL a ser executada quando o texto do item for clicado.
<i>String3</i>	Opcional. Título que aparecerá quando o usuário passa com o mouse sobre o texto do item.
<i>String4</i>	Opcional. Nome da image adicionada para ser utilizada como ícone do item.

Comentários

O código abaixo ilustra o uso do método **AddItem**.

```
TreeView.DefaultImage = "Folder"
TreeView.AddImage "Folder", "/HiperTools/Img/Folder/"
TreeView.AddImage "User", "/HiperTools/Img/Folder/", "aberto", "fechado"
TreeView.BeginTree
    TreeView.AddItem "1º item", "ScriptAction.asp?Action=1", , "User"
    TreeView.BeginNode "1º nó", "ScriptAction.asp?Action=2"
        TreeView.AddItem "1º item", "ScriptAction.asp?Action=3"
        TreeView.AddSubCaption "SubTítulo", "ScriptAction.asp?Action=4"
    TreeView.EndNode
TreeView.EndTree
```

Aplica-se
TreeView.

Veja também

Método AddSubCaption

Descrição

Adiciona títulos secundários a um item existente. Útil para a realização de tarefas auxiliares do item.

Sintaxe

objeto.AddSubCaption *String1*[, *String2*, *String3*, *String4*]

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>TreeView</i> .
<i>String1</i>	Requerido. Texto que será mostrado na descrição do item.
<i>String2</i>	Opcional. URL a ser executada quando o texto do item for clicado.
<i>String3</i>	Opcional. Título que aparecerá quando o usuário passa com o mouse sobre o texto do item.
<i>String4</i>	Opcional. Alvo (target) onde a URL deverá ser aberta.

Comentários

O código abaixo ilustra o uso do método **AddSubCaption**.

```
TreeView.DefaultImage = "Folder"
TreeView.AddImage "Folder", "/HiperTools/Img/Folder/"
TreeView.AddImage "User", "/HiperTools/Img/Folder/", "aberto", "fechado"
TreeView.BeginTree
    TreeView.AddItem "1º item", "ScriptAction.asp?Action=1", , "User"
    TreeView.BeginNode "1º nó", "ScriptAction.asp?Action=2"
        TreeView.AddItem "1º item", "ScriptAction.asp?Action=3"
        TreeView.AddSubCaption "SubTítulo", "ScriptAction.asp?Action=4"
    TreeView.EndNode
TreeView.EndTree
```

Aplica-se

TreeView.

Veja também

Método AddImage

Descrição

Adiciona uma imagem à lista de imagens do TreeView para posterior utilização nos ícones dos itens e nós adicionados. Estas imagens serão utilizadas pelo nome do prefixo.

Sintaxe

objeto.AddImage *String1*, *String2*[, *String3*, *String4*, *String5*]

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>TreeView</i> .
<i>String1</i>	Requerido. Prefixo do nome da imagem a ser adicionada.
<i>String2</i>	Requerido. Caminho URL da imagem.

<i>String3</i>	Requerido. Sufixo do nome da imagem para quando ela estiver aberta, o valor padrão é "Open".
<i>String4</i>	Requerido. Sufixo do nome da imagem para quando ela estiver aberta, o valor padrão é "Close".
<i>String5</i>	Requerido. Extensão da item, o valor padrão é ".gif".

Comentários

O código abaixo ilustra o uso do método **AddImage**.

```
TreeView.DefaultImage = "Folder"
TreeView.AddImage "Folder", "/HiperTools/Img/Folder/"
TreeView.AddImage "User", "/HiperTools/Img/Folder/", "aberto", "fechado"
TreeView.BeginTree
    TreeView.AddItem "1º item", "ScriptAction.asp?Action=1", , "User"
    TreeView.BeginNode "1º nó", "ScriptAction.asp?Action=2"
        TreeView.AddItem "1º item", "ScriptAction.asp?Action=3"
        TreeView.AddSubCaption "SubTítulo", "ScriptAction.asp?Action=4"
    TreeView.EndNode
TreeView.EndTree
```

Aplica-se

TreeView.

Veja também

Objeto Upload

Objeto Upload

Objeto que realiza o *upload* de um arquivo enviado através do protocolo HTTP utilizando o método "post".

Sintaxe

Set *Upload* = Server.CreateObject("HiperTools30.Upload")

Comentários

O formulário que for enviar o arquivo deve ser construído utilizando o "multipart/form-data" como forma de codificação dos dados enviados, como exemplificado abaixo através do método *EncType* do objeto *Form*.

O que permitirá ao usuário enviar os arquivos para *upload* serão os "inputs" do tipo *file* adicionados ao formulário. Quando um "input" do tipo *file* é adicionado ao formulário o navegador desenha uma caixa de edição ao lado do botão de procura, este botão será utilizado para que o usuário possa selecionar o nome do arquivo que será enviado.

É necessário que se defina o tipo de envio dos dados do formulário como "post" e de forma alguma utilizando o método "get", como exemplificado abaixo através do método *BeginForm* do objeto *Form*

Quando o objeto *Upload* é criado lê os dados enviados através do protocolo HTTP no instante de sua criação. Quando um objeto *Upload* é criado o conteúdo da resposta HTTP é totalmente limpo e armazenado dentro do objeto, portanto, para acessar o conteúdo de outros dados do formulário enviado será necessário que se utilize a propriedade *Form* ou a propriedade *Items* do objeto *Upload*.

O código abaixo ilustra como obter o objeto *Upload* e como utilizar alguma de suas propriedades e métodos.

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#INCLUDE VIRTUAL="/HiperTools/All2.inc"-->
<!--#INCLUDE VIRTUAL="/HiperTools/Objetcs.inc"-->
<%
REM =====
REM /UploadAction.asp
REM -----
REM Descricao : Sistema de upload utilizando o HiperTools 3.0
REM Criação : 12:00h 15/11/99
REM Local : Brasília/DF
REM Elaborado : José da Silva
REM Versao : 1.0.0
REM -----

' Executa o procedimento principal do sistema
MainBody

REM =====
REM Mostra a página inicial do sistema
REM -----
Private Sub ShowFirstPage

' O "EncType" do formulário deve ser igual a "multipart/form-data"
Form.EncType = "multipart/form-data"

' Inicializa o formulário de Upload onde o usuário entra com o nome e pode
```

```
' enviar até dois arquivos de uma vez.
Form.BeginForm Initializer.ScriptName & "?DoUpload=True", "frmUpload"

' Campo onde o usuário entrará com sua identificação
Show.HTML "Autor<BR>"
Form.Input "", "txtAuthor"

' Permite ao usuário escolher o primeiro arquivo a enviar
Show.Message "Arquivo 1<BR>"
Form.Input "file", "txtFileName1"

' Permite ao usuário escolher o segundo arquivo a enviar
Show.Message "Arquivo 2<BR>"
Form.Input "file", "txtFileName2"

' Adiciona o botão de envio
Show.BR
Form.Submit "Submit", Upload

' Finaliza o formulário
Form.EndForm

End Sub
REM -----
REM Fim de ShowFirstPage
REM =====

REM =====
REM Realiza o Upload
REM -----
Private Sub DoUpload

' Só é possível realizar o upload utilizando-se do método "post"
If Request.ServerVariables("REQUEST_METHOD")= "post" Then

' Cria o objeto de upload
Dim Upload, Index
Set Upload = Server.CreateObject("HiperTools30.Upload")

' Mostra o conteúdo do formulário que foi enviado
Show.Message "<H3>Conteúdo do formulário</H3>"
For Index = 1 to Upload.FormCount
    If (Not Upload.Form(Index).IsFile) Then
        Show.Message Upload.Form(Index).Name & "=" & Upload.Form(Index)
    End If
Next

' Mostra as informações dos arquivos enviados
Show.Message "<H3>Informações dos arquivos</H3>"
For Index = 1 to Upload.FormCount
    If (Upload.Form(Index).IsFile) Then
        Show.Message Upload.Form(Index).Name & "=" & _
            Upload.Form(Index).UserFilename & "=" & _
            Upload.Form(Index).TotalBytes & "=" & _
            Upload.Form(Index).ServerName & "=" & _
    End If
Next
```

```
' Configura o comportamento do objeto
Upload.MaxBytes      = 15360
Upload.OverwriteFiles = True
Upload.Path          = "c:\inertpub\wwwroot\XYZ\upload"

' Salva o primeiro arquivo enviado
Upload.SaveAs File.ExtractFileName(Upload.ServerName)

' Salva o segundo arquivo enviado
Upload.Items("txtFileName2").SaveAs _
    File.ExtractFileName(Upload.Items("txtFileName2").ServerName)

' Destrói o objeto de upload
Set Upload = Nothing

End If

End Sub
REM -----
REM Fim de ShowFirstPage
REM =====

REM =====
REM Corpo Principal do sistema
REM -----
Private Sub MainBody

    ' Inicializa o HTML
    Default.HTMLBegin
    Default.HeadAll "Upload"
    Default.BodyBegin

    ' Se o usuário já envio os arquivos para upload
    If Request.QueryString("DoUpload")= verdadeiro Then

        ' Realiza o upload
        DoUpload

    ' Senão
    Else

        ' Mostra um formulário que permite ao usuário enviar os arquivos
        ShowFirstPage

    End If

    ' Finaliza o HTML
    Default.BodyEnd
    Default.HTMLEnd

End Sub
REM -----
REM Fim do MainBody
REM =====

REM -----
REM Fim do UploadAction.asp
```



```
REM =====  
%>
```

Veja também

Objeto UploadItems, Objeto UploadItem.

Coleções Form

Descrição

Retorna o objeto *UploadItems* com a lista dos valores contidos no formulário enviado pelo usuário, esta propriedade é *read-only*.

Sintaxe

[*UploadItems* =] **objeto.Form**

O argumento *objeto* sempre será o objeto *Upload*. O valor para **Form** indica se deverá ou não.

Comentários

Esta propriedade é semelhante à propriedade *Items* e existe apenas por questão de compatibilidade com a propriedade *Form* do objeto *Request*.

O código abaixo ilustra o uso da propriedade *Form*

```
Dim Upload, Index  
Set Upload = Server.CreateObject("HiperTools30.Upload")  
For Index = 1 to Upload.FormCount  
    Show.Message Upload.Form(Index).Name & "=" & Upload.Form(Index).Value  
Next
```

Aplica-se

Upload

Veja também

Objeto UploadItems, Objeto UploadItem.

Coleções Items

Descrição

Retorna o objeto *UploadItems* com a lista dos valores contidos no formulário enviado pelo usuário, esta propriedade é *read-only*.

Sintaxe

[*UploadItems* =] **objeto.Items**

Comentários

Esta propriedade é semelhante à propriedade *Form*

O código abaixo ilustra o uso da propriedade **Items**.

```
Dim Upload, Index  
Set Upload = Server.CreateObject("HiperTools30.Upload")  
For Index = 1 to Upload.Items.Count  
    Show.Message Upload.Items(Index).Name & "=" & Upload.Items(Index).Value  
Next
```

Aplica-se

Upload

Veja também

Objeto UploadItems, Objeto UploadItem

Propriedade MaxBytes

Descrição

Especifica qual será o número total de bytes que os arquivos enviados poderão ter, não permitindo que sejam gravados com tamanho superior ao especificado.

Sintaxe

objeto.**MaxBytes** [= Integer]

O argumento *objeto* sempre será o objeto *Upload*.

Comentários

Caso o valor dessa propriedade seja diferente de zero, este valor será utilizado para restringir a gravação.

O código abaixo ilustra o uso da propriedade **MaxBytes**.

```
Dim Upload
Set Upload = Server.CreateObject("HiperTools30.Upload")
Upload.MaxBytes = 15360
Upload.OverwriteFiles = True
Upload.Path = "c:\inertpub\wwwroot\XYZ\upload"
Upload.SaveAs File.ExtractFileName(Upload.ServerName)
Set Upload = Nothing
```

Aplica-se

Upload

Veja também

Save, SaveAs.

Propriedade OverwriteFiles

Descrição

Especifica se será necessário sobrepor o arquivo existente caso ele já exista no destino.

Sintaxe

objeto.**OverwriteFiles** [= Boolean]

O argumento *objeto* sempre será o objeto *Upload*.

Comentários

Caso o valor da propriedade seja igual a *verdadeiro* indicará que será necessário sobrepor o arquivo caso ele exista no destino, se o valor for igual a *falso* indicará que no caso do arquivo já existir no destino será gerado um erro.

O código abaixo ilustra o uso da propriedade **OverwriteFiles**.

```
Dim Upload
Set Upload = Server.CreateObject("HiperTools30.Upload")
Upload.MaxBytes = 15360
Upload.OverwriteFiles = True
Upload.Path = "c:\inertpub\wwwroot\XYZ\upload"
Upload.SaveAs File.ExtractFileName(Upload.ServerName)
Set Upload = Nothing
```

Aplica-se
Upload

Veja também
Save, SaveAs.

Propriedade Path

Descrição

Especifica o caminho no qual os arquivos deverão ser armazenados quando for executado o método *Save* ou *SaveAs*.

Sintaxe

objeto.**Path** [= *Boolean*]

O argumento *objeto* sempre será o objeto *Upload*.

Comentários

O código abaixo ilustra o uso da propriedade **Path**.

```
Dim Upload
Set Upload = Server.CreateObject("HiperTools30.Upload")
Upload.MaxBytes = 15360
Upload.OverwriteFiles = True
Upload.Path = "c:\inertpub\wwwroot\XYZ\upload"
Upload.SaveAs File.ExtractFileName(Upload.ServerName)
Set Upload = Nothing
```

Aplica-se
Upload

Veja também
Save, SaveAs, ServerName.

Propriedade ServerName

Descrição

Retorna o nome completo do primeiro arquivo enviado, informando onde ele deverá ser salvo ou onde ele foi salvo.

Sintaxe

[*String* =] *objeto*.**ServerName**

O argumento *objeto* sempre será o objeto *Upload*.

Comentários

O código abaixo ilustra o uso da propriedade **ServerName**.

```
Dim Upload
Set Upload = Server.CreateObject("HiperTools30.Upload")
Upload.MaxBytes = 15360
Upload.OverwriteFiles = True
Show.Message Upload.ServerName
```

Retorna: "c:\inertpub\wwwroot\Arquivo.ext"

```
Upload.Path = "c:\inertpub\wwwroot\XYZ\upload"
```

Retorna: "c:\inertpub\wwwroot\XYZ\upload\Arquivo.ext"

```
Upload.SaveAs File.ExtractFileName(Upload.ServerName)  
Upload.Path = "c:\inertpub\wwwroot\XYZ\NewUpload"
```

Retorna: "c:\inertpub\wwwroot\XYZ\Upload\Arquivo.ext"

```
Set Upload = Nothing
```

Aplica-se
Upload

Veja também
Save, SaveAs, Path.

Método Save

Descrição

Salva o arquivo enviado no caminho indicado na propriedade *Path* e com o nome sugerido inicialmente pelo objeto. O caminho sugerido inicialmente pelo objeto é o caminho raiz da aplicação e o nome sugerido é o nome do arquivo na máquina do usuário.

Sintaxe

objeto.Save

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Upload</i> .

Comentários

O código abaixo ilustra o uso do método **Save**.

```
Dim Upload  
Set Upload = Server.CreateObject("HiperTools30.Upload")  
Upload.MaxBytes = 15360  
Upload.OverwriteFiles = True  
Upload.Path = "c:\inertpub\wwwroot\XYZ\upload"  
Upload.Save  
Set Upload = Nothing
```

Aplica-se
Upload

Veja também
SaveAs, Path, ServerName, OverwriteFiles, MaxBytes.

Método SaveAs

Descrição

Salva o arquivo enviado no caminho indicado no path e com o nome indicado no parametro *String*. O caminho sugerido inicialmente pelo objeto é o caminho raiz da aplicação.

Sintaxe

objeto.SaveAs [*String*]

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Upload</i> .
<i>String</i>	Requerido. Novo nome do arquivo a ser salvo.

Comentários

O código abaixo ilustra o uso do método **SaveAs**.

```
Dim Upload
Set Upload = Server.CreateObject("HiperTools30.Upload")
Upload.MaxBytes = 15360
Upload.OverwriteFiles = True
Upload.Path = "c:\inertpub\wwwroot\XYZ\upload"
Upload.SaveAs File.ExtractFileName(Upload.ServerName)
Set Upload = Nothing
```

Aplica-se

Upload

Veja também

Save, Path, ServerName, OverwriteFiles, MaxBytes.

Objeto UploadItem

Objeto UploadItem

Objeto da coleção *UploadItems* que armazena os dados do formulário enviado e os arquivos submetidos para *upload*. Este objeto é utilizado internamente pelos objetos *Upload* e *UploadItems*.

Comentários

Quando o objeto *Upload* é criado ele cria uma coleção de *UploadItem* que são gerenciados pelo objeto *UploadItems*. Cada objeto *UploadItem* possui as informações de um item do formulário enviado para *upload*.

Se este item do formulário não for um item de envio de arquivo, receberá um tratamento semelhante ao oferecido pelo *Request.Form*, ou seja, o valor da propriedade *Name* será o nome do item do formulário e o valor da propriedade *Value* será o conteúdo do item.

Se for um item de envio de arquivo o valor da propriedade *Name* será o nome do item no formulário e o valor da propriedade *Value* será o conteúdo do arquivo enviado. As demais propriedades serão apenas utilizadas quando o item for um item de envio de arquivo.

Veja também

Objeto Upload, Objeto UploadItems.

Propriedade IsFile

Descrição

Informa se o item enviado é um item de envio de arquivo ou se é apenas um item normal de formulário. Esta propriedade é *read-only*.

Sintaxe

[*Boolean* =] *objeto.IsFile*

O argumento *objeto* sempre será o objeto *UploadItem*.

Comentários

Caso o valor retornado pela propriedade **IsFile** seja *verdadeiro* indica que este é um item de envio de arquivo, caso seja *falso* indica que é um item de formulário normal.

O código abaixo ilustra o uso da propriedade **IsFile**.

```
' Mostra as informações dos arquivos enviados
Show.Message "<H3>Informações dos arquivos</H3>"
For Index = 1 to Upload.FormCount
    If (Upload.Form(Index).IsFile) Then
        Show.Message Upload.Form(Index).Name & "=" & _
            Upload.Form(Index).UserFilename & "=" & _
            Upload.Form(Index).TotalBytes & "=" & _
            Upload.Form(Index).ServerName
    End If
Next
```

Aplica-se

UploadItem.

Veja também

ServerName, UserFilename, TotalBytes, Value.

Propriedade Name

Descrição

Retorna o nome do item adicionado a coleção UploadItems, equivale ao nome do item adicionado ao formulário HTML enviado através do método "post" do protocolo HTTP. Esta propriedade é *read-only*.

Sintaxe

[String =] *objeto*.Name

O argumento *objeto* sempre será o objeto UploadItem.

Comentários

O código abaixo ilustra o uso da propriedade **Name**.

```
' Mostra as informações dos arquivos enviados
Show.Message "<H3>Informações dos arquivos</H3>"
For Index = 1 to Upload.FormCount
    If (Upload.Form(Index).IsFile) Then
        Show.Message Upload.Form(Index).Name & "=" & _
            Upload.Form(Index).UserFilename & "=" & _
            Upload.Form(Index).TotalBytes & "=" & _
            Upload.Form(Index).ServerName
    End If
Next
```

Aplica-se

UploadItem.

Veja também

Propriedade Value

Descrição

Retorna o valor do item adicionado à coleção de UploadItems, equivale ao valor do item adicionado no formulário HTML enviado através do método "post" do protocolo HTTP. Esta propriedade é *read-only*.

Sintaxe

[Variant =] *objeto*.Value

O argumento *objeto* sempre será o objeto UploadItem.

Comentários

Caso o item seja um arquivo o valor retornado será o conteúdo do arquivo, caso contrário será o valor do item do formulário.

O código abaixo ilustra o uso da propriedade **Value**.

```
' Mostra as informações dos arquivos enviados
Show.Message "<H3>Informações dos arquivos</H3>"
For Index = 1 to Upload.FormCount
    If (Upload.Form(Index).IsFile) Then
        Show.Message Upload.Form(Index).Name & "=" & _
            Upload.Form(Index).Value
    End If
Next
```

Aplica-se

UploadItem.

Veja também

IsFile, *ServerName*, *UserFilename*, *TotalBytes*.

Propriedade UserFilename

Descrição

Retorna o nome completo original do arquivo na máquina do usuário que o enviou arquivo. Esta propriedade é *read-only*.

Sintaxe

[Result =] *objeto*.**UserFilename**

O argumento *objeto* sempre será o objeto UploadItem.

Comentários

O código abaixo ilustra o uso da propriedade **UserFilename**.

```
' Mostra as informações dos arquivos enviados
Show.Message "<H3>Informações dos arquivos</H3>"
For Index = 1 to Upload.FormCount
    If (Upload.Form(Index).IsFile) Then
        Show.Message Upload.Form(Index).Name & "=" & _
            Upload.Form(Index).UserFilename & "=" & _
            Upload.Form(Index).TotalBytes & "=" & _
            Upload.Form(Index).ServerName
    End If
Next
```

Aplica-se

UploadItem.

Veja também

IsFile, *ServerName*, *TotalBytes*, *Value*.

Propriedade TotalBytes

Descrição

Retorna o número total de bytes do arquivo enviado. Esta propriedade é *read-only*.

Sintaxe

[Result =] *objeto*.**TotalBytes**

O argumento *objeto* sempre será o objeto UploadItem.

Comentários

O código abaixo ilustra o uso da propriedade **TotalBytes**.

```
' Mostra as informações dos arquivos enviados
Show.Message "<H3>Informações dos arquivos</H3>"
For Index = 1 to Upload.FormCount
    If (Upload.Form(Index).IsFile) Then
        Show.Message Upload.Form(Index).Name & "=" & _
            Upload.Form(Index).UserFilename & "=" & _
            Upload.Form(Index).TotalBytes & "=" & _
            Upload.Form(Index).ServerName
    End If
Next
```

Next

Aplica-se

UploadItem.

Veja também

IsFile, ServerName, UserFilename, TotalBytes, Value.

Propriedade ServerName

Descrição

Retorna o nome completo onde o nome arquivo enviado foi ou será salvo. Esta propriedade é *read-only*.

Sintaxe

[Result =] *objeto*.**ServerName**

O argumento *objeto* sempre será o objeto UploadItem.

Comentários

O código abaixo ilustra o uso da propriedade **ServerName**.

```
' Mostra as informações dos arquivos enviados
Show.Message "<H3>Informações dos arquivos</H3>"
For Index = 1 to Upload.FormCount
    If (Upload.Form(Index).IsFile) Then
        Show.Message Upload.Form(Index).Name & "=" & _
            Upload.Form(Index).UserFilename & "=" & _
            Upload.Form(Index).TotalBytes & "=" & _
            Upload.Form(Index).ServerName
    End If
Next
```

Aplica-se

Upload, UploadItem.

Veja também

Objeto Upload, IsFile, UserFilename, TotalBytes, Value.

Método Save

Descrição

Salva o arquivo enviado no caminho indicado na propriedade *Path* e com o nome sugerido inicialmente pelo objeto. O caminho sugerido inicialmente pelo objeto é o caminho raiz da aplicação e o nome sugerido é o nome do arquivo na máquina do usuário.

Sintaxe

objeto.**Save**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto UploadItem.

Comentários

O código abaixo ilustra o uso do método **Save**.

```
Dim Upload
Set Upload = Server.CreateObject("HiperTools30.Upload")
Upload.MaxBytes = 15360
Upload.OverwriteFiles = True
Upload.Path = "c:\inertpub\wwwroot\XYZ\upload"
Upload.Save
Set Upload = Nothing
```

Aplica-se

Upload, UploadItem.

Veja também

Objeto Upload, IsFile, ServerName, UserFilename, TotalBytes, Value, SaveAs.

Método SaveAs

Descrição

Salva o arquivo enviado no caminho indicado no path e com o nome indicado no parametro *String*. O caminho sugerido inicialmente pelo objeto é o caminho raiz da aplicação.

Sintaxe

objeto.**SaveAs** [*String*]

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto UploadItem.
<i>String</i>	Requerido. Novo nome do arquivo a ser salvo.

Comentários

O código abaixo ilustra o uso do método **SaveAs**.

```
Dim Upload
Set Upload = Server.CreateObject("HiperTools30.Upload")
Upload.MaxBytes = 15360
Upload.OverwriteFiles = True
Upload.Path = "c:\inertpub\wwwroot\XYZ\upload"
Upload.SaveAs File.ExtractFileName(Upload.ServerName)
Set Upload = Nothing
```

Aplica-se

Upload, UploadItem.

Veja também

Objeto Upload, IsFile, ServerName, UserFilename, TotalBytes, Value, Save.

Objeto UploadItems

Objeto UploadItems

Objeto que mantém a coleção com os dados enviados pelo formulário através do método "post" do protocolo HTTP. Contém também os dados dos arquivos enviados por *Upload*.

Comentários

Este objeto não pode ser criado diretamente, sendo criado apenas internamente pelo objeto *Upload* para permitir o acesso indexado aos objetos *UploadItem* criados também internamente pelo objeto *Upload*.

Veja também

Objeto Upload, Objeto UploadItem.

Propriedade Count

Descrição

Retorna a quantidade de itens adicionados a essa coleção. Esta propriedade é *read-only*.

Sintaxe

[Integer =] *objeto*.**Count**

O argumento *objeto* sempre será o objeto *Upload*.

Comentários

O código abaixo ilustra o uso da propriedade **Count**.

```
Dim Upload, Index
Set Upload = Server.CreateObject("HiperTools30.Upload")
For Index = 1 to Upload.Items.Count
    Show.Message Upload.Items(Index).Name & "=" & Upload.Items(Index).Value
Next
```

Aplica-se

UploadItems.

Veja também

Item.

Propriedade Item

Descrição

Retorna um objeto do tipo *UploadItem* previamente adicionado à coleção. Esta propriedade é *read-only*.

Sintaxe

[*UploadItem* =] *objeto*.**Item**(*Variant*)

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto UploadItems.
<i>Variant</i>	Requerido. Índice que identifica o objeto. Pode ser o número ou o nome do item no formulário de envio.

Comentários

O código abaixo ilustra o uso da propriedade **Item**.

```
Dim Upload, Index
Set Upload = Server.CreateObject("HiperTools30.Upload")
For Index = 1 to Upload.Items.Count
    Show.Message Upload.Items(Index).Name & "=" & Upload.Items(Index).Value
Next
```

Aplica-se

UploadItems.

Veja também

Objeto UploadItem, Item.

Propriedade Name

DescriçãoRetorna o valor do item informado pelo índice. Esta propriedade é *read-only*.**Sintaxe**[String =] objeto.**Value**(Variant)

Parte	Descrição
<i>Result</i>	Opcional. Nome retornado pela propriedade Name .
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto UploadItems.
<i>Variant</i>	Requerido. Índice que identifica o nome. Pode ser o número ou o nome do item no formulário de envio.

ComentáriosO código abaixo ilustra o uso da propriedade **Name**.**Aplica-se**

UploadItems, UploadItem.

Veja também

Objeto UploadItem, Count, Item.

Propriedade Value

DescriçãoRetorna o valor do item informado pelo índice. Esta propriedade é *read-only*.**Sintaxe**[Variant =] objeto.**Value**(Variant)

Parte	Descrição
<i>Result</i>	Opcional. Valor retornado pela propriedade Value .
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto UploadItems.
<i>Variant</i>	Requerido. Índice que identifica o valor. Pode ser o número ou o nome do item no formulário de envio.

Comentários

O código abaixo ilustra o uso da propriedade **Value**.

```
Dim Upload, Index
Set Upload = Server.CreateObject("HiperTools30.Upload")
For Index = 1 to Upload.Items.Count
    Show.Message Upload.Items(Index).Name & "=" & Upload.Items(Index).Value
Next
```

Aplica-se

UploadItems, UploadItem.

Veja também

Objeto UploadItem, Item, Count.

Objeto URL

Objeto URL

Este objeto visa facilitar a manipulação das URLs que utilizam o mesmo HREF sem a necessidade de ficar repetindo comandos.

Sintaxe

```
Set URL = Server.CreateObject("HiperTools30.URL")
```

Propriedade Ampersand

Descrição

Define ou retorna o "?" ou o "&" durante a criação de uma QueryString para URL's completas.

Sintaxe

```
objeto.Ampersand [= String]
```

O argumento *objeto* sempre será o objeto *URL*.

Comentários

O valor para o **Ampersand** deve ser um símbolo (? ou &).

Seu valor padrão é "?". Após sua primeira utilização seu valor é alterado para "&" automaticamente.

O código abaixo ilustra o uso da propriedade **Ampersand**.

```
Response.Write URL.Ampersand  
Retorna: ?  
Response.Write URL.Ampersand  
Retorna: &
```

Aplica-se

URL

Veja também

Propriedade Parameters

Descrição

Retorna um objeto *Pack* contendo as QueryStrings adicionadas no escopo de um BeginURL e EndURL.

Sintaxe

```
objeto.Parameters
```

O argumento *objeto* sempre será o objeto *URL*.

Comentários

O código abaixo ilustra o uso da propriedade **Parameters**.

```
URL.BeginURL Initializer.ScriptName  
URL.Add efQueryStrAction, efQSActionList  
URL.Add efQueryStrWhat, conWhatChamaTecnico
```

```
URL.Show "Listagem de Chamados Técnicos"
Show.BR

URL.Parameters(efQueryStrWhat) = conWhatUsuario
URL.Show "Listagem de Usuários"
Show.BR
URL.EndURL
```

Aplica-se
URL

Veja também
Pack

Método Add

Descrição

Adiciona uma opção e um parametro em uma QueryString na criação de uma ou mais URL's dentro do escopo de um BeginURL e EndURL.

Sintaxe

objeto.Add String1, String2

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>URL</i> .
<i>String1</i>	Requerido. <i>String1</i> é a opção do QueryString. Opcao=.
<i>String2</i>	<i>String2</i> é o parametro da opção do QueryString. =parametro.

Comentários

O código abaixo ilustra o uso do método **Add**.

```
URL.BeginURL Initializer.ScriptName
URL.Add efQueryStrAction, efQSActionSummary
URL.Add efQueryStrWhat, conWhatChamaTecnico
URL.Show "Relatório por Setor", URL.Equal(efQueryStrList, 1)
Show.BR
URL.Show "Relatório por Usuário", URL.Equal(efQueryStrList, 2)
Show.BR
URL.EndURL
```

Aplica-se
URL

Veja também

Método AddEditForm

Descrição

Adiciona opções específicas para o Objeto EditForm em uma QueryString na criação de uma ou mais URL's dentro do escopo de um BeginURL e EndURL.

Sintaxe

objeto.**AddEditForm** [*String1*, *String2*, *String3*, *String4*, *String5*, *String6*]

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>URL</i> .
<i>String1</i>	Opcional. <i>String1</i> é a opção de edição. Define se o formulário permitirá edição de registros. efQueryStrEditable.
<i>String2</i>	Opcional. <i>String2</i> é a opção de ação. Define qual ação (inclusão, exclusão, alteração, etc...) será tomada pelo EditForm. efQueryStrAction.
<i>String3</i>	Opcional. <i>String3</i> é a opção de escolha. Define qual formulário será apresentado. efQueryStrWhat.
<i>String4</i>	Opcional. <i>String4</i> é a opção de filtro. Define uma situação para um determinado formulário de forma a filtrar os registros. efQueryStrFilter
<i>String5</i>	Opcional. <i>String5</i> é a opção de valores padrão. Define valores padrão para um ou mais campos na inclusão de um registro. efQueryStrDefaults.
<i>String6</i>	Opcional. <i>String6</i> é a opção de listagens. Define qual listagem será apresentada a partir de opções criadas pelo usuário. efQueryStrList.

Comentários

O código abaixo ilustra o uso do método **AddEditForm**.

```
URL.BeginURL Initializer.ScriptName
URL.AddEditForm efQueryStrEditable, efQSActionSummary, conWhatChamaTecnico,
"intCodigo > 10 and strNome like '%Pessoa%'", , conList1
URL.Show "Relatório 1"
URL.EndURL
Show.BR

URL.BeginURL Initializer.ScriptName
URL.AddEditForm efQueryStrEditable, efQSActionAdd, conWhatUser, ,
"intCodigo=20;strNome='Pessoa'"
URL.Show "Inclusão de Usuário"
Show.BR
URL.EndURL
```

Aplica-se

URL

Veja também

Método BeginURL

Descrição

Inicia a criação de uma URL ou uma sequência de URL's que serão direcinadas para o mesmo local (HREF).

Sintaxe

objeto.**BeginURL** [*String*]

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>URL</i> .
<i>String</i>	Opcional. <i>String</i> é o nome do arquivo que será referenciado pelo HREF. Se não for indicado, o objeto vai utilizar o nome do arquivo atual (<i>Initializer.ScriptName</i>).

Comentários

O código abaixo ilustra o uso do método **BeginURL**.

```
URL.BeginURL Initializer.ScriptName
URL.Add efQueryStrAction, efQSActionSummary
URL.Add efQueryStrWhat, conWhatChamaTecnico
URL.Show "Relatório por Setor", URL.Equal(efQueryStrList, 1)
Show.BR
URL.Show "Relatório por Usuário", URL.Equal(efQueryStrList, 2)
Show.BR
URL.EndURL
```

Aplica-se

URL

Veja também

Método EndURL

Descrição

Finaliza o comando de criação de URL.

Sintaxe

objeto.**EndURL**

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>URL</i> .

Comentários

O código abaixo ilustra o uso do método **EndURL**.

```
URL.BeginURL Initializer.ScriptName
URL.Add efQueryStrAction, efQSActionSummary
URL.Add efQueryStrWhat, conWhatChamaTecnico
URL.Show "Relatório por Setor", URL.Equal(efQueryStrList, 1)
Show.BR
URL.Show "Relatório por Usuário", URL.Equal(efQueryStrList, 2)
Show.BR
URL.EndURL
```

Aplica-se

URL

Veja também

Método Equal

Descrição

Retorna um parametro completo para a criação de uma QueryString.

Sintaxe

objeto.**Equal** (*String1* , *String2*)

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>URL</i> .
<i>String1</i>	Requerido. <i>String1</i> é a opção do QueryString. Opcao=.
<i>String2</i>	<i>String2</i> é o parametro da opção do QueryString. =parametro.

Comentários

O código abaixo ilustra o uso do método **Equal**.

```
URL.BeginURL Initializer.ScriptName
URL.Add efQueryStrAction, efQSActionSummary
URL.Add efQueryStrWhat, conWhatChamaTecnico
URL.Show "Relatório por Setor", URL.Equal(efQueryStrList, 1)
Show.BR
Response.Write URL.GetURL "Relatório por Usuário", URL.Equal(efQueryStrList,
2)
Show.BR
URL.EndURL
```

Aplica-se

URL

Veja também

Método GetURL

Descrição

Retorna a URL completa de acordo com os parametros fornecidos no BeginURL e Add.

Sintaxe

objeto.GetURL (*String1* [, *String2*])

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>URL</i> .
<i>String1</i>	Requerido. <i>String1</i> é o texto que será apresentado para referência da URL.
<i>String2</i>	<i>String2</i> é a opção completa do QueryString. Opcao=parametro.

Comentários

O código abaixo ilustra o uso do método **GetURL**.

```
URL.BeginURL Initializer.ScriptName
URL.Add efQueryStrAction, efQSActionSummary
URL.Add efQueryStrWhat, conWhatChamaTechnico
Response.Write URL.GetURL "Relatório por Setor", URL.Equal(efQueryStrList, 1)
Show.BR
Response.Write URL.GetURL("Relatório por Usuário", URL.Equal(efQueryStrList,
2))
Show.BR
URL.EndURL
```

Aplica-se

URL

Veja também

Método Show

Descrição

Escreve uma URL completa de acordo com os parametros fornecidos no URLBegin e Add.

Sintaxe

objeto.Show *String1* [, *String2*]

Parte	Descrição
<i>objeto</i>	Requerido. Sempre terá que ser um objeto <i>URL</i> .
<i>String1</i>	Requerido. <i>String1</i> é o texto que será apresentado para referência da URL.
<i>String2</i>	<i>String2</i> é a opção completa do QueryString. Opcao=parametro.

Comentários

O código abaixo ilustra o uso do método **Show**.

```
URL.BeginURL Initializer.ScriptName
URL.Add efQueryStrAction, efQSActionSummary
URL.Add efQueryStrWhat, conWhatChamaTecnico
URL.Show "Relatório por Setor", URL.Equal(efQueryStrList, 1)
Show.BR
URL.Show "Relatório por Usuário", URL.Equal(efQueryStrList, 2)
Show.BR
URL.EndURL
```

Aplica-se

URL

Veja também

Objeto Validate

Objeto Validate

Esse objeto facilita validações tanto no cliente, utilizando JavaScript, quanto no servidor, via ASP.

Propriedade AllowJSAlert

Descrição

Retorna ou define se deve-se permitir que os métodos que apresentam funções JavaScript apresentem alertas ("alert(...)")

Sintaxe

objeto.**AllowJSAlert** [= *Boolean*]

O argumento *objeto* sempre terá que ser um objeto *Validate*.

Comentários

O código abaixo ilustra o uso do método **AllowJSAlert**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")

Response.Write Validate.AllowJSAlert

Retorna: True

Validate.AllowJSAlert = False

Validate.FormJavaValidCPF

Retorna: Toda função de validação de CPF só que não apresenta a
mensagem:
    "Este CPF é inválido"
```

Aplica-se

Validate

Veja também

FormJavaValidSepDate, FormJavaValidDate, FormJavaValidCEP, FormJavaValidCGC,
FormJavaValidCPF, FormJavaValidCPFCGC, FormJavaValidEMail

Propriedade JSStart

Descrição

Retorna ou define o estado do objeto de validação. Se já foi iniciada uma *TAG SCRIPT*

Sintaxe

objeto.**JSStart** [= *Boolean*]

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre será um objeto <i>Validate</i> .

Comentários

O código abaixo ilustra o uso do método **JSStart**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")

If Not Validate.JSStart Then
    Validate.BeginJS
End If

Response.Write "function fazAlgo(){
Response.Write "    if(true){
Response.Write "        ..."
Response.Write "    }"
Response.Write "}"

Validate.EndJS 'Aqui JSStart volta a ser False
```

Aplica-se

Validate

Veja também

BeginJS, EndJS, BeginJava, EndJava

Método BeginJava**Descrição**

Inicia a criação de uma função para a validação de um formulário específico. Os parâmetros especificam o nome do formulário específico, definido utilizando-se o objeto Form, com o método:

Form.BeginForm "meuAplicativo.asp?O=XZ", "frmFormulario"

Sintaxe

objeto.**BeginJava** *String1*, *String2*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .
<i>String1</i>	Requerido. <i>String1</i> é o nome da função em JavaScript que será criada.
<i>String2</i>	Requerido. <i>String2</i> é o nome do formulário que será validado.

Comentários

O código abaixo ilustra o uso do método **BeginJava**.

```
Dim Validate, Form
Set Validate = Server.CreateObject("HiperTools30.Validate")
Set Form      = Server.CreateObject("HiperTools30.Form")

Form.BeginForm "meuAplicativo.asp?O=XY", "frmFormulario", , ,
"validaForm()"
```

```
Show.HTML "Texto Requirdo:"  
Show.BR  
  
Form.Text "txtText", 10, , 100 `Apresenta um INPUT TEXT com MAXLENGTH  
100  
  
Form.Submit "btnSubmit", "Enviar"  
  
Validate.BeginJava "validaForm", "frmFormulario"  
  
Retorna: <SCRIPT LANGUAGE="JavaScript">  
        Function validaForm()  
  
Validate.JavaRequired "txtText", "O texto é requerido"  
  
Retorna: Código em javascript para garantir que o campo txtText seja  
preenchido, se a pessoa não preenchê-lo e tentar enviá-lo, será  
apresentada a mensagem "O texto é requerido"  
  
Validate.EndJava  
  
Form.EndForm
```

Aplica-se

Validate

Veja também

EndJava, JavaRequired, JavaCompare, JavaClearField, JavaFocus

Método BeginJS

Descrição

Inicia o sistema de validação no cliente usando JavaScript. Ele verifica se já foi iniciado ou aberto a *TAG SCRIPT*, se não estiver aberta (JSStart) então abre faz com que JSStart seja True

Sintaxe

objeto.BeginJS

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .

Comentários

O código abaixo ilustra o uso do método **BeginJS**.

```
If Not Validate.JSStart Then  
    Validate.BeginJS  
End If  
  
Response.Write "function fazAlgo(){"  
Response.Write "    if(true){"  
Response.Write "        ..."  
Response.Write "    }"  
Response.Write "}"
```

```
Validate.EndJS 'Aqui JSStart volta a ser False
```

Aplica-se

Validate

Veja também

EndJS, JSStart

Método CompareDate

Descrição

??????????????????

Sintaxe*objeto.CompareDate String1, String2*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .
<i>String1</i>	Requerido. ??????????????????????
<i>String2</i>	Requerido. ??????????????????????

ComentáriosO código abaixo ilustra o uso do método **CompareDate**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método EndJava

Descrição

Termina a criação de uma função para a validação de um formulário específico definido em BeginJava.

Sintaxe*objeto.EndJava*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .

ComentáriosO código abaixo ilustra o uso do método **EndJava**.

```
Dim Validate, Form
Set Validate = Server.CreateObject("HiperTools30.Validate")
Set Form      = Server.CreateObject("HiperTools30.Form")

Form.BeginForm "meuAplicativo.asp?O=XY", "frmFormulario", , ,
"validaForm()"
Show.HTML "Texto Requirido:"
Show.BR

Form.Text "txtText", 10, , 100 'Apresenta um INPUT TEXT com MAXLENGTH
100

Form.Submit "btnSubmit", "Enviar"

Validate.BeginJava "validaForm", "frmFormulario"

Retorna: <SCRIPT LANGUAGE="JavaScript">
        Function validaForm()

Validate.JavaRequired "txtText", "O texto é requerido"

Retorna: Código em javascript para garantir que o campo txtText seja
preenchido, se a pessoa não preenchê-lo e tentar enviá-lo, será
apresentada a mensagem "O texto é requerido"

Validate.EndJava

Form.EndForm
```

Aplica-se

Validate

Veja também

BeginJava, JavaRequired, JavaCompare, JavaClearField, JavaFocus

Método EndJS

Descrição

Termina o sistema de validação no cliente usando JavaScript. Neste método se altera o estado da Propriedade JSStart

Sintaxe

objeto.EndJS

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .

Comentários

O código abaixo ilustra o uso do método **EndJS**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")
If Not Validate.JSStart Then
```

```
Validate.BeginJS
End If

Response.Write "function fazAlgo(){ "
Response.Write "  if(true){ "
Response.Write "    ..."
Response.Write "  } "
Response.Write "}"

Validate.EndJS `Aqui JSStart volta a ser False
```

Aplica-se

Validate

Veja também

BeginJS, JSStart

Método FormatDate

Descrição

??????????????????

Sintaxe*objeto.FormatDate String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .
<i>String</i>	Requerido. ??????????????????????

ComentáriosO código abaixo ilustra o uso do método **FormatDate**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método FormatTime

Descrição

??????????????????

Sintaxe*objeto.FormatTime String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .
<i>String</i>	Requerido. ??????????????????????

Comentários

O código abaixo ilustra o uso do método **FormatTime**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método FormJavaCompareDate

Descrição

????????????????????

Sintaxe

objeto.**FormJavaCompareDate** *String1, String2, String3, String4, String5*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .
<i>String1</i>	Requerido. ??????????????????????
<i>String2</i>	Requerido. ??????????????????????
<i>String3</i>	Requerido. ??????????????????????
<i>String4</i>	Requerido. ??????????????????????
<i>String5</i>	Requerido. ??????????????????????

Comentários

O código abaixo ilustra o uso do método **FormJavaCompareDate**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método FormJavaCompareDateSep

Descrição

????????????????

Sintaxe

objeto.**FormJavaCompareDateSep** *String1, String2, String3, String4, String5, _*
String6, String7, String8, String9, Boolean, String10

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .
<i>String1</i>	Requerido. ??????????????????
<i>String2</i>	Requerido. ??????????????????
<i>String3</i>	Requerido. ??????????????????
<i>String4</i>	Requerido. ??????????????????
<i>String5</i>	Requerido. ??????????????????
<i>String6</i>	Requerido. ??????????????????
<i>String7</i>	Requerido. ??????????????????
<i>String8</i>	Requerido. ??????????????????
<i>String9</i>	Requerido. ??????????????????
<i>Boolean</i>	Requerido. ??????????????????
<i>String10</i>	Requerido. ??????????????????

Comentários

O código abaixo ilustra o uso do método **FormJavaCompareDateSep**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método FormJavaValidCEP

Descrição

????????????????

Sintaxe*objeto*.**FormJavaValidCEP** *String1, String2, String3*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .
<i>String1</i>	Requerido. ??????????????????????
<i>String2</i>	Requerido. ??????????????????????
<i>String3</i>	Requerido. ??????????????????????

Comentários

O código abaixo ilustra o uso do método **FormJavaValidCEP**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método FormJavaValidCGC

Descrição

????????????????????

Sintaxe*objeto*.**FormJavaValidCGC** *String1, String2, String3*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .
<i>String1</i>	Requerido. ??????????????????????
<i>String2</i>	Requerido. ??????????????????????
<i>String3</i>	Requerido. ??????????????????????

Comentários

O código abaixo ilustra o uso do método **FormJavaValidCGC**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método FormJavaValidCPF

Descrição

??????????????????

Sintaxe

objeto.**FormJavaValidCPF** *String1, String2, String3*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .
<i>String1</i>	Requerido. ??????????????????????
<i>String2</i>	Requerido. ??????????????????????
<i>String3</i>	Requerido. ??????????????????????

Comentários

O código abaixo ilustra o uso do método **FormJavaValidCPF**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método FormJavaValidEmail

Descrição

??????????????????

Sintaxe

objeto.**FormJavaValidEmail** *String1, String2, String3*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .
<i>String1</i>	Requerido. ??????????????????????
<i>String2</i>	Requerido. ??????????????????????
<i>String3</i>	Requerido. ??????????????????????

Comentários

O código abaixo ilustra o uso do método **FormJavaValidEmail**.

```
Dim Validate  
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método FormJavaValidSepDate

Descrição

Valida o formulário de data separada.

Sintaxe

objeto.**FormJavaValidSepDate**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .

Comentários

O código abaixo ilustra o uso do método **FormJavaValidSepDate**.

```
Dim Validate  
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método FormJavaValidDate

Descrição

Valida o campo do formulário com data.

Sintaxe

objeto.**FormJavaValidDate** *String1, String2, String3, Boolean, String4*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .
<i>String1</i>	Requerido. <i>String1</i> é o nome do campo data que será testado.
<i>String2</i>	Requerido. <i>String2</i> é o nome do formulário que será

	validado.
<i>Boolean</i>	Requerido. <i>Boolean</i> é se a data poderá ser maior que hoje.
<i>String3</i>	Requerido. <i>String3</i> é a mensagem que será apresentada no caso de erro.

Comentários

O código abaixo ilustra o uso do método **FormJavaValidDate**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método IsCGC

Descrição

??????????????????

Sintaxe

objeto.IsCCG String

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .
<i>String</i>	Requerido. ??????????????????????

Comentários

O código abaixo ilustra o uso do método **IsCGC**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método IsCPF

Descrição

??????????????????

Sintaxe

objeto.IsCPF String

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .
<i>String</i>	Requerido. ??????????????????????

Comentários

O código abaixo ilustra o uso do método **IsCPF**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método IsEmail

Descrição

??????????????????

Sintaxe

objeto.IsEmail String

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .
<i>String</i>	Requerido. ??????????????????????

Comentários

O código abaixo ilustra o uso do método **IsEmail**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método IsValidDate

Descrição

??????????????????

Sintaxe*objeto.IsValidDate String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .
<i>String</i>	Requerido. ??????????????????????

Comentários

O código abaixo ilustra o uso do método **IsValidDate**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se*Validate***Veja também**

Método IsValidTime

Descrição*????????????????????***Sintaxe***objeto.IsValidTime String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .
<i>String</i>	Requerido. ??????????????????????

Comentários

O código abaixo ilustra o uso do método **IsValidTime**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se*Validate***Veja também**

Método JavaCompare

Descrição

Cria uma função para comparação entre dois campos.

Sintaxe

objeto.**JavaCompare** *String1, String2, String3*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .
<i>String1</i>	Requerido. <i>String1</i> é o nome de um dos campos que serão comparados.
<i>String2</i>	Requerido. <i>String2</i> é o nome de um dos campos que serão comparados.
<i>String3</i>	Requerido. <i>String3</i> é a mensagem que será apresentada no caso de erro.

Comentários

O código abaixo ilustra o uso do método **JavaCompare**.

```
Dim Validate  
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

BeginJava, EndJava, JavaRequired

Método JavaClearField

Descrição

Apaga o conteúdo do campo.

Sintaxe

objeto.**JavaClearField** *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .
<i>String</i>	Requerido. <i>String</i> é o nome do campo que será apagado.

Comentários

O código abaixo ilustra o uso do método **JavaClearField**.

```
Dim Validate  
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método JavaEmpty

Descrição

Retorna a função Vazio em JavaScript que verifica se um certo campo (INPUT) de um formulário está vazio. Se estiver vazio apresenta passada por parâmetro (em JavaScript, não no próprio ASP).

Sintaxe

objeto.**JavaEmpty**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .

Comentários

O código abaixo ilustra o uso do método **JavaEmpty**.

```
Dim Validate, Form
Set Validate = Server.CreateObject("HiperTools30.Validate")
Set Form      = Server.CreateObject("HiperTools30.Form")

Form.BeginForm "meuAplicativo.asp?O=XY", "frmFormulario", , ,
"validaForm()"
Show.HTML "Texto Requirido:"
Show.BR

Form.Text "txtText", 10, , 100 'Apresenta um INPUT TEXT com MAXLENGTH
100

Validate.JavaEmpty

Validate.BeginJS

Validate.EndJS

Validate.JavaRequired "txtText", "O texto é requerido"
```

Aplica-se

Validate

Veja também

Método JavaFocus

Descrição

Posiciona o cursor no campo definido.

Sintaxe

objeto.**JavaFocus** *String*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .
<i>String</i>	Requerido. <i>String</i> é o nome do campo que o cursor deverá se posicionar.

Comentários

O código abaixo ilustra o uso do método **JavaFocus**.

```
Dim Validate  
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método JavaRequired

Descrição

Cria uma função para verificação se o campo esta ou não vazio. Retorna mensagem no cliente e posiciona no campo quando estiver errado.

Sintaxe

objeto.**JavaRequired** *String1*, *String2*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .
<i>String1</i>	Requerido. <i>String1</i> é o nome do campo que será testado.
<i>String2</i>	Requerido. <i>String2</i> é a mensagem que será apresentada no caso de erro.

Comentários

O código abaixo ilustra o uso do método **JavaRequired**.

```
Dim Validate  
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método JSMaskVal

Descrição

Valida a máscara apresentada no campo.

Sintaxe

objeto.**JSMaskVal**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .

Comentários

O código abaixo ilustra o uso do método **JSMaskVal**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método JSDateToString

Descrição

??????????????????

Sintaxe

objeto.**JSDateToString**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .

Comentários

O código abaixo ilustra o uso do método **JSDateToString**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método JSIsDate

Descrição

??????????????????

Sintaxe

objeto.**JSIsDate**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .

Comentários

O código abaixo ilustra o uso do método **JSIsDate**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método JSLeadingZeroes

Descrição

??????????????????

Sintaxe

objeto.**JSLeadingZeroes**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .

Comentários

O código abaixo ilustra o uso do método **JSLeadingZeroes**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método JSLeft

Descrição

??????????????????

Sintaxe

objeto.**JSLeft** *String1*, *String2*, *String3*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .

Comentários

O código abaixo ilustra o uso do método **JSLeft**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método JSMid

Descrição

??????????????????

Sintaxe

objeto.**JSMid** *String1*, *String2*, *String3*

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .

Comentários

O código abaixo ilustra o uso do método **JSMid**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método JSMidToEnd

Descrição

??????????????????

Sintaxe

objeto.**JSMidToEnd**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .

Comentários

O código abaixo ilustra o uso do método **JSMidToEnd**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método JSNormalize

Descrição

??????????????????

Sintaxe

objeto.**JSNormalize**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .

Comentários

O código abaixo ilustra o uso do método **JSNormalize**.

```
Dim Validate
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Método JSStringToDate

Descrição

??????????????????

Sintaxe

objeto.**JSStringToDate**

Parte	Descrição
<i>Objeto</i>	Requerido. Sempre terá que ser um objeto <i>Validate</i> .

Comentários

O código abaixo ilustra o uso do método **JSStringToDate**.

```
Dim Validate  
Set Validate = Server.CreateObject("HiperTools30.Validate")
```

Aplica-se

Validate

Veja também

Índice

Índice

A

A Equipe do HiperTools v3.0, 4
A Zevallos, 3
Adovbs.inc, 8
Apresentação do Ambiente, 9

B

Barcode, 14
 Métodos, 14
 ShowI2o5, 14
Border, 18
 Propriedades, 18
 Color, 18
 Width, 18
Browse, 20
 Page, 21
 PageSize, 22
 Propriedades, 20
 Bar, 23
 BarPages, 20
 Current, 20
 EndOfPage, 24
 IncLine, 25
 Line, 21
 QueryString, 22
 RecordSet, 23

C

Cabeçalho dos programas, 10
Cancelando o registro do componente
 HiperTools® v3.0, 8
Character, 28
 DrawButton, 29
 DrawString, 29
 Font, 28
 FontDirectory, 28
 Métodos, 29
 Propriedades, 28
Coleções
 Form, 393
 Items, 393

Comandos Disponíveis, 33
Configurando um site no servidor IIS, 8
Connection, 32

Constantes

conConnAccess, 32
conConnDBase, 32
conConnDBase5, 33
conConnDBaseIV, 33
conConnDriverAccess, 33
conConnDriverdBaseIII, 33
conConnExcel, 32
conConnExcel3, 33
conConnExcel4, 34
conConnExcel5, 34
conConnExcel7, 34
conConnExcel97, 34
conConnFoxPro20, 34
conConnFoxPro25, 34
conConnFoxPro26, 34
conConnParadox, 32
conConnParadox3, 34
conConnParadox4, 34
conConnParadox5, 34
conConnSQL, 32
conConnText, 32, 34

Métodos, 42

Clear, 42
ConnectionString, 43

Propriedades, 37

CommandTimeout, 36
ConnectionTimeout, 37
ConnectionType, 37
Database, 38
DBQ, 39
DefaultDir, 39
Password, 40
ServerAddress, 41
UserID, 41

Constantes

Alphabetic Index, 101
Character Case, 100
conConnAccess, 32
conConnDBase, 32
conConnDBase5, 33
conConnDBaseIV, 33
conConnDriverAccess, 33

conConnDriverdBaselll, 33
conConnExcel, 32
conConnExcel3, 33
conConnExcel4, 34
conConnExcel5, 34
conConnExcel7, 34
conConnExcel97, 34
conConnFoxPro20, 34
conConnFoxPro25, 34
conConnFoxPro26, 34
conConnParadox, 32
conConnParadox3, 34
conConnParadox4, 34
conConnParadox5, 34
conConnSQL, 32
conConnText, 32, 34
Connection Types, 101
Data Types, 92
efCharCaseLower, 100
efCharCaseNormal, 100
efCharCaseUpper, 100
efCondDisable, 99, 100
efCondEnable, 99, 100
efDataTypeBit, 94
efDataTypeChar, 93
efDataTypeDateTime, 93
efDataTypeFloat, 92, 303, 304
efDataTypeInt, 93
efDataTypeMoney, 93
efDataTypeReal, 93, 303, 304
efDataTypeSmallInt, 93
efDataTypeText, 93
efDataTypeTinyInt, 93
efDataTypeVarChar, 93
efFldTypeAtualization, 97
efFldTypeCheck, 96
efFldTypeColor, 97
efFldTypeEMail, 96
efFldTypeFile, 97
efFldTypeHTTP, 96
efFldTypeImage, 97
efFldTypeLookup, 96
efFldTypePassword, 96
efFldTypeRadio, 96
efFldTypeSelect, 96
efFldTypeSeparateDate, 97
efFldTypeText, 95
efFldTypeTextArea, 96
efFldTypeUF, 96
efNext, 102
efNotRequired, 94
efQSActionAdd, 91
efQSActionAdvancedFind, 91
efQSActionCommonFind, 91
efQSActionCopy, 91
efQSActionDelete, 92
efQSActionEdit, 92
efQSActionEditor, 91
efQSActionList, 91
efQSActionSave, 92
efQSActionSaveAdd, 92
efQSActionSaveCopy, 92
efQSActionScreen, 92
efQSActionSummary, 91
efQueryStrAction, 89
efQueryStrDefaults, 90
efQueryStrEditable, 89
efQueryStrEditableStr, 90
efQueryStrFilter, 90
efQueryStrFind, 90
efQueryStrGrid, 90
efQueryStrGridStr, 90
efQueryStrList, 90
efQueryStrTab, 90
efQueryStrWhat, 89
efRelIntegrCascade, 94
efRelIntegrNullifies, 95
efRelIntegrRestricted, 94
efRelType1to1, 95
efRelType1toN, 95
efRequired, 94
efValLocClient, 89, 240, 269
efValLocServer, 89
efValOptCEP, 99
efValOptCGC, 98
efValOptCGCCPF, 99
efValOptCompareDates, 98
efValOptCPF, 98
efValOptCPFCGC, 99
efValOptDate, 98
efValOptDateMToday, 98
efValOptEMail, 98
efValOptNone, 97
efValOptSepDate, 98
efValOptSepDateMToday, 98
efValOptTime, 98
Field Disable Condition, 99
Field Requirement, 94
Field Types, 95
Identity, 102
Operator types, 100
QueryString Action Values, 91

QueryString Parameters, 89
Relation Type, 95
Relational Integrity, 94
[RgbBinary](#), 244
[RgbDword](#), 244
[RgbExpandSz](#), 244
[RgbSz](#), 244
Validation Location, 89
Validation Options, 97
Constants.inc, 8
Convenções utilizadas, 9
Copiando os arquivos do HiperTools® v3.0
para o servidor, 8

D

Default, 46
Métodos
 BeginBody, 70
 BeginHead, 71
 BeginHTML, 72
 EndBody, 73
 EndHead, 74
 EndHTML, 75
 PageFooterDefault, 76
 WriteBaseFont, 77
 WriteHead, 78
 WriteLink, 79
 WriteMeta, 80
 WritePageFooter, 82
 WritePageHeader, 82
Propriedades, 46
 BaseFont, 46
 BodyALink, 46
 BodyBackground, 47
 BodyBGColor, 47
 BodyBGProperties, 48
 BodyLeftMargin, 48
 BodyLink, 48
 BodyOnLoad, 49
 BodyOnUnLoad, 49
 BodyStyle, 50
 BodyText, 50
 BodyTopMargin, 51
 BodyVLink, 51
 BodyWidth, 52
 CenterBody, 52
 FooterCopyright, 53
 FooterSogestions, 54

FooterTitle, 54
[HeaderTitle](#), 54
IgnoreBodyWidth, 55
LinkCopyrightHRef, 56
[LinkCopyrightTitle](#), 56
[LinkHomeHRef](#), 57
[LinkHomeTitle](#), 57
[LinkMadeHRef](#), 58
[LinkMadeTitle](#), 58
[LinkStyleSheetHRef](#), 59
[MetaAuthor](#), 59
[MetaCategory](#), 60
[MetaCharSet](#), 60
[MetaContentType](#), 61
[MetaDescription](#), 61
[MetaDistribution](#), 62
[MetaExpires](#), 62
[MetaFormatter](#), 62
[MetaGenerator](#), 63
[MetaKeyPhrase](#), 63
[MetaKeywords](#), 64
[MetaLocale](#), 64
[MetaPicsLabel](#), 65
[MetaPragma](#), 65
[MetaProduct](#), 66
[MetaRefresh](#), 66
[MetaReply](#), 67
[MetaTechnology](#), 67
PageFooter, 67
PageHeader, 68
[SiteCounter](#), 68
[SourceCode](#), 69
[TableBorder](#), 69
[UserID](#), 70

E

EditForm, 86
Constantes
 Alphabetic Index, 101
 Character Case, 100
 Connection Types, 101
 Data Types, 92
 efCharCaseLower, 100
 efCharCaseNormal, 100
 efCharCaseUpper, 100
 efCondDisable, 99, 100
 efCondEnable, 99, 100
 efDataTypeBit, 94

- efDataTypeChar, 93
- efDataTypeDateTime, 93
- efDataTypeFloat, 92, 303, 304
- efDataTypeInt, 93
- efDataTypeMoney, 93
- efDataTypeReal, 93, 303, 304
- efDataTypeSmallInt, 93
- efDataTypeText, 93
- efDataTypeTinyInt, 93
- efDataTypeVarChar, 93
- efFldTypeAtualization, 97
- efFldTypeCheck, 96
- efFldTypeColor, 97
- efFldTypeEmail, 96
- efFldTypeFile, 97
- efFldTypeHTTP, 96
- efFldTypeImage, 97
- efFldTypeLookup, 96
- efFldTypePassword, 96
- efFldTypeRadio, 96
- efFldTypeSelect, 96
- efFldTypeSeparatedDate, 97
- efFldTypeText, 95
- efFldTypeTextArea, 96
- efFldTypeUF, 96
- efNext, 102
- efNotRequired, 94
- efQSActionAdd, 91
- efQSActionAdvancedFind, 91
- efQSActionCommonFind, 91
- efQSActionCopy, 91
- efQSActionDelete, 92
- efQSActionEdit, 92
- efQSActionEditor, 91
- efQSActionList, 91
- efQSActionSave, 92
- efQSActionSaveAdd, 92
- efQSActionSaveCopy, 92
- efQSActionScreen, 92
- efQSActionSummary, 91
- efQueryDefaults, 90
- efQueryFilter, 90
- efQueryGrid, 90
- efQueryGridStr, 90
- efQueryList, 90
- efQueryStrAction, 89
- efQueryStrEditable, 89
- efQueryStrEditableStr, 90
- efQueryStrFind, 90
- efQueryStrWhat, 89
- efQueryTab, 90
- efRelIntegrCascade, 94
- efRelIntegrNullfies, 95
- efRelIntegrRestricted, 94
- efRelType1to1, 95
- efRelType1toN, 95
- efRequired, 94
- efValLocClient, 89, 240, 269
- efValLocServer, 89
- efValOptCEP, 99
- efValOptCGC, 98
- efValOptCGCCPF, 99
- efValOptCompareDates, 98
- efValOptCPF, 98
- efValOptCPFCGC, 99
- efValOptDate, 98
- efValOptDateMToday, 98
- efValOptEmail, 98
- efValOptNone, 97
- efValOptSepDate, 98
- efValOptSepDateMToday, 98
- efValOptTime, 98
- Field Disable Condition, 99
- Field Requirement, 94
- Field Types, 95
- Identity, 102
- Operator types, 100
- QueryString Action Values, 91
- QueryString Parameters, 89
- Relation Type, 95
- Relational Integrity, 94
- Validation Location, 89
- Validation Options, 97
- Métodos, 117
 - AddField, 133
 - AddFilter, 135
 - AddHeader, 136
 - AddOrder, 134
 - AddReport, 137
 - BeginForm, 126
 - CreateIndex, 160
 - CreateTable, 159
 - DataAddField, 121
 - DataAddIndex, 125
 - DataAddPrimaryKey, 122
 - DataAddRelation, 123
 - DataAddRelationFields, 124
 - DataIndexClustered, 126
 - DataTable, 121
 - DropTable, 159
 - EmptyListMessage, 139, 157, 158
 - EndForm, 128

- FieldAddILTarget, 153
 - FieldAddValue, 153
 - FieldAlphabeticIndex, 139
 - FieldCharCase, 144
 - FieldDisableValue, 145
 - FieldDoubleValidate, 157
 - FieldExternalLink, 151
 - FieldHint, 154
 - FieldImage, 141
 - FieldInternalLink, 152
 - FieldListChars, 147
 - FieldListSize, 146
 - FieldLookup, 155
 - FieldLookupFilter, 156
 - FieldMask, 149
 - FieldRadioColumns, 143
 - FieldSepDate, 151
 - FieldShowSize, 149
 - FieldTextAreaHeight, 140
 - FieldValue, 118
 - FormFind, 129
 - FormList, 130
 - FormTabs, 132
 - FormUnit, 131
 - IsMyAction, 117
 - IsTable, 118
 - OpenConnection, 120
 - RedirectActions, 119
 - ShowActions, 119
 - UserID, 117
 - Propriedades, 102
 - AllowIdentityUpdate, 102
 - AlphabeticIndex, 103
 - Buttons, 103
 - ConnectionString, 105
 - ConnetionType, 106
 - CurrConnection, 106
 - DebugMode, 107
 - Errors, 107
 - FilterBooleanOperator, 108
 - FirstPageLink, 108
 - LinkTarget, 109
 - ListOrdernatiosCaption, 110
 - ListReports, 110
 - ListTitle, 109
 - LockTimeOut, 111
 - parAction, 112
 - parWhat, 112
 - Password, 111
 - ReportTitle, 113
 - ShowButtons, 113
 - ShowListUpperBar, 114
 - ShowOrdernations, 114
 - ShowRecordCount, 114
 - ShowReportsLink, 115
 - ShowSQLQuery, 115
 - ShowUnitUpperBar, 116
 - Style, 116
 - UnitTitle, 116
 - EditForm.inc, 8
 - Escopo das variaveis, 9
 - Estrutura de diretórios recomendada, 7
- ## F
- File, 164
 - Copy, 164
 - DateCreated, 166
 - DateLastModified, 166
 - Exists, 164
 - ExtractFileName, 165
 - Métodos, 164
 - Size, 167
 - Font, 170
 - Propriedades, 170, 171
 - BeginFont, 172
 - Bold, 170
 - Color, 170
 - EndFont, 173
 - Face, 171
 - Italic, 171
 - Size, 171
 - Underline, 172
 - Form, 176
 - Métodos, 180
 - BeginForm, 180
 - BeginSelect, 181
 - CheckBox, 182
 - EndForm, 183
 - EndSelect, 184
 - File, 184
 - FormInput, 185
 - Hidden, 186
 - Image, 187
 - Pallette, 188
 - Password, 189
 - Radio, 190
 - Reset, 191
 - SelectChooseOption, 191

- SelectCountry, 193
- SelectMonthName, 202
- SelectOption, 203
- SelectState, 204
- SelectWeekDay, 207
- Submit, 208
- Text, 209
- TextArea, 210
- Propriedades, 176
 - AccessKey, 176
 - Alt, 176
 - Class, 177
 - Enable, 177
 - EncType, 177
 - MaxLength, 178
 - onChange, 178
 - onClick, 179
 - ReadOnly, 179
 - Title, 179

G

- global.asa, 8
- global.HiperTools30.pdf, 8

H

- HiperTools.zip, 8
- HiperTools30.dll, 8

I

- Initializer
 - Métodos, 219
 - InitDefault, 219
 - LastQueryString, 215
 - Propriedades, 217
 - ErrorMessage, 214
 - HiperToolsWorkPath, 214
 - HTTPHost, 215
 - QueryString, 216
 - ScriptFull, 216
 - ScriptName, 217
 - ScriptURL, 217
 - SiteRootDir, 218
 - WhenStarted, 218
- Instalação, 7
 - Cancelando o registro do componente HiperTools® v3.0, 8
 - Configurando um site no servidor IIS, 8

- Copiando os arquivos do HiperTools® v3.0 para o servidor, 8
- Estrutura de diretórios recomendada, 7
- Necessidades mínimas recomendadas, 7
- Registrando o componente HiperTools® v3.0, 8

L

- Licença, i
- Listas de Discussão, 5

M

- Métodos
 - Add, 225, 413
 - AddEditForm, 414
 - AddField, 133
 - AddFilter, 135
 - AddHeader, 136
 - AddImage, 387
 - AddItem, 385
 - AddOrder, 134
 - AddReport, 137
 - AddSubCaption, 386
 - AddTip, 360
 - AddTipBegin, 360
 - AddTipEnd, 361
 - AllServerVariables, 268
 - BeginBegin, 126
 - BeginBody, 70
 - BeginCell, 330
 - BeginForm, 180
 - BeginHead, 71
 - BeginHTML, 72
 - BeginItemList, 269
 - BeginNode, 383
 - BeginRow, 331
 - BeginSelect, 181
 - BeginTable, 332
 - BeginTip, 362
 - BeginTree, 382
 - BeginURL, 415
 - Bold, 270
 - BoldEnd, 270
 - BoldText, 271
 - BR, 271
 - BreakString, 293
 - Cell, 334
 - CellGraph, 335

Center, 272
CenterEnd, 272
CheckBox, 182
Clear, 42
ConnectionString, 43
Copy, 164
CreateIndex, 160
CreateKey, 243
CreateTable, 159
CreateValue, 243
CRHTML, 273
CRHTMLCR, 273
DataAddField, 121
DataAddIndex, 125
DataAddPrimaryKey, 122
DataAddRelation, 123
DataAddRelationFields, 124
DataIndexClustered, 126
DataTable, 121
DateCreated, 166
DateLastModified, 166
DayOfWeek, 293
DeleteKey, 244
DeleteValue, 245
DrawButton, 29
DrawString, 29
Drive, 234
DropTable, 159
EmptyListMessage, 139, 157, 158
EndCell, 336
EndEnd, 73
EndForm, 128, 183
EndHead, 74
EndHTML, 75
EndItemList, 274
EndNode, 384
EndRow, 337
EndSelect, 184
EndTable, 338
EndTip, 362
EndTree, 383
EndURL, 415
Equal, 416
Exists, 164
ExtractFileName, 165
FieldAddILTarget, 153
FieldAddValue, 153
FieldAlphabeticIndex, 139
FieldCharCase, 144
FieldDisableValue, 145
FieldDoubleValidate, 157
FieldExternalLink, 151
FieldHint, 154
FieldImage, 141
FieldInternalLink, 152
FieldListChars, 147
FieldListSize, 146
FieldLookup, 155
FieldLookupFilter, 156
FieldMask, 149
FieldRadioColumns, 143
FieldSepDate, 151
FieldShowSize, 149
FieldTextAreaHeight, 140
FieldValue, 118
File, 184
Find, 226
FormatCGC, 294
FormatCPF, 295
FormatDigit, 296
FormatInt, 296
FormatMoney, 297
FormatNumber, 297
FormatPhone, 298
FormFind, 129
FormInput, 185
FormJavaValidDate, 431
FormJavaValidSepDate, 430
FormList, 130
FormTabs, 132
FormUnit, 131
GetPiece, 299
GetPieceAsDate, 300
GetPieceAsStr, 301
GetPieceRest, 301
GetSlash, 236
GetURL, 417
GetWord, 302
GraphLegend, 339
Hidden, 186
HTML, 275
HTMLComment, 275
HTMLCR, 276
Image, 187, 277
ImportDate, 303
ImportInt, 305
ImportLong, 305
ImportMoney, 304
InitDefault, 219
InvertSlash, 234

IsMyAction, 117
IsTable, 118
Italic, 277
ItalicEnd, 278
ItalicText, 278
ItemList, 279
ItemListError, 280
JavaBegin, 421
JavaClearField, 435
JavaCompare, 434
JavaEmpty, 435
JavaEnd, 423
JavaFocus, 436
JavaRequired, 437
JSAler, 280
JSBegin, 422
JSEnd, 424
JSMaskVal, 437
LastQueryString, 215
LeadingZeroes, 306
Link, 281
LinkRel, 282
LinkRelStyleSheet, 282
LinkRev, 283
LongDate, 307
LongDateTime, 307
LongTime, 308
LTrimZeroes, 309
Mask, 309
Message, 284
MessageAlert, 284
MessageError, 285
Meta, 285
MetaHTTPEquiv, 286
MetaName, 286
MonthName, 310
MonthSize, 311
NbSp, 287
NormalizeAccents, 312
NormalizeNumber, 313
NormalizeString, 313
NormalSlash, 235
Only, 235
OpenConnection, 120
PageFooterDefault, 76
Pallet, 188
Paragraph, 288
Password, 189
Piece, 292
Radio, 190
RedirectActions, 119
Remove, 226
RemoveRightSlash, 236
Reset, 191
Reverse, 314
Save, 396, 404
SaveAs, 397, 404
SecondsTime, 314
SelectChooseOption, 191
SelectCountry, 193
SelectMonthName, 202
SelectOption, 203
SelectState, 204
SelectWeekDay, 207
ShortTime, 315
Show, 417
ShowActions, 119
ShowI2o5, 14
ShowPieceRest, 316
Size, 167
SkipPiece, 316
SQLDateTime, 318
SQLLong, 319
SQLLongDate, 317
SQLLongDateBR, 318
SQLString, 319
Submit, 208
Text, 209
TextArea, 210
TimeSeconds, 320
ToString, 321
Underline, 288
UnderlineEnd, 289
UnderlineText, 289
URLSlash, 237
USADate, 321
[ValueAsBinary](#), 246
[ValueAsBoolean](#), 246
[ValueAsLong](#), 247
[ValueAsString](#), 247
WriteBaseFont, 77
WriteHead, 78
WriteLink, 79
WriteMeta, 80
WritePageFooter, 82
WritePageHeader, 82

N

Necessidades mínimas recomendadas, 7
nomenclatura de objetos, 9

O

O que é o HiperTools v3.0, 2

Objetos

- Barcode, 14
- Border, 18
- Browse, 20
- Character, 28
- Connection, 32
- Default, 46
- EditForm, 86
- File, 164
- Font, 170
- Form, 176
- Initializer, 214
- Pack, 222
- PackItem, 230
- Path, 234
- Registry, 240
- RegistrySubKey, 250
- RegistrySubKeys, 254
- RegistryValue, 258
- RegistryValues, 262
- Show, 266
- String, 292
- Table, 324
- TableStyle, 342
- TreeView, 366
- Upload, 390
- UploadItem, 400
- UploadItems, 408
- URL, 412
- Validade, 420

P

Pack

- Métodos, 225
 - Add, 225
 - Find, 226
 - Remove, 226
- Propriedades, 224
 - Count, 223
 - Item, 224

PackItem

- Propriedades, 231
 - Key, 230
 - Pack, 230
 - Position, 231

- Value, 231
- padronização da documentação, 10
- Padronização dos Programas, 11
- Path, 234
 - Métodos, 236
 - Drive, 234
 - GetSlash, 236
 - IntertSlash, 234
 - NormalSlash, 235
 - Only, 235
 - RemoveRightSlash, 236
 - URLSlash, 237
- Propriedades
 - AccessKey, 176
 - AllowIdentityUpdate, 102
 - AllowJSAalert, 420
 - AlphabeticIndex, 103
 - Alt, 176
 - AlternateColor, 342
 - Ampersand, 412
 - AutoCreate, 240
 - AutoWrite, 266
 - BackgroundFormat, 342
 - Bar, 23
 - BarPages, 20
 - BaseColor, 343
 - BaseFont, 46
 - BeginFont, 172
 - BodyBackground, 47
 - BodyBGColor, 47
 - BodyBGProperties, 48
 - BodyLeftMArgin, 48
 - BodyLink, 48
 - BodyOnLoad, 49
 - BodyOnUnLoad, 49
 - BodyStyle, 50
 - BodyText, 50
 - BodyTopMargin, 51
 - BodyVLink, 51
 - BodyWidth, 52
 - Bold, 170
 - BoldyALink, 46
 - BorderColor, 343
 - BorderFormat, 344
 - BorderWidth, 344
 - Buttons, 103
 - CellAlign, 324
 - CellBackground, 345
 - CellColSpan, 324
 - CellHeight, 325

CellNoWrap, 325
CellRowSpan, 326
CellVAlign, 327
CellWidth, 327
CenterBody, 52
Class, 177
Color, 18, 170
[ColorFormat](#), 345
CommandTimeout, 36
ConnectionString, 105
ConnectionTimeout, 37
ConnectionType, 37, 106
ContentClass, 358
Count, 223, 254, 262, 408
CrLf, 266
CrLfAfter, 267
CrLfBefore, 267
CurrConnection, 106
Current, 20
CurrentChild, 379
Database, 38
[DataType](#), 258
DateSeparator, 292
DBQ, 39
DebugMode, 107
DefaultDir, 39
DefaultImage, 368
Enable, 177
EncType, 177
EndFont, 173
EndOfPage, 24
ErrorMessage, 214
Errors, 107
ExternalBorder, 346
Face, 171
FilterBooleanOperator, 108
[FirstAltBackground](#), 346
[FirstAltColor](#), 347
FirstPageLink, 108
Font, 28
FontBold, 348
FontColor, 347
FontDirectory, 28
FontItalic, 348
FontSize, 348
FontUnderline, 349
FooterCopyright, 53
FooterSugestions, 54
FooterTitle, 54
GetLevelID, 380
GraphMax, 328
GraphTotal, 328
HeadBorder, 349
HeadColor, 350
HeaderCellBackground, 350
HeaderFont, 351
HeaderRowBackground, 351
[HeaderTitle](#), 54
HiperToolsWorkPath, 214
[HKEY](#), 240
[HKeyName](#), 241
HTTPHost, 215
IgnoreBodyWidth, 55
IncLine, 25
InternalBorder, 352
InternalFont, 352
IsFile, 400
IsLevelID, 381
Italic, 171
Item, 224, 254, 262, 408
ItemBGColor, 368
ItemBold, 370
ItemFontColor, 369
ItemFontFace, 369
ItemFontSize, 370
ItemHAlign, 373
ItemHaveMore, 381
ItemHeight, 375
ItemItalic, 371
ItemMaxSize, 376
ItemNoWrap, 372
ItemState, 382
ItemStyle, 376
ItemTarget, 374
ItemUnderline, 372
ItemVAlign, 373
ItemWidth, 374
JSStart, 420
Key, 230, 241
[LastColor](#), 353
[LastWrite](#), 250
Line, 21
LinkCopyrightHRef, 56
[LinkCopyrightTitle](#), 56
[LinkHomeHRef](#), 57
[LinkHomeTitle](#), 57
[LinkMadeHRef](#), 58
[LinkMadeTitle](#), 58
[LinkStyleSheetHRef](#), 59
LinkTarget, 109
ListOrdernatiosCaption, 110
ListReports, 110

ListTitle, 109
LockTimeOut, 111
MaxBytes, 394
MaxLength, 178
[MetaAuthor](#), 59
[MetaCategory](#), 60
[MetaCharSet](#), 60
[MetaContentType](#), 61
[MetaDescription](#), 61
[MetaDistribution](#), 62
[MetaExpires](#), 62
[MetaFormatter](#), 62
[MetaGenerator](#), 63
[MetaKeyPhrase](#), 63
[MetaKeywords](#), 64
[MetaLocale](#), 64
[MetaPicsLabel](#), 65
[MetaPragma](#), 65
[MetaProduct](#), 66
[MetaRefresh](#), 66
[MetaReply](#), 67
[MetaTechnology](#), 67
[Name](#), 250, 258, 401, 409
NeedTwoTables, 353
onChange, 178
onClick, 179
OutColor, 358
OverColor, 359
OverWriteFiles, 394
Pack, 230
Padding, 329
Page, 21
PageFooter, 67
PageHeader, 68
PageSize, 22
parAction, 112
Parameters, 412
parWhat, 112
Password, 40, 111
Path, 395
Position, 231
QueryString, 22, 216
ReadOnly, 179
RecordSet, 23
RowBackground, 354
ScriptFull, 216
ScriptName, 217
ScriptURL, 217
[SecondAltBackground](#), 354
ServerAddress, 41
ServerName, 395, 403
SetLevelID, 380
ShowBorder, 329
ShowButtons, 113
ShowComments, 268
ShowListUpperBar, 114
ShowOrdernations, 114
ShowRecordCount, 114
ShowReportsLink, 115
ShowSQLQuery, 115
ShowUnitUpperBar, 116
[SiteCounter](#), 68
SiteRootDir, 218
Size, 171
[SourceCode](#), 69
Spacing, 329
Style, 116, 330
[SubKeys](#), 242
TableBackground, 355
[TableBorder](#), 69
Title, 113, 179, 355
TitleClass, 359
TitleFont, 356
TotalBytes, 402
TreeBGColor, 375
TreeStyleActive, 377
TreeStyleHover, 379
TreeStyleLink, 377
TreeStyleVisited, 378
Undeline, 172
UnitTitle, 116
UserFilename, 402
UserID, 41, 70, 117
Value, 231, 401, 409
[Values](#), 242
WhenStarted, 218
Width, 18

R

Registrando o componente HiperTools® v3.0,
8

[Registry](#), 240

[Constantes](#)

[RgBinary](#), 244
[RgDowrd](#), 244
[RgExpandSz](#), 244
[RgSz](#), 244
[Métodos](#), 243

- CreateKey, 243
- CreateValue, 243
- DeleteKey, 244
- DeleteValue, 245
- ValueAsBinary, 246
- ValueAsBoolean, 246
- ValueAsLong, 247
- ValueAsString, 247
- Propriedades, 240
 - AutoCreate, 240
 - HKEY, 240
 - HKEYName, 241
 - Key, 241
 - SubKeys, 242
 - Values, 242
- RegistrySubKey, 250
 - Propriedades
 - LastWrite, 250
 - Name, 250
 - Propriedades, 250
- RegistrySubKeys, 254
 - Propriedades, 254
 - Count, 254
 - Item, 254
- RegistryValue, 258
 - Propriedades, 258
 - DataType, 258
 - Name, 258
- RegistryValues, 262
 - Propriedades, 262
 - Count, 262
 - Item, 262
- S**
- Serviço de Suporte, 5
- Show, 266
 - Métodos, 268
 - AllServerVariables, 268
 - BeginItemList, 269
 - Bold, 270
 - BoldEnd, 270
 - BoldText, 271
 - BR, 271
 - Center, 272
 - CenterEnd, 272
 - CRHTML, 273
 - CRHTMLCR, 273
 - EndItemList, 274
 - HTML, 275
 - HTMLComment, 275
 - HTMLCR, 276
 - Image, 277
 - Italic, 277
 - ItalicEnd, 278
 - ItalicText, 278
 - ItemList, 279
 - ItemListError, 280
 - JSAlert, 280
 - Link, 281
 - LinkRel, 282
 - LinkRelStyleSheet, 282
 - LinkRev, 283
 - Message, 284
 - MessageAlert, 284
 - MessageError, 285
 - Meta, 285
 - MetaHTTPEquiv, 286
 - MetaName, 286
 - NbSp, 287
 - Paragraph, 288
 - Underline, 288
 - UnderlineEnd, 289
 - UnderlineText, 289
 - Propriedades, 266
 - AutoWrite, 266
 - CrLf, 266
 - CrLfAfter, 267
 - CrLfBefore, 267
 - ShowComments, 268
- String, 292
 - Métodos
 - BreakString, 293
 - DayOfWeek, 293
 - FormatCGC, 294
 - FormatCPF, 295
 - FormatDigit, 296
 - FormatInt, 296
 - FormatMoney, 297
 - FormatNumber, 297
 - FormatPhone, 298
 - GetPiece, 299
 - GetPieceAsDate, 300
 - GetPieceAsStr, 301
 - GetPieceRest, 301
 - GetWord, 302
 - ImportDate, 303
 - ImportInt, 305
 - ImportLong, 305
 - ImportMoney, 304
 - LeadingZeroes, 306

- LongDate, 307
- LongDateTime, 307
- LongTime, 308
- LTrimZeroes, 309
- Mask, 309
- MonthName, 310
- MonthSize, 311
- NormalizeAccents, 312
- NormalizeNumber, 313
- NormalizeString, 313
- Piece, 292
- Reverse, 314
- SecondsTime, 314
- ShortTime, 315
- ShowPieceRest, 316
- SkipPiece, 316
- SQLDateTime, 318
- SQLLong, 319
- SQLLongDate, 317
- SQLLongDateBR, 318
- SQLString, 319
- TimeSeconds, 320
- ToString, 321
- USADate, 321
- Propriedades, 292
 - DateSeparator, 292
- String.inc, 8

T

Table

- Métodos, 331
 - BeginCell, 330
 - BeginRow, 331
 - BeginTable, 332
 - Cell, 334
 - CellGraph, 335
 - EndCell, 336
 - EndRow, 337
 - EndTable, 338
 - GraphLegend, 339
- Propriedades, 324
 - CellAlign, 324
 - CellColSpan, 324
 - CellHeight, 325
 - CellNoWrap, 325
 - CellRowSpan, 326
 - CellVAlign, 327
 - CellWidth, 327

- GraphMax, 328
- GraphTotal, 328
- HeadColor, 350
- Padding, 329
- ShowBorder, 329
- Spacing, 329
- Style, 330
- TableStyle
 - Propriedades, 342
 - AlternateColor, 342
 - Background, 342
 - BaseColor, 343
 - BorderColor, 343
 - BorderFormat, 344
 - BorderWidth, 344
 - CellBackground, 345
 - ColorFormat, 345
 - ExternalBorder, 346
 - FirstAltBackground, 346
 - FirstAltColor, 347
 - FontBold, 348
 - FontColor, 347
 - FontItalic, 348
 - FontSize, 348
 - FontUnderline, 349
 - HeadBorder, 349
 - HeaderCellBackground, 350
 - HeaderFont, 351
 - HeaderRowBackground, 351
 - InternalBorder, 352
 - InternalFont, 352
 - LastColor, 353
 - NeedTwoTables, 353
 - RowBackground, 354
 - SecondAltBackground, 354
 - TableBackground, 355
 - TitleColor, 355
 - TitleFont, 356

Tip

- Métodos, 360
 - AddTip, 360
 - AddTipBegin, 360
 - AddTipEnd, 361
 - BeginTip, 362
 - EndTip, 362
- Propriedades, 358
 - ContentClass, 358
 - OutColor, 358
 - OverColor, 359
 - TitleClass, 359

- Tipo de Conexão, 32
- Tipos de Drives, 33
- TreeView, 366
 - Métodos, 382
 - AddImage, 387
 - AddItem, 385
 - AddSubCaption, 386
 - BeginNode, 383
 - BeginTree, 382
 - EndNode, 384
 - EndTree, 383
 - Propriedades, 368, 382
 - CurrentChild, 379
 - DefaultImage, 368
 - GetLevelID, 380
 - IsLevelID, 381
 - ItemBGColor, 368
 - ItemBold, 370
 - ItemFontColor, 369
 - ItemFontFace, 369
 - ItemFontSize, 370
 - ItemHAlign, 373
 - ItemHaveMore, 381
 - ItemHeight, 375
 - ItemItalic, 371
 - ItemMaxSize, 376
 - ItemNoWrap, 372
 - ItemState, 382
 - ItemStyle, 376
 - ItemTarget, 374
 - ItemUnderline, 372
 - ItemVAlign, 373
 - ItemWidth, 374
 - SetLevelID, 380
 - TreeBGColor, 375
 - TreeStyleActive, 377
 - TreeStyleHover, 379
 - TreeStyleLink, 377
 - TreeStyleVisited, 378

U

- Upload, 390
 - Coleções, 393
 - Form, 393
 - Items, 393
 - Métodos, 396
 - Save, 396
 - SaveAs, 397
 - Propriedades

- MaxBytes, 394
- OverWriteFiles, 394
- Path, 395
- ServerName, 395
- UploadItem, 400
 - Métodos, 404
 - Save, 404
 - SaveAs, 404
 - Propriedades, 400
 - IsFile, 400
 - Name, 401
 - ServerName, 403
 - TotalBytes, 402
 - UserFilename, 402
 - Value, 401
- UploadItems
 - Propriedades, 408
 - Count, 408
 - Item, 408
 - Name, 409
 - Value, 409
- URL, 412
 - Métodos, 413
 - Add, 413
 - AddEditForm, 414
 - BeginURL, 415
 - EndURL, 415
 - Equal, 416
 - GetURL, 417
 - Show, 417
 - Propriedades, 412
 - Ampersand, 412
 - Parameters, 412

V

- Validate
 - Métodos
 - FormJavaValidDate, 431
 - FormJavaValidSepDate, 430
 - JavaBegin, 421
 - JavaClearField, 435
 - JavaCompare, 434
 - JavaEmpty, 435
 - JavaEnd, 423
 - JavaFocus, 436
 - JavaRequired, 437
 - JSBegin, 422
 - JSEnd, 424
 - JSMaskVal, 437

Métodos, 422
Propriedades, 420
 AllowJSAAlert, 420

JSStart, 420
variáveis, 9