



Delphi

4.0/5.0

高级类

参
考
详
解



张 济 袁 野 王 秀 娟 编 著



清 华 大 学 出 版 社



<http://www.tup.tsinghua.edu.cn>

Delphi 4.0/5.0 高级类参考详解

张 济 袁 野 王秀娟 编著

清华大学出版社

(京)新登字 158 号

JSI 7/57

内 容 简 介

本书是 Delphi 4.0/5.0 的语言参考手册中的一本,另一本书是《Delphi 4.0/5.0 高级类参考详解》,用来配合本书的使用。

本书内容包括了进行一般应用程序设计所要使用的类。详尽地介绍了类及类中的属性,方法和事件;还详细介绍了过程和函数以及变量、数据类型、语句等,对读者的编程有很好的参考价值。

本书内容翔实,是一套极好的语言参考手册,适合广大 Delphi 程序开发人员使用。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

Delphi 4.0/5.0 高级类参考详解/张济等编著.-北京:清华大学出版社,1999
ISBN 7-302-03792-2

I .D… II .张… III .Delphi 语言-程序设计 IV .TP312

中国版本图书馆 CIP 数据核字(1999)第 63369 号

出版者:清华大学出版社(北京清华大学校内,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者:昌平环球印刷厂

发行者:新华书店总店北京发行所

开 本:787×1092 1/16 印 张:31.25 字 数:1222 千字

版 次:1999 年 11 月第 1 版 1999 年 11 月第 1 次印刷

书 号:ISBN 7-302-03792-2/TP·2133

印 数:0001~5000

定 价:55.00 元

前 言

Delphi 4.0/5.0 是 Borland 公司继 Delphi 3.0 后的又一力作。它比 Delphi 3.0 开发效率更高、功能更强大、使用也更加方便,受到了广大用户的欢迎。目前越来越多的用户已经采用 Delphi 4.0/5.0 来开发他们的软件,但目前市场上全面介绍其语言组成的手册还很少,这就给用户开发应用软件带来了诸多不便。根据这一情况,本书对 Delphi 4.0/5.0 的 VCL 类库进行了详细的介绍。

Delphi 4.0/5.0 与以前的版本相比增加了以下新功能:Advanced Debugging 模块窗口,能够显示当前所有映射到应用程序地址空间的模块;对 CodeInsight 功能作了很多改进,如类自动完成,符号预知,代码模板,类成员提示,参数提示等;增强对象接口技术和 DAX 技术,完全支持 ActiveX,可以很方便地创建、注册、安装、发布和使用 ActiveX 控件、ActiveForm 和 OLE 自动化对象;对 COM 的支持,由于 COM 对象具有语言和平台无关性,用 Delphi 4.0/5.0 创建的对象可以与用 Visual Basic, Java, C++ 及其他语言实现的对象交互;在多层 Client/Server 环境中,COM 对象可以封装商业规则,为分布式的客户提供服务;支持 Internet 应用,可方便地实现 WinSock 编程,访问 HTTP, UDP, FTP, SMTP, POP3 和 NNTP 等服务;更具开放性与灵活性,可支持多个数据库引擎,任何一个数据库引擎都能与 Delphi 4.0/5.0 的数据显示控件协调工作;MIDAS(Multi - Tiered Distributed Application Services)技术是 Delphi 4.0/5.0 的精髓,是开发多层 Client/Server 应用程序的最佳编程工具。此外,Delphi 4.0/5.0 还支持 Microsoft Transaction Server(MTS)。

鉴于 Delphi 4.0/5.0 的优良特性,为了顺应广大用户的要求,我们在积累大量开发经验的基础上编写了这套书。本套丛书是作为软件开发的参考书而非指导书编写的。

考虑到各种层次用户的需求,本套书共分两册,《Delphi 4.0/5.0 类参考详解》和《Delphi 4.0/5.0 高级类参考详解》。《Delphi 4.0/5.0 类参考详解》主要为初级使用者考虑,介绍了用户在编写 Delphi 应用程序时要用到的一些基本类和函数,对于一般应用程序的设计,如文字处理程序,这本书可以提供良好的帮助,而对高级些的应用,则需要另一本书的帮助了。《Delphi 4.0/5.0 高级类参考详解》主要为高级用户而编写,通过这些类和函数可以实现许多特殊的功能。两本书都列出了 Delphi 4.0/5.0 中的数据类型、变量、语句等内容。两本书相互配合使用,可以产生最好的效果。我们建议读者购买两本书,因为有很多的查阅内容,是在另一本书上才能找到的。而且,通过查阅这些内容,可以更好地把握 Delphi 所包含的内容,有助于从总体上提高编程的能力。

本套书内容翔实,是一套极好的语言参考手册,是从事 Delphi 程序开发和编程的广大技术人员必备的工具书,适合于各个层次的 Delphi 用户。

Delphi 4.0/5.0 的 VCL 类库十分巨大,内容也很丰富,光靠我们几个人的力量是无法完成此书的。在本书的完成过程中,还得到了许多人的大力协助,阚亚斌、陈书伟等对本书

的初稿提出了许多很好的建议,修正了书中的许多错误,在此特别感谢他们对本书的出版给予的热情支持。还有许多其他同行对本书的修订做了大量工作,在此一并表示感谢。

由于作者水平有限、经验不多,再加上时间仓促,书中的错误与不足在所难免,希望广大用户批评指正。

作 者

目 录

Actnlist 单元	1	TCorbaStubManager	39
TAction	1	TORB	40
TActionLink	1	Corbastd 单元	41
TActionList	3	TDataBrokerSkeleton	41
TContainedAction	3	TDataBrokerStub	42
TCustomAction	4	Corbavcl 单元	42
TCustomActionList	6	TCorbaVCLComponentFactory	42
Bdeprov 单元	8	Databkr 单元	43
TProvider	8	TRemoteDataModule	43
Cgiapp 单元	9	Db 单元	43
TCGIApplication	9	EDatabaseError	43
TCGIRequest	10	EDBEngineError	43
TCGIResponse	15	TADTField	44
TWinCGIRequest	21	TAggregateField	44
TWinCGIResponse	25	TArrayField	46
ComCorba 单元	25	TAutoIncField	46
TCorbaComObjectFactory	25	TBCDField	46
ComServer 单元	26	TBinaryField	49
TComServer	26	TBlobField	49
Corbacon 单元	28	TBooleanField	52
TCorbaConnection	28	TBytesField	53
Corbadm 单元	30	TCheckConstraint	54
TCorbaDataModule	30	TCheckConstraints	54
CorbaObj 单元	31	TCurrencyField	55
ICorbaObject	31	TDataLink	55
TBOA 单元	31	TDataSet	58
TCorbaDispatchStub	33	TDataSetDesigner	77
TCorbaFactory	33	TDataSetField	77
TCorbaImplementation	34	TDataSource	78
TCorbaImplementation	34	TDateField	80
TCorbaInterfaceIDManager	34	TDateTimeField	80
TCorbaListManager	35	TDefCollection	83
TCorbaObjectFactory	35	TDetailDataLink	84
TCorbaSkeleton	36	TField	84
TCorbaSkeletonManager	37	TFieldDef	96
TCorbaStub	38	TFieldDefList	98
		TFieldDefs	99
		TFieldList	100
		TFields	101

TFlatList	103	TDBRadioGroup	170
TFloatField	103	TDBRichEdit	172
TGraphicField	105	TDBText	174
TIndexDef	105	TFieldDataLink	175
TIndexDefs	106	TListSourceLink	178
TIntegerField	107	TMouse	178
TLargeintField	109	TNavDataLink	179
TLookupList	110	TPaintControl	180
TMemoField	111	Dbexcept 单元	180
TNamedItem	111	TDbEngineErrorDlg	180
TNestedTable	111	Dbgrids 单元	181
TNumericField	112	TBookmarkList	181
TObjectField	113	TColumn	183
TParam	115	TColumnTitle	188
TParams	120	TCustomDBGrid	189
TReferenceField	122	TDBGrid	202
TSmallintField	122	TDBGridColumns	202
TStringField	123	TGridDataLink	204
TTimeField	124	Dblogdlg 单元	206
TVarBytesField	125	TLoginDialog	206
TWordField	125	Dbocletl 单元	206
Dbcgrids 单元	126	TBDoleControl	206
TDBCtrlGrid	126	TDateBindItem	207
TDBCtrlGridLink	129	TDataBindings	207
TDBCtrlPanel	130	Dbpwdlg 单元	208
Dbclient 单元	130	TPasswordDialog	208
EDBClient	130	Dbtables 单元	209
EReconcileError	130	ENoResultSet	209
TAggregate	131	TBatchMove	209
TAggregates	133	TBDECallback	213
TClientBlobStream	134	TBDEDataSet	214
TClientDataSet	135	TBlobStream	222
TCustomRemoteServer	150	TDatabase	223
Dbcommon 单元	152	TDataSetUpdateObject	230
TMasterDataLink	152	TDBDataSet	231
Dbctrls 单元	154	TDBError	232
TDataSourceLink	154	TParamList	233
TDBCheckBox	154	TQuery	234
TDBComboBox	156	TSession	238
TDBEdit	158	TSessionList	246
TDBImage	160	TStoredProc	247
TDBListBox	162	TTable	250
TDBLookupComboBox	163	TUpdateSQL	257
TDBLookupControl	165		
TDBLookupListBox	167		
TDBNavigator	168		

Dbweb 单元	259	TMemIniFile	332
TDataSetTableProducer	259	Isapiapp 单元	334
TDSTableProducer	259	TISAPIApplication	334
TDSTableProducerEditor	265	TISAPIRequest	335
THTMLTableColumn	266	TISAPIResponse	340
THTMLTableColumns	268	Jpeg 单元	345
THTTPDataLink	269	TJPEGData	345
TQueryTableProducer	271	TJPEGImage	346
Ddeman 单元	272	Mask 单元	349
TDdeClientConv	272	EDBEditError	349
TDdeClientItem	276	TCustomMaskEdit	349
TDdeServerConv	277	TMaskEdit	353
TDdeServerItem	278	Masks 单元	353
Dsgnintf 单元	280	TMask	353
EPropertyError	280	Mconnect 单元	354
IFormDesigner	280	TCOMConnection	354
TComponentEditor	283	TCustomObjectBroker	354
TComponentList	285	TDCOMConnection	355
TPropertyEditor	286	TDispatchConnection	356
Dsprod 单元	291	TOLEnterpriseConnection	359
TDataSetPageProducer	291	MIDASCon 单元	360
Httpapp 单元	293	TMIDASConnection	360
TCookie	293	TRemoteServer	361
TCookieCollection	294	Mtsobj 单元	361
TCustomContentProducer	295	TMtsAutoObject	361
TCustomPageProducer	295	Mtsrdm 单元	363
TCustomWebDispatcher	297	TMTSDataModule	363
THTMLTableAttributes	299	Mxarrays 单元	365
THTMLTableCellAttributes	301	EArrayError	365
THTMLTableElementAttributes	301	ELowCapacityError	365
THTMLTableHeaderAttributes	302	EUnsupportedTypeError	365
THTMLTableRowAttributes	303	TBaseArray	365
THTMLTagAttributes	303	TSmallIntArray	368
TPageProducer	304	TValueArray	369
TWebActionItem	305	Mxcommon 单元	369
TWebActionItems	307	TDimensionItem	369
TWebApplication	308	TDimensionItems	371
TWebDispatcher	310	Mxldb 单元	373
TWebModule	313	EDimensionMapError	373
TWebRequest	313		
TWebResponse	320		
Inifiles 单元	327		
TCustomIniFile	327		
TIniFile	331		

EDimIndexError	373	Registry 单元	418
TDecisionCube	373	ERegistryException	418
TDecisionSource	375	TRegIniFile	418
Mxgraph 单元	383	TRegistry	421
TCustomDecisionGraph	383	TRegistryIniFile	429
TDecisionGraph	383	Scktcomp 单元	432
Mxgrid 单元	383	ESocketError	432
TCustomDecisonGrid	383	TAbstractSocked	432
TDecisionGrid	390	TClientSocket	435
TDisplayDim	390	TClientWinSocket	436
TDisplayDims	391	TCustomServerSocket	437
Mxpivsrc 单元	391	TCustomSocket	441
TDecisionPivot	391	TCustomWinSocket	444
Mxstore 单元	393	TServerClientThread	451
ECacheErro	393	TServerClientWinSocket	454
TCubeDim	394	TServerSocket	454
TCubeDims	395	TServerWinSocket	455
TCustomDataStore	395	TWinSocketStream	459
Mxtables 单元	400	Sconnect 单元	460
TDecisionQuery	400	EInterpreterError	460
Objbrkr 单元	400	ESocketConnectionError	460
EBrokerException	400	IDataBlock	460
TServerCollection	400	IDataIntercept	462
TServerItem	401	ISendDataBlock	462
TSimpleObjectBroker	402	ITransport	463
Olectnrs 单元	404	TDataBlockInterpreter	463
TOleContainer	404	TSocketConnection	464
TOleForm	410	TStreamedConnection	465
Olectrls 单元	410	Stdvcl 单元	467
EOleCtrlError	410	IDataBroker	467
TEnumPropDesc	410	IProvider	468
TOleControl	411	SvcMgr 单元	469
Orbpas 单元	413	TService	469
ICorbaObj	413	TServiceApplication	474
IMarshalInBuffer	413	TServiceThread	475
IMarshalOutBuffer	414	TDependencies	475
ISkeleton	416	TDependency	475
ISkeletonObject	416	Syncobjs 单元	476
IStub	416	TCriticalSection	476
IStubObject	418	TEvent	477
		THandleObject	478
		TSimpleEvent	479
		TSynchroObject	479

Sysutils 单元 479

EAbort	479
EAbstractError	479
EAccessViolation	479
EAssertionFailed	480
EControlC	480
EConvertError	480
EDivByZero	480
EExternal	480
EExternalException	480
EHeapException	480
EInOutError	481
EIntError	481
EIntfCastError	481
EIntOverflow	481
EInvalidCast	481
EInvalidOp	481
EInvalidPointer	481
EMathError	481
EOutOfMemory	482
Eoverflow	482

EPackageError	482
Eprivilege	482
EPropReadOnly	482
EPropWriteOnly	482
ERangeError	482
EStackOverflow	482
EUnderflow	482
EVariantError	482
EWin32Error	482
Exception	483
EZeroDivide	485
TMultiReadExclusiveWriteSynchronizer	485

Toolwin 单元 486

TToolWindow	486
-------------------	-----

Vclcom 单元 487

TComponentFactory	487
TToolDockForm	488
TToolDragDockObject	488

Actnlist 单元

TAction

单元 ActnList

类关系 TObject→TPersistent→TComponent→TBasicAction
→TContainedAction→TCustomAction

TAction 是创建 Action 对象的通用类, 它可以执行用户命令而不使用控件。

TAction 可以实现 TCustomAction 中的通用行为。TAction 中有许多从 TCustomAction 继承的属性, 但没有添加。可以使用 Action 对象响应用户命令。如果想保留 TCustomAction 对象成员, 则从 TAction 派生。如果想修改 TCustomAction 对象成员, 则从 TCustomAction 派生。

想增加 TAction 特殊行为, 参见 StdActns 和 DbActns 单元中的 Action 类。

属性列表

ComObject	指定支持 COM 接口的组件实现的接口引用
ComponentCount	隶属于 Components 数组属性列表中被列出组件的数量
ComponentIndex	隶属于 Components 数组属性列表的组件的位置
Components	提供对隶属于某组件的所有组件进行变址访问
ComponentState	撤消确定的动作时描述组件的当前状态
ComponentStyle	管理组件的行为
DesignInfo	窗体设计器使用的信息
Owner	拥有该组件的组件
Tag	存储整型数值为组件的要素
VCLComObject	支持 COM 组件内部使用的信息

TActionLink

单元 ActnList

类关系 TObject→TBasicActionLink

TActionLink 是 Action 连接的基类, 它将通用的 VCL 对象的属性、组件和控件连接起来。

TActionLink 引入了有关处理 Actions 和它们的 VCL 对象之间连接的方法原型。TactionLink 方法确保行为是支持客户行为的适当类型。这些方法被其子类所重设, 使行为事件和客户属性之间能够保持适当的连接。在 TActionLink 对象中未生效的方法原型也被引入, 通过一个 Action 连接, 从而作为设置关于客户属性行为的接口界面。子 Action 连接类对这些方法进行重设, 确保了客户类型具有适当的行为。

如果想将客户而不是控件或菜单项与 Actions 进行连接, 可以将 TActionLink 对象看作是一个基类。TActionLink 对象为支持下述操作的方法引入了接口界面, 这些操作中

的大多数与同名的属性相连接:

Caption, Checked, Enabled, HelpContext, Hint, ImageIndex, ShortCut, Visible。

方法列表

IsCaptionLinked	确定标题是否与客户相连接
IsCheckedLinked	确定复选状态行为是否与客户连接
IsEnabledLinked	确定激活状态行为是否与客户连接
IsHelpContextLinked	确定上下文帮助行为是否与客户连接
IsHintLinked	确定提示行为是否与客户连接
IsImageIndexLinked	确定图像索引行为是否与客户连接
IsShortCutLinked	确定快捷方法行为是否与客户连接
IsVisibleLinked	确定 Visible 状态行为是否与客户连接
SetCaption	为客户设置标题行为引入接口界面
SetChecked	为客户设置复选状态行为引入接口界面
SetEnabled	为客户激活状态行为引入接口界面
SetHelpContext	为客户设置帮助上下文行为引入接口界面
SetHint	为客户设置提示行为引入接口界面
SetImageIndex	为客户设置图像索引
SetShortCut	为客户设置快捷方法行为引入接口界面
SetVisible	为客户设置对象是否可见

方 法

TActionLink.IsCaptionLinked

IsCaptionLinked 确定标题是否与客户相连接。

function IsCaptionLinked: Boolean; virtual;

IsCaptionLinked 函数为客户确保了支持适当事件的行为类型。其子类重设 IsCaptionLinked 函数, 以确定标题行为是否与客户属性相连接。如果客户请求行为控制着事件, 则 IsCaptionLinked 函数返回值为 True, 否则返回值为 False。

TActionLink.IsCheckedLinked

IsCheckedLinked 确定复选状态行为是否与客户连接。

function IsCheckedLinked: Boolean; virtual;

IsCheckedLinked 函数为客户确保了支持适当事件的行为类型。其子类重设 IsCheckedLinked 函数, 以确定复选状态行为是否与客户属性相连接。如果客户请求行为控制着事件, 则 IsCheckedLinked 函数返回值为 True, 否则返回值为 False。

TActionLink.IsEnabledLinked

IsEnabledLinked 确定激活状态行为是否与客户连接。

function IsEnabledLinked: Boolean; virtual;

IsEnabledLinked 函数为客户确保了支持适当事件的行为类型。其子类重设 IsEnabledLinked 函数, 以确定复选状

态行为是否与适当的客户属性相连接。如果客户请求行为控制着事件,则 `IsEnabledLinked` 函数返回值为 `True`, 否则返回值为 `False`。

TActionLink.IsHelpContextLinked

`IsHelpContextLinked` 确定上下文帮助行为是否与客户连接。

function IsHelpContextLinked: Boolean; virtual;

`IsHelpContextLinked` 函数为客户确保了支持适当事件的行为类型。其子类重设 `IsHelpContextLinked` 函数, 以确定帮助上下文行为是否与适当的客户属性相连接。如果客户请求行为控制着事件, 则 `IsHelpContextLinked` 函数返回值为 `True`, 否则返回值为 `False`。

TActionLink.IsHintLinked

`IsHintLinked` 确定提示行为是否与客户连接。

function IsHintLinked: Boolean; virtual;

`IsHintLinked` 函数为客户确保了支持适当事件的行为类型。其子类重设 `IsHintLinked` 函数, 以确定提示行为是否与适当的客户属性相连接。如果客户请求行为控制着事件, 则 `IsHintLinked` 函数返回值为 `True`, 否则返回值为 `False`。

TActionLink.IsImageIndexLinked

`IsImageIndexLinked` 确定图像索引行为是否与客户连接。

function IsImageIndexLinked: Boolean; virtual;

`IsImageIndexLinked` 函数为客户确保了支持适当事件的行为类型。其子类重设 `IsImageIndexLinked` 函数, 以确定图像索引行为是否与适当的客户属性相连接。如果客户请求行为控制着事件, 则 `IsImageIndexLinked` 函数返回值为 `True`, 否则返回值为 `False`。

TActionLink.IsShortCutLinked

`IsShortCutLinked` 确定快捷方法行为是否与客户连接。

function IsShortCutLinked: Boolean; virtual;

`IsShortCutLinked` 函数为客户确保了支持适当事件的行为类型。其子类重设 `IsShortCutLinked` 函数, 以确定快捷方法行为是否与适当的客户属性相连接。如果客户请求行为控制着事件, 则 `IsShortCutLinked` 函数返回值为 `True`, 否则返回值为 `False`。

TActionLink.IsVisibleLinked

`IsVisibleLinked` 确定 `Visible` 状态行为是否与客户连接。

function IsVisibleLinked: Boolean; virtual;

`IsVisibleLinked` 函数为客户确保了支持适当事件的行为类型。其子类重设 `IsVisibleLinked` 函数, 以确定 `Visible` 状态行为是否与适当的客户属性相连接。如果客户请求行为控制着事件, 则 `IsVisibleLinked` 函数返回值为 `True`, 否则返回值为 `False`。

TActionLink.SetCaption

`SetCaption` 为客户设置标题行为引入了一个接口界面。

procedure SetCaption(const Value: string); virtual;

`SetCaption` 过程是一个方法原型, 它通过行为连接为客户设置标题行为引入了一个接口界面。Value 参数指定了将被设置的标题的字符串。子行为连接类重设 `SetCaption` 过程, 确保它们的客户类型具有相应的行为。

TActionLink.SetChecked

`SetChecked` 为客户设置复选状态行为引入了一个接口界面。

procedure SetChecked(Value: Boolean); virtual;

`SetChecked` 过程是一个方法原型, 它通过行为连接为客户设置复选状态行为引入了一个接口界面。Value 参数表示复选状态是否被设置。子行为连接类重设 `SetChecked` 过程, 确保它们的客户类型具有相应的行为。

TActionLink.SetEnabled

`SetEnabled` 为客户激活状态行为引入了一个接口界面。

procedure SetEnabled(Value: Boolean); virtual;

`SetEnabled` 过程是一个方法原型, 它通过行为连接为客户设置激活状态行为引入了一个接口界面。Value 参数表示激活状态是否被设置。子行为连接类重设 `SetEnabled` 过程, 确保它们的客户类型具有相应的行为。

TActionLink.SetHelpContext

`SetHelpContext` 为客户设置帮助上下文行为引入了一个接口界面。

procedure SetHelpContext(Value: THelpContext); virtual;

`SetHelpContext` 过程是一个方法原型, 它通过行为连接为客户设置帮助上下文行为引入了一个接口界面。Value 参数指定了将被设置的帮助上下文的数值。子行为连接类重设 `SetHelpContext` 过程, 确保它们的客户类型具有相应的行为。

TActionLink.SetHint

`SetHint` 为客户设置提示行为引入了一个接口界面。

procedure SetHint(const Value: string); virtual;

`SetHint` 过程是一个方法原型, 它通过行为连接为客户设置提示行为引入了一个接口界面。Value 参数指定了将被设置的提示字符串。子行为连接类重设 `SetHint` 过程, 确保它们的客户类型具有相应的行为。

TActionLink.SetImageIndex

`SetImageIndex` 为客户设置行为引入了一个接口界面。

procedure SetImageIndex(Value: Integer); virtual;

`SetImageIndex` 是一个方法原型, 它通过行为连接为客户设置图像索引行为引入了一个接口界面。Value 参数指定了将被设置的图像的索引。子行为连接类重设 `SetImageIndex`, 确保它们的客户类型具有相应的行为。

TActionLink.SetShortCut

`SetShortCut` 为客户设置快捷方法行为引入了一个接口界面。

procedure SetShortCut(Value: TShortCut); virtual;

`SetShortCut` 是一个方法原型, 它通过行为连接为客户

设置快捷方法行为引入了一个接口界面。Value 参数指定了将被设置的快捷方法的 TShortCut。子行为连接类重设 SetShortCut 过程,确保它们的客户类型具有相应的行为。

TActionLink. SetVisible

SetVisible 为客户设置行为引入了一个接口界面。

procedure SetVisible(Value: Boolean); virtual;

SetVisible 是一个方法原型,它通过行为连接为客户设置可视状态行为引入了一个接口界面。Value 参数表示可视状态是否被设置。子行为连接类重设 SetVisible 过程,确保它们的客户类型具有相应的行为。

TActionList 单元 ActnList

类关系 TObject→TPersistent→TComponent→TCustomActionList

TActionList 是创建组件的通用类,它提供了使用诸如菜单选项和按钮这样的组件和控件行为的列表。

TActionList 可以执行在 TCustomActionList 中采用的一般行为,但没有添加新行为。TActionList 采用了从 TCustomActionList 继承的一些成员。

行为列表由行为和行为连接联合使用,以聚集对用户命令(行为)的响应。行为列表组件是与行为一起运转的用户接口。从组件选项板的标准页中,可以增加行为列表组件到窗体或数据模块中。双击行为列表,可以显示行为列表编辑器。在行为列表编辑器中可以增加、删除以及重新排列行为。

属性列表

ComObject	指定支持 COM 接口的组件实现的接口引用
ComponentCount	隶属于 Components 数组属性列表中被列出组件的数量
ComponentIndex	隶属于 Components 数组属性列表的组件的位置
Components	提供对隶属于某组件的所有组件进行变址访问
ComponentState	撤消确定的行为时描述组件的当前状态
ComponentStyle	管理组件的行为
DesignInfo	窗体设计器使用的信息
Name	指定引用代码的组件名称
Owner	拥有该组件的组件
Tag	存储整型数值为组件的要素
VCLComObject	支持 COM 组件内部使用的信息

TContainedAction 单元 ActnList

类关系 TObject→TPersistent→TComponent→TBasicAction

TContainedAction 是能够按类分组并在行为列表显示的行为的基类。

TContainedAction 引入了能够使行为在一个行为列表

上显示的属性和方法。TContainedAction 对象的属性包含了行为与行为列表之间相互关系的信息。TContainedAction 对象的方法重设了该基类的那些方法,以包括行为的行为列表。在行为列表组件上显示的 TContainedAction 类型的对象,与行为类型的一个类型关联。

属性列表

ActionList	包含行为的行为列表
Category	包含在行为列表上显示行为的类型
Index	保存行为列表中行为的索引

方法列表

Destroy	删除容纳行为对象的实例
Execute	是否调用 OnExecute 事件处理程序
GetParentComponent	返回包含行为的列表
HasParent	行为是否包含在行为列表中
Update	是否调用 OnUpdate 事件处理程序

属性

TContainedAction. ActionList

ActionList 属性指包含行为的行为列表。

property ActionList: TCustomActionList;

ActionList 属性表示哪一个行为列表组件包含行为。

只有包含的行为(TContainedAction 对象的实例或者其子类)能够在一个行为列表上显示。

参见 TContainedAction. Category。

TContainedAction. Category

Category 属性包含在行为列表上显示行为的类型。

property Category: string;

Category 属性是行为一个部分的组或者种类。Category 属性作为行为列表编辑器上一列的一个标题显示。行为按共性(通常是行为或者机能)被无条件地分组。在设计阶段,能够修改或者设置从行为列表选取的一个行为的一个 Category 属性。

TContainedAction. Index

Index 属性保存行为列表中行为的索引。

property Index: Integer;

Index 属性包含表示行为列表中行为位置的下标数值。通过使用上下箭头或工具栏或行为列表编辑器的快捷菜单,可以对行为列表中一个行为的位置进行修改。

方法

TContainedAction. Destroy

删除容纳行为对象的实例。

destructor Destroy; override;

在应用程序中不要直接调用 Destroy,而应调用 Free。Free 检查到行为对象还没有被释放后,才调用 Destroy。如果行为包含在一个行为列表中,则 Destroy 从该列表中删除行为,然后再调用继承的 Destroy。

TContainedAction.Execute

表示是否调用一个 OnExecute 事件处理程序。

function Execute: Boolean; override;

Execute 函数表示是否调用一个 OnExecute 事件处理程序。如果设定了行为的事件处理程序,则调用该处理程序,否则调用行为列表的事件处理程序。如果行为和行为列表都没有为 OnExecute 事件设定处理程序,则 Execute 函数委派应用程序为该事件执行一个关联的处理程序。如果一个事件处理程序被调用,则 Execute 函数返回 True。

如果在行为对象、行为列表或者应用程序中,都没有找到 OnExecute 事件处理程序,则 Execute 函数返回 False 并且使那个特定行为不能使用。

TContainedAction.GetParentComponent

返回包含行为的行为列表。

function GetParentComponent: TComponent; override;

GetParentComponent 函数被装入并保存 VCL 组件的流系统调用。它可以确保对象与其父对象一起被装入和保存。TContainedAction 对象重设继承的方法,以使其关联的行为列表等同于其父对象。

参见 TContainedAction.ActionList。

TContainedAction.HasParent

表示行为是否包含在一个行为列表中。

function HasParent: Boolean; override;

HasParent 函数被装入并保存 VCL 组件的流系统调用。它可以确保对象与其父对象一起被装入和保存。TContainedAction 对象重设继承的方法,如果行为被包含在一个行为列表中,则 HasParent 函数返回 True。

参见 TContainedAction.GetParentComponent, TContainedAction.ActionList。

TContainedAction.Update

表示是否调用 OnUpdate 事件处理程序。

function Update: Boolean; override;

Update 函数表示是否调用一个 OnUpdate 事件处理程序。如果设定了行为的事件处理程序,则调用该处理程序,否则调用行为列表的事件处理程序。如果行为和行为列表都没有为 OnUpdate 事件设定处理程序,则 Update 函数委派应用程序为该事件执行一个关联的处理程序。如果一个事件处理程序被调用,则 Update 函数返回 True。

如果在行为对象、行为列表或者应用程序中,都没有找到 OnUpdate 事件处理程序,则 Update 函数返回 False 并且使那个特定行为不能使用。

TCustomAction

单元 Actnlist

类关系 TObject→TPersistent→TComponent→TBasicAction→TContainedAction

TCustomAction 是菜单项目和控件使用行为的基类。

TCustomAction 引入了支持与行为对象相链接的菜单

项目和控件的属性和方法。当派生公有关联控件具体属性的行为时,使用 TCustomAction 作为一个基类。VCL 提供了一个通用的子类,即公开 TCustomAction 所有属性,以及诸如数据集或文本控件这样的组件指定的行为类。

行为组件用于集中响应用户的命令,以便能够同步进行,如可用状态与一个加速按钮的标题和一个菜单项。行为对象有一个通过一个行为链接与之通讯的目标或客户。在设计阶段,能够以一个行为列表的行为列表编辑器内的行为作为工作对象。行为列表是行为的一个容器,它能够按照类型进行分组。

在 TCustomAction 内,或通过 TAction 支持的组件和控件的属性和事件有:

Caption, Checked, Enabled, HelpContext, Hint, ImageIndex, ShortCut, Visible, OnHint, OnUpdate, OnExecute。

属性列表

Caption	包含行为的 Caption 属性
Checked	包含行为的 Checked 属性
DisableIfNoHandler	表示没有找到句柄时关联组件是否无效
Enabled	包含行为的 Enabled 属性
HelpContext	包含行为的 HelpContext 属性
Hint	包含行为的 Hint 属性
ImageIndex	包含行为的 ImageIndex 属性
Name	包含作为其他组件引用的行为名称
ShortCut	包含行为的 ShortCut 属性
Visible	包含行为的 Visible 属性

方法列表

AssignTo	将对象的属性复制到目的对象
Create	实例化并初始化 TCustomAction 对象
DefineProperties	为储存行为的未公开数据指定方法
Destroy	释放 TCustomAction 的实例
DoHint	产生 OnHint 事件
Execute	更新行为并在激活时执行
SetName	设置 Name 属性数值

事件列表

OnHint	鼠标在关联控件或菜单项上停顿时事件发生
--------	---------------------

属 性

TCustomAction.Caption

包含行为的 Caption 属性。

property Caption: string;

当行为标题被设置时, Caption 属性包含行为标题的字符串。该字符串表示关联控件和菜单项的 Caption 属性。

参见 TControl.Caption, TMenuItem.Caption。

TCustomAction.Checked

包含行为的 Checked 属性。

property Checked: Boolean;

Checked 属性指定行为的复选状态。该数值表示关联控件和菜单项的 Checked 属性数值。

参见 TButtonControl.Checked, TCheckListBox.Checked, TDateTimePicker.Checked, TMenuItem.Checked。

TCustomAction.DisableIfNoHandler

表示没有找到处理程序时关联组件是否无效。

property DisableIfNoHandler: Boolean;

如果没有找到行为处理程序时, DisableIfNoHandler 属性指定关联组件的 Enabled 属性是否设置为 False。那些组件检查其在 UpdateAction 方法中的一个行为处理程序。

参见 TBasicAction.HandlesTarget, TComponent.UpdateAction, TCustomAction.Enabled。

TCustomAction.Enabled

包含行为的 Enabled 属性。

property Enabled: Boolean;

Checked 属性指定行为的可用状态。该数值表示关联控件和菜单项的 Enabled 属性数值。

参见 TControl.Enabled, TMenuItem.Enabled。

TCustomAction.HelpContext

包含行为的 HelpContext 属性。

type HelpContext = -MaxLongint..MaxLongint;

property HelpContext: THelpContext;

当行为的帮助上下文被设置时, HelpContext 属性包含该数值。该数值表示关联控件和菜单项的 HelpContext 属性。

参见 TMenuItem.HelpContext, TWinControl.HelpContext。

TCustomAction.Hint

包含行为的 Hint 属性。

property Hint: string;

当行为提示被设置时, Hint 属性包含行为提示字符串。该数值表示关联控件和菜单项的 Hint 属性。

参见 TControl.Hint, TMenuItem.Hint。

TCustomAction.ImageIndex

包含行为的 ImageIndex 属性。

property ImageIndex: Integer;

ImageIndex 属性是包含行为的行为列表的 Image 属性的一个下标。它识别使用该行为的关联控件和菜单项的一个图像。

参见 TCustomActionList.Images, TMenuItem.ImageIndex, TToolButton.ImageIndex。

TCustomAction.Name

包含作为其他组件引用的行为名称。

property Name: TComponentName;

当以代码引用行为时, 使用 Name 属性。如果 Name 属性设置为与 Caption 属性相同的字符串, 则两个属性变为一致; 即对 Name 属性数值修改的结果也更新 Caption 属性。

参见 TCustomAction.Caption。

TCustomAction.ShortCut

包含行为的 ShortCut 属性。

property ShortCut: TShortCut;

当行为快捷键被设置时, ShortCut 属性包含该数值。该数值表示关联菜单项的 ShortCut 属性数值。

参见 TMenuItem.ShortCut。

TCustomAction.Visible

包含行为的 Visible 属性。

property Visible: Boolean;

Visible 属性指定行为的可视状态。该数值表示关联控件和菜单项的 Visible 属性。

参见 TControl.Visible, TMenuItem.Visible。

方法

TCustomAction.AssignTo

将一个对象的属性复制到一个目的对象。

procedure AssignTo(Dest: TPersistent); override;

AssignTo 被重设, 以便存在的 TCustomAction 类型的对象能够被复制到相同类型的其他行为对象。AssignTo 确保属性数值的正确设置。

TCustomAction.Create

实例化并初始化一个 TCustomAction 对象。

constructor Create(AOwner: TComponent); override;

应用程序不需直接调用 Create。当在行为列表编辑器内将一个行为与一个客户相连接时, 行为对象被自动创建。如果要在运行时创建一个行为对象, 则应调用 Create 并将一个 TActionList 组件赋值至其行为列表属性。Create 调用继承的 Create, 然后初始化下列属性:

- DisableIfNoHandler 初始化为 True。
- Enabled 属性初始化为 True。
- ImageIndex 属性初始化为 -1。
- Visible 属性初始化为 True。

参见 TContainedAction.ActionList, TCustomAction.DisableIfNoHandler, TCustomAction.Enabled, TCustomAction.ImageIndex, TCustomAction.Visible。

TCustomAction.DefineProperties

在如一个窗体文件的流中, 为储存行为的未公开数据指定方法。

procedure DefineProperties(Filer: TFiler); override;

DefineProperties 作为流系统的一部分被自动调用, 不要直接调用 DefineProperties。TCustomAction 重设 DefineProperties, 以考虑行为与其行为列表之间的连接。

TCustomAction.Destroy

释放 TCustomAction 的一个实例。

```
procedure DefineProperties(Filer: TFile); override;
```

在应用程序中不要直接调用 Destroy, 而应调用 Free。

Free 检查到行为对象还没有被释放后, 才调用 Destroy。如果行为包含在一个行为列表中, 则 Destroy 从该列表中删除行为。

参见 TObject.Free。

TCustomAction.DoHint

产生一个 OnHint 事件。

```
function DoHint(var HintStr: string): Boolean;
```

DoHint 检查一个 OnHint 事件处理程序并向其传递 HintStr 参数。如果未设定事件处理程序, 则 DoHint 返回 True。

TCustomAction.Execute

更新行为, 然后只有在行为被激活时才执行。

```
function TCustomAction.Execute: Boolean; override;
```

Execute 确保行为被更新, 并且只有在 Enabled 属性为 True 时, 才调用继承的 Execute 方法。

参见 TCustomAction.Enabled, TWinControlActionLink。

TCustomAction.SetName

设置 Name 属性数值。

```
procedure SetName(const Value: TComponentName);
```

override;

SetName 被用作 Name 属性的输入部分。它调用 ChangeName 方法, 该方法执行实际的名称修改。ChangeName 方法不是虚拟的。SetName 被重设, 以修改 Name 属性行为, 以便它与 Caption 属性相同时也修改 Caption 属性。

事件

TCustomAction.OnHint

当鼠标在能够显示一个帮助提示的关联控件或菜单项上停顿时, OnHint 事件发生。

THintEvent procedure (var HintStr: string; var CanShow: Boolean) of object;

```
property OnHint: THintEvent;
```

当用户使鼠标在一个关联控件或菜单项上停顿时, 使用 OnHint 事件可以重设显示的缺省提示。在事件处理程序中, 当在一个弹出式窗口中显示提示字符串时, 使用 HintStr 重设 Hint 属性。设置 CanShow 可以表示是否一直显示提示。

参见 TCustomAction.Hint。

TCustomActionList

单元 Actnlist

类关系 TObject→TPersistent→TComponent

TCustomActionList 是向行为提供 IDE 用户接口的行为列表对象的基类。

TCustomActionList 是一个组件, 它允许在设计阶段以

行为作为工作对象。行为列表编辑器是以行为作为工作对象的用户接口。从行为列表编辑器中, 能够增加、删除、重新分组和修改行为。TActionList 保存能够按照种类分组的—个行为列表。在一个行为列表中的行为是 TContainedAction 或子类型。

属性列表

ActionCount	在行为列表中的行为数量
Actions	行为列表对象中行为的一个被索引列表
Images	包含行为列表中行为可用图像的列表

方法列表

Change	调用 OnChange 事件处理程序的方法
Create	实例化并初始化一个 TCustomActionList 对象
Destroy	删除一个行为列表对象的实例
ExecuteAction	当列表中的行为对象执行一个行为时产生 ExecuteAction
GetChildren	为行为列表中的每一子行为调用指定的方法
IsShortCut	表示输入键组合是否为列表中一个行为的快捷键
Notification	向所有拥有组件发送通知消息
SetChildOrder	修改子行为组件在行为列表中的顺序
UpdateAction	通知行为列表来自一个行为的更新

事件列表

OnChange	当在行为列表中出现一个变化时, 产生 OnChange 事件
OnExecute	当行为引发 Execute 事件的触发器时, 产生 OnExecute 事件
OnUpdate	当行为列表由一个行为通知更新时, 产生 OnUpdate 事件

参见 TActionList, TContainedAction。

属性

TCustomActionList.ActionCount

表示在行为列表中的行为数量。

```
property ActionCount: Integer;
```

ActionCount 属性表示在行为列表中包含多少个行为。

TCustomActionList.Actions

表示行为列表对象中行为的一个索引列表。

```
property Actions[Index: Integer]: TContainedAction;
```

Actions 属性是一个索引的属性, 它包含行为列表组件中行为的一个列表。

TCustomActionList.Images

包含行为列表中行为可用图像的列表。

```
property Images: TCustomImageList;
```

Images 属性是一个位图列表, 该位图能够在行为列表编辑器内任何行为的左边显示。

方 法

TCustomActionList.Change

是调用 OnChange 事件处理程序的方法。

procedure Change; **virtual**;

当在行为列表内出现一个变化, 如, 对 Categories 列表进行一个修改时, 调用 Change。如果已经设定了一个 OnChange 事件处理程序, 则 Change 援引该处理程序。

TCustomActionList.Create

实例化并初始化一个 TCustomActionList 对象。

constructor Create (AOwner: TComponent); **override**;

行为列表组件时, 自动创建 TBasicActionLink 实例。AOwner 参数是行为列表的所有者, 典型地为窗体。

Create 调用继承的 Create 方法, 并且通过创建其能够处理行为和图像列表的子对象来初始化对象。

TCustomActionList.Destroy

删除一个行为列表对象的实例。

destructor Destroy; **override**;

不需为一个行为链接调用 Destroy。当创建行为列表对象时, 向其构造函数中传递的所有者, 将自动删除该链接。

TCustomActionList.ExecuteAction

当列表中的一个行为对象执行一个行为时发生。

function ExecuteAction (Action: TBasicAction): Boolean; **override**;

由 Action 参数表示的一个行为的 Execute 方法调用 ExecuteAction。ExecuteAction 调用行为列表的 OnExecute 事件处理程序。

TCustomActionList.GetChildren

为列表中的每一个子行为调用指定的方法。

type TGetChildProc **procedure** (Child: TComponent) of object;

procedure GetChildren (Proc: TGetChildProc; Root: TComponent); **override**;

GetChildren 由 Delphi 流系统使用。TCustomActionList 重设 GetChildren, 返回其子行为组件。对于每一个子项目, GetChildren 将调用在 Proc 参数中传递的方法。子行为必须按其子行为顺序返回。

TCustomActionList.IsShortCut

表示一个输入键组合是否为列表中一个行为的一个快捷键。

function IsShortCut (var Message: TWMKey): Boolean;

如果由 Message 参数指定的快捷键是列表中一个行为的一个快捷键组合, 则 IsShortCut 返回 True。该函数为内部使用, 不需要直接调用它。

TCustomActionList.Notification

向所有拥有组件发送通知消息。

procedure Notification (AComponent: TComponent; Operation: TOperation); **override**;

Notification 通知组件: 由 AComponent 参数指定的组件将被插入或删除。插入或删除由 Operation 参数指定。缺省时, 组件向其拥有组件传递通知。如果需要, 一个组件能够影响到一个组件将被插入或删除的通知。特别地, 如果一个组件具有包含对其他组件引用的对象字段或属性, 则它将检查删除组件的通知, 且当需要时使那些引用无效。

TCustomActionList 重设 Notification, 以更新组件, 包括从列表中删除图像。

TCustomActionList.SetChildOrder

修改子行为组件在行为列表中的顺序。

procedure SetChildOrder (Component: TComponent; Order: Integer); **override**;

SetChildOrder 修改子行为组件在 GetChildren 返回的子列表中的显示顺序。SetChildOrder 由 Delphi 流系统使用。不需要直接调用该方法。

TCustomActionList.UpdateAction

通知行为列表来自一个行为的一个更新。

function UpdateAction (Action: TBasicAction): Boolean; **override**;

当在需要行为列表通知的行为中出现一个变化时, 调用 UpdateAction。Action 参数是被更新的行为。UpdateAction 调用 OnUpdate 事件处理程序, 并且当设定了一个 OnUpdate 事件处理程序时返回 True, 否则返回 False。

事 件

TCustomActionList.OnChange

当在行为列表中出现一个变化时, OnChange 事件发生。

property OnChange: TNotifyEvent;

当在行为列表中出现一个变化时, 使用 OnChange 事件可以写一个处理一个行为的事件处理程序。OnChange 事件也通知行为, 以便调用每一个行为的 OnChange 事件处理程序。例如, 引发一个 OnChange 事件对行为列表的修改为对一个 Category 的一个修改。

TCustomActionList.OnExecute

当一个行为引发一个 Execute 事件的触发器时, OnExecute 事件发生。

ActionEvent procedure (Action: TBasicAction; var Handled: Boolean) of object; **property** OnExecute: TActionEvent;

写一个响应一个行为引发一个 Execute 事件时的 OnExecute 事件处理程序。Action 参数为触发事件的行为。Handled 参数表示是否为 OnExecute 事件设定了一个句柄。

TCustomActionList.OnUpdate

当行为列表由一个行为通知更新时, OnUpdate 事件发