

Kontakt

PrintFormList



PrintFormList
Liste,
Siehe Eigenschaften,
PrintForm,
Blog

Das PrintFormList-Objekt dient der Erstellung von Drucklisten. Es handelt sich bei dem Objekt um einen Formular-Typ, analog zu einem PrintForm-Objekt, jedoch mit spezieller Funktionalität.

Dem PrintFormList-Objekt können in der ersten Ebene nur Zeilenobjekte (PrtLine-Objekte) hinzugefügt werden. Die PrintFormList richtet die Zeilenobjekte so aus, dass eine Zeilenstruktur (also eine Liste) entsteht.

Das Zeilenobjekt kann für optische Zwecke oder zur Erzeugung eines Zeilenabstandes mit einem Rahmenobjekt (PrtLineBorder) versehen werden. Die Zeilenobjekte können ebenfalls ausgeblendet werden, wenn kein Inhalt dargestellt wird (Eigenschaft SkipPrint).

Die Breite aller Zeilenobjekte definiert sich über die Eigenschaft Width des PrintFormList-Objekts. Außerdem haben alle Zeilenobjekte einen Abstand vom linken Rand der PrintFormList (MarginLeft) und einen Abstand vom oberen Rand (MarginTop).

PrtLine

 **PrtLine**

[Liste](#),

Siehe [Eigenschaften](#),

[PrintFormList](#)

Das Zeilenobjekt kann aus beliebigen druckbaren Objekten bestehen. Jedoch kann ein Zeilenobjekt keine weiteren Zeilenobjekte aufnehmen. Die Breite des Zeilenobjektes ist fest (siehe [PrintFormList](#)). Die Höhe ergibt sich jedoch aus dem umschließenden Rechteck aller untergeordneten und sichtbaren Objekte.

Bei "sichtbar" sind hier unterschiedliche Kriterien möglich. Ist ein PrtLine-Objekt nach obiger Definition nicht sichtbar, wird es bei der Druckaufbereitung (durch den Befehl [PrtAdd\(\)](#)) ignoriert (d. h. es gelangt nicht in den Druckjob).

Die Kriterien für die Sichtbarkeit des PrtLine-Objektes definiert die Eigenschaft [SkipPrint](#).

Kontakt

PrtLineBorder

PrtLineBorder

Liste,

Siehe Eigenschaften,

PrintFormList

Mit diesem Objekt kann eine Zeile (PrtLine) eines PrintFormList-Objektes hervorgehoben werden.

Jedem PrtLine-Objekt kann maximal ein Rahmenobjekt zugeordnet werden. Die Sichtbarkeit des Rahmenobjektes bei der Druckaufbereitung hängt von der Sichtbarkeit des PrtLine-Objektes ab. Ist dieses unsichtbar, ist auch das Rahmenobjekt unsichtbar. Ist das PrtLine-Objekt sichtbar, entscheidet die Eigenschaft VisiblePrint über die Sichtbarkeit des Rahmenobjektes.

Die Eigenschaft StylePen bestimmt, ob eine Linie und welche Linie zur Umrandung gezeichnet wird. Die Farbe der Linie ergibt sich aus den Einstellungen in den Eigenschaften ColFg, ColFillEven und ColFillOdd.

Die Dicke der Linie ergibt sich aus der Eintragung in der Eigenschaft WidthPen.

Das PrtLineBorder-Objekt kann nur eingefügt werden, wenn ein PrtLine-Objekt selektiert ist. Der Rahmen um dieses Objekt gezeichnet. Der Rahmen passt sich immer an die Größe des Objektes an.

Kontakt

PrintDoc



[PrintDoc](#)

[Liste](#),

Siehe [Eigenschaften](#),

[PrtInfo\(\)](#),

[Blog](#)

Das PrintDoc-Objekt (**Druckdokument**) definiert eine druckbare Vorlage für eine oder mehrere Seiten mit beliebigem Inhalt. Das Objekt sollte immer dann verwendet werden, wenn alle Informationen innerhalb der Seite auf festen Positionen zu finden sind. Zum Beispiel in Briefvorlagen oder Formularen (Überweisungsträger usw.).

Das Druckdokument kann nur **Seiten**-Objekte aufnehmen. Es enthält Eigenschaften, die das Papierformat ([PageFormat](#)) und die Ausrichtung ([Orientation](#)) der Seiten bestimmen.

Die Seitenobjekte selbst enthalten wiederum Druckobjekte, die den Seiteninhalt bestimmen. Wird ein PrintDoc-Objekt im Designer erstellt, sind in der Objektpalette nur die Registerreiter "Druck" und "Seite" vorhanden.

Das PrintDoc-Objekt kann bis zu zehn Seiten-Objekte aufnehmen. Der Druckjob selbst kann mehr Seiten umfassen, da einzelne Seiten aus dem PrintDoc-Objekt mehrfach (mit unterschiedlichen Inhalten) gedruckt werden können.

In einem Druckjob ([PrtJobOpen\(\)](#)) kann das PrintDoc-Objekt über die Funktion [PrtInfo\(\)](#) mit der Option [PrtDoc](#) ermittelt werden.

Kontakt

PrintDocRecord



[PrintDocRecord](#)

[Liste](#),

Siehe [Eigenschaften](#),

[Blog](#)

Das PrintDocRecord-Objekt (Tabellen-Dokument) ist ein spezielles [PrintDoc](#)-Objekt. Es kann lediglich eine Seite beinhalten. Wie beim PrintDoc-Objekt, definiert das Objekt ein Papierformat ([PageFormat](#)), sowie die Ausrichtung ([Orientation](#)) der Seite.

Die im Tabellen-Dokument enthaltene Seite kann wiederum nur ein Tabellenobjekt aufnehmen. Durch diese Kombination von Druckdokument, Seite und Tabelle ist die einfache Erstellung von Drucklisten möglich.

Ein PrintDocRecord-Objekt kann bei der Erstellung im Designer mit dem Tabellen-Assistent konfiguriert werden. Wurde das Objekt erstellt, sind in der Objekt-Palette die Registerreiter "Druck", "Seite" und "Tabelle" vorhanden.

Die angegebene Anzahl von Spalten werden beim Erzeugen gleichmäßig über die gesamte Seitenbreite angeordnet. Die Breite der Spalten kann entweder mit der Maus oder über die Eigenschaft [Width](#) verändert werden. Wird beim Ändern der Breite

einer Spalte mit der Maus die -Taste gedrückt, werden alle übereinander liegenden Spalten mit verändert.

In der Tabelle werden alle Datensätze einer Datei in der Reihenfolge des angegebenen Schlüssels ausgegeben. Die Anzahl der Datensätze kann über eine Selektion ([DbSelection](#)) oder Filter ([DbFilter](#)) zur Laufzeit eingeschränkt werden.

Druck-Objekte

Siehe Alle **Druckformular-Objekte**

PrtText



PrtTblCellText



PrtRtf



PrtPicture



PrtMetaPicture



PrtPdf



PrtBorder



PrtGanttGraph



PrtBarcode



PrtDivider



PrtGroupbox



Kontakt

PrText



PrText

Siehe [Liste](#),

Eigenschaften

Mit dem PrText-Objekt ist die Ausgabe von Text für die Druckausgabe möglich.

Bei der Bearbeitung im Designer kann durch einen Doppelklick auf die entsprechende Zeile der Eigenschaftsliste ein Text-Eingabe-Dialog gestartet werden. Dieser vereinfacht die Eingabe von mehrzeiligen Texten.

Mit der Eigenschaft [AutoSize](#) wird festgelegt, ob sich die Objektgröße an die Textgröße anpasst oder nicht.

Die Eigenschaft [Caption](#) definiert den auszugebenden Text. Soll der Text dagegen einem Datenbankfeld entnommen werden, ist die Eigenschaft [DbFieldName](#) zu setzen. Bei der Ausgabe werden führende Leerzeichen nicht gedruckt. Soll Text eingerückt werden, kann dazu die Eigenschaft [AreaLeft](#) oder geschützte Leerzeichen [StrChar\(255\)](#) verwendet werden. Je nach Datentyp des Datenbankfeldes sind weitere Textformatierungen möglich. Handelt es sich z. B. um ein Fließkomma-Feld, so kann der Text über die Eigenschaften [FmtFloatFlags](#) und [FmtPostComma](#) formatiert werden.

Ausschließlich bei Fließkomma-Feldern sind auch Eigenschaften der Gruppe [Function](#) vorhanden.

Die Ausgabe von berechneten Funktionswerten ist möglich, wenn die Eigenschaft [StyleCaption](#) entsprechend gesetzt ist.

Eine Ausrichtung in horizontaler und vertikaler Richtung kann über die Eigenschaften [Justify](#) und [JustifyVert](#) erfolgen.



Durch die Angabe von [WinColTransparent](#) als Hintergrundfarbe ([ColBkg](#)) kann das Objekt über ein anderes Objekt gedruckt werden.

Kontakt

PrtRtf

PrtRtf

Liste,
Eigenschaften,
Text und
Daten

Siehe **mischen, Text**
suchen und
erstellen,

In diesem Objekt kann Text im Rich Text Format (RTF) ausgedruckt werden. Der formatierte Text kann aus unterschiedlichen Quellen stammen. Die Quelle wird in der Eigenschaft StreamSource angegeben. Je nach gewählter Quelle wird der Inhalt in der Eigenschaft Caption, DbFieldName oder FileName angegeben.

Der RTF-Text enthält bereits alle notwendigen Formatierungsanweisungen. Der Text wird nur noch auf die Breite des PrtRtf-Objektes angepasst. Sollen unter dem Objekt liegende Objekte angezeigt werden (zum Beispiel Grafiken), kann die Hintergrundfarbe (ColBkg) auf WinColTransparent gesetzt werden.

Über die Eigenschaft range kann der Bereich angegeben werden, der in dem PrtRtf-Objekt gedruckt werden soll.

Soll ein Text mit Daten gemischt gedruckt werden, muss in der Eigenschaft PrtRtfFlags die Ausprägung PrtRtfMix gesetzt sein.

Über die Funktion PrtRtfSearch() kann eine Textsuche- und Ersetzung vorgenommen werden.



Bei dem Objekt kann in der Eigenschaft PrtDevice ein PrintDevice-Objekt angegeben werden. In der Druckvorschau und beim Drucken können dann die Besonderheiten der verschiedenen Windows-Druckertreiber berücksichtigt werden.

Kontakt

Text und Daten mischen

Texte mit Datensätzen verbinden

Ein bestehender Text oder RTF-Text kann mit dem Inhalt von Datensatzpuffern bzw. von globalen Variablen verbunden werden. Dazu müssen im Text entsprechende Platzhalter eingetragen werden. Die Platzhalter werden dabei in ein Markierungszeichen geklammert, um sie vom normalen Text unterscheiden zu können. Dies kann sowohl zur Erstellung von Serienbriefen als auch zum dynamischen Erzeugen von einzelnen Texten verwendet werden.

Standardmäßig wird die Tilde "~" als Markierungszeichen verwendet. Das Zeichen kann aber im Applikationsobjekt in der Eigenschaft RtfMixMarker für RtfEdit-Objekte und in der Eigenschaft DocMixMarker für CtxDocEdit-Objekte auf ein beliebiges anderes Zeichen geändert werden.

Das Mischen des Textes mit den Daten kann zu verschiedenen Zeitpunkten erfolgen:

- Beim Lesen eines Textes in ein RtfEdit-Objekt

Dazu muss bei den Befehlen (zum Beispiel WinRtfLoad()) die Option WinRtfLoadMix angegeben werden.

- Beim Speichern eines Textes aus einem RtfEdit-Objekt

Dazu muss bei den Befehlen (zum Beispiel WinRtfSave()) die Option WinRtfSaveMix angegeben werden.

- Beim Lesen eines Textes in ein CtxDocEdit-Objekt

Dazu muss bei den Befehlen (zum Beispiel WinDocLoadName()) die Option WinDocLoadMix angegeben werden.

- Beim Speichern eines Textes aus einem CtxDocEdit-Objekt

Dazu muss bei den Befehlen (zum Beispiel WinDocSaveName()) die Option WinDocSaveMix angegeben werden.

- Beim Drucken eines Textes mit dem PrtRtf-Objekt

Ist beim Drucken des Objektes in der Eigenschaft PrtRtfFlags die Ausprägung PrtRtfMix gesetzt, werden die Markierungen durch die entsprechenden Inhalte ersetzt.

Folgende Markierungen werden beim Mischen von Text und Daten ersetzt:

- ~F:<Feldname>~

Als <Feldname> wird der Name eines Feldes aus der Datenstruktur oder die Nummer der Datei, des Teildatensatzes und des Feldes durch Komma getrennt verwendet. Der Platzhalter wird durch den Inhalt des angegebenen Feldes ersetzt. Das Ausgabeformat wird durch die Ländereinstellungen des Betriebssystems bestimmt. Soll ein anderes Ausgabeformat verwendet werden, kann mit den Befehlen LocaleLoad() und LocaleSelect() dieses Format ausgewählt werden. Darüber hinaus können je nach Typ des Feldes noch folgende Formatoptionen angegeben werden:

Kontakt

Feldtyp	Option	Beschreibung
Alphanumerisch '/W1' <u>(alpha)</u>		Erste Wortumstellung: Die Wortumstellungen gehen davon aus, dass die verschiedenen Namensbestandteile (Name, Vorname und Titel) durch Kommata voneinander getrennt in einem Datenbankfeld stehen ('Müller-Lüdenscheidt, Franz, Dr.'). Durch die erste Wortumstellung wird der Name in der Form Titel, Vorname und Name dargestellt. Aus dem Namen im Beispiel wird dadurch 'Dr. Franz Müller-Lüdenscheidt'.
'/W2'		Zweite Wortumstellung: In dieser Wortumstellung wird der Vorname ausgelassen. Das Beispiel wird in 'Dr. Müller-Lüdenscheidt' umgewandelt.
'/W3'		Dritte Wortumstellung: Hier werden keine Doppelnamen angezeigt. Das Beispiel wird in 'Dr. Müller' umgewandelt.
'/Bn'		Ist das Feld nicht leer, werden nach der Ausgabe des Inhaltes noch die in <i>n</i> angegebenen Leerzeichen ausgegeben. Es können bis zu 999 Leerzeichen angegeben werden.
'/Sn'		Das Feld wird mit einer statischen Ausgabelänge ausgegeben. Unabhängig von Inhalt des Feldes werden die in <i>n</i> angegebenen Zeichen ausgegeben. <i>n</i> kann einen Wert von 1 bis 999 angegeben werden. Der Ausgabewert wird abgeschnitten oder mit Leerzeichen aufgefüllt.
Ganzzahlig (<u>byte</u> , '/G' <u>word</u> , <u>int</u> und <u>bigint</u>)		Es findet keine Tausender-Trennung statt.
'/Z'		Ist der darzustellende Wert gleich 0, wird keine Ausgabe erzeugt (Nullunterdrückung).
Gleitkomma (<u>float</u> und <u>decimal</u>)	'/G'	Es findet keine Tausender-Trennung statt.
	'/Z'	Ist der darzustellende Wert gleich 0, wird keine Ausgabe erzeugt (Nullunterdrückung).
	'/P'	Anstelle des Dezimaltrennzeichens wird ein Punkt ausgegeben.
	'/n'	

Kontakt

Datum (date) '/L'

Über diese Option wird die Anzahl der Nachkommastellen definiert. Soll die Anzahl der Nachkommastellen definiert werden, muss entweder zuvor mindestens einer der bereits genannten Formatoptionen oder ein "/" ohne Optionen angegeben werden. Werden keine Nachkommastellen angegeben, werden alle Nachkommastellen angezeigt.

Zeit (time) '/S'
 '/H'

Mit dieser Option wird das lange Datumsformat eingestellt. Wird kein langes Datumsformat gewählt, erscheint das Datum in der kurzen Anzeige.

Logisch (logic)

'/<true>/<false>' Logische Felder und Variablen müssen mit Werten versehen werden, die in Abhängigkeit des Wertes angezeigt werden sollen. Die erste Zeichenkette wird angezeigt, wenn der Wert true ist, die zweite Zeichenkette wird angezeigt, wenn der Wert false ist.

- ~G:<Variablenbereich>:<Variablenname>~

Für Variablen gelten die gleichen Formatierungsmöglichkeiten, wie für Felder.

Es können mehrere Optionen hinter einem "/" gesammelt werden. Bei Zahlenwerten können die Formatierungsanweisungen (bis auf die Anzahl der Nachkommastellen) kombiniert werden: '~F:KND.fNumber/GZ/2~'. Die Anzahl der Nachkommastellen werden in einer zweiten Option angegeben.

Damit die Platzhalter korrekt ersetzt werden können, müssen die Namen korrekt geschrieben sein und es darf kein Formatwechsel zwischen den Markierungszeichen vorgenommen werden.

Markierungen, die nicht ersetzt werden können, bleiben im Text enthalten.

Beispiele:

Platzhalter	ersetzter Text
'~F:KND.aName~'	'Müller, Walter, Dr.'
'~F:KND.aName/W1~'	'Dr. Walter Müller'
'~F:10,1,2/W2~'	'Dr. Müller'
'~F:KND.fNumber~'	'1.256'
'~F:KND.fNumber/G/2~'	'1256,00'
'~F:KND.fNumber//2~'	'1.256,00'
'~F:KND.lActive/ja/nein~' 'nein'	

Kontakt

PrtPdf



Objekt zum Drucken von PDF-Dokumenten

[Liste](#),

Siehe [Eigenschaften](#),

[DocView](#)

Das Objekt ermöglicht den Druck von PDF-Dokumenten, ohne dass hierzu eine entsprechende Software eines Drittanbieters auf dem Rechner installiert sein muss. Über die Eigenschaft [CurrentInt](#) kann die Nummer der anzuzeigenden Seite eingestellt werden.

Der Pfad und der Name des Dokuments wird in der Eigenschaft [FileName](#) angegeben.



Unter Windows XP können die Inhalte von PDF-Dokumenten der Version 1.7 in diesem Objekt aus Kompatibilitätsgründen weder angezeigt noch gedruckt werden.

Kontakt

PrtTblCellText

 PrtTblCellText

Siehe [Liste](#),

[Eigenschaften](#)

Das Zelltext-Objekt ist im wesentlichen identisch zum [Text-Objekt](#). Es kann jedoch nur in ein [Zell-Objekt](#) eingefügt werden.

PrtPicture



PrtPicture

[Liste](#),

Siehe [Eigenschaften](#),
[Picture](#)

Das PrtPicture-Objekt kann extern oder in der Datenbank vorliegende Grafiken in den Formaten BMP, JPEG, GIF, PNG, TIFF und ICO anzeigen. Bei einem mehrseitigen TIFF-, oder ICO-Bild, wird die darzustellende Seite in der Eigenschaft CurrentInt angegeben. Die Eigenschaft AutoSize legt fest, ob sich die Größe des Objekts an die Größe der Grafik anpassen soll oder nicht.

Die Eigenschaft Caption bestimmt den Namen der Grafik. Die anzuzeigende Grafik kann extern oder in der Datenbank vorliegen. Die Angabe einer externen Datei kann mit einem absoluten oder relativen Pfad erfolgen. Bei der Angabe eines absoluten Pfades wird ein '*', bei der eines relativen Pfades (relativ zum Startverzeichnis des CONZEPT 16-Clients) wird ein '!' vorangestellt.

Bei Angabe des Namens ohne Präfix ('!' und '*') wird eine Bilddatei in der Datenbank referenziert. Über die Ressourcenverwaltung lassen sich Grafiken der unterstützten Formate in die Datenbank einlesen.

Ist die darzustellende Grafik in der Datenbank als binäres Objekt gespeichert, kann es durch die Angabe von '>0' gefolgt vom Pfad und dem Namen des Objektes in der Eigenschaft Caption angezeigt werden. Ist das binäre Objekt verschlüsselt gespeichert, muss der Schlüssel in der Eigenschaft CryptKey angegeben werden. Bei mit DbaConnect() verbundenen Datenbanken wird nach '>' der entsprechende Dateinummernbereich (z. B. '>2') angegeben.

Beispiele:

```
'!Picture\Article.tif'           // externe Datei relativ angegeben'*C:\C16\Picture\Article.tif' //
```

Mit der Eigenschaft PictureMode können unterschiedliche Darstellungsweisen erfolgen.

Kontakt

PrtMetaPicture



PrtMetaPicture

Siehe [Liste](#),

Eigenschaften

Das Objekt PrtMetaPicture-Objekt dient zum Anzeigen von Vektorgrafiken im Metafile-Format. Diese können extern vorliegen oder auch in die Datenbank importiert werden. Der Name der Grafik wird über die Eigenschaft Caption gesetzt werden.

Die Angabe einer externen Datei kann mit einem absoluten oder relativen Pfad erfolgen. Bei der Angabe eines absoluten Pfades wird ein '*', bei der eines relativen Pfades (relativ zum Startverzeichnis des CONZEPT 16-Clients) wird ein '!' vorangestellt.

Bei Angabe des Namens ohne Präfix ('!' und '*') wird eine Bilddatei in der Datenbank referenziert. Über die Ressourcenverwaltung lassen sich Grafiken vom Format BMP, JPG und TIFF in die Datenbank einlesen.

Ist die darzustellende Grafik in der Datenbank als binäres Objekt gespeichert, kann es durch die Angabe von '>0' gefolgt vom Pfad und dem Namen des Objektes in der Eigenschaft Caption angezeigt werden. Ist das binäre Objekt verschlüsselt gespeichert, muss der Schlüssel in der Eigenschaft CryptKey angegeben werden. Bei mit DbaConnect() verbundenen Datenbanken wird nach '>' der entsprechende Dateinummernbereich (z. B. '>2') angegeben.

Beispiele:

```
'!Picture\Artikel.emf'          // externe Datei relativ angegeben'*C:\C16\Picture\Artikel.emf' //
```

Verschiedene Darstellungsweisen werden über die Eigenschaft PictureMode gesetzt.

Der Vorteil von Meta-Grafiken liegt in der Skalierbarkeit der Grafik, die (anders als bei Rastergrafiken) ohne Darstellungsverluste erfolgt.

Folgende Metafile-Formate werden unterstützt:

- Windows Meta Files

Diese Dateien haben gewöhnlich die Endung WMF. Da dieses Dateiformat jedoch keine Größen-Angaben für die Grafik speichert, kann diese bei Bedarf über die Eigenschaften Width und Height definiert werden.

- Aldus Placeable Meta Files

Diese Dateien haben, wie die Windows Meta Files, ebenfalls die Endung WMF. Sie besitzen auch dasselbe Dateiformat wie diese, jedoch werden zusätzlich auch Größen-Angaben für die Grafik gespeichert. Die Eigenschaften Width und Height werden bei diesem Dateityp ignoriert und in der Eigenschaftsliste ausgeblendet.

- Enhanced Meta Files

Kontakt

Diese Dateien haben gewöhnlich die Endung EMF. Das Dateiformat ist nicht mit den obigen Dateiformaten kompatibel. Auch hier wird eine Größen-Angabe für die Grafik gespeichert, so dass die Eigenschaften Width und Height ignoriert und in der Eigenschaftsliste ausgeblendet sind.



Durch die Angabe von WinColTransparent als Hintergrundfarbe (ColBkg) kann das Objekt über ein anderes Objekt gedruckt werden.

Kontakt

PrtDivider



Divider

Siehe [Liste](#),

Eigenschaften

Der PrtDivider zeichnet eine horizontale oder vertikale Trennlinie.

Die Richtung wird über die Eigenschaft Vertical gesetzt.

Die Strichstärke der Linie kann über die Eigenschaft WidthPen definiert werden.

Kontakt

PrtBorder



Druckbare Umrandung

Siehe [Liste](#),

Eigenschaften

Mit diesem Objekt können bestimmte Bereiche eines Ausdrucks hervorgehoben werden.

Die Eigenschaft [StylePen](#) bestimmt, ob eine Linie und welche Linie zur Umrandung gezeichnet wird. Die Farbe der Linie ergibt sich aus den Einstellungen in den Eigenschaften [ColFg](#), [ColFillEven](#) und [ColFillOdd](#).

Die Dicke der Linie ergibt sich aus der Eintragung in der Eigenschaft [WidthPen](#).

Wird das [PrtBorder](#)-Objekt eingefügt, während ein anderes Objekt selektiert ist, wird der Rahmen automatisch um dieses Objekt gezeichnet. Das Objekt wird in die Eigenschaft [ObjLink](#) eingetragen. Der Rahmen passt sich immer an die Größe des Objekts an.

Kontakt

PrtGroupbox



PrtGroupbox

Siehe [Liste](#),

Eigenschaften

Das PrtGroupbox-Objekt dient der Aufnahme von Unterobjekten, die in dem PrtGroupbox-Objekt gruppiert werden und auch eine relative Position zum PrtGroupbox-Objekt besitzen.

Die PrtGroupbox kann entweder direkt in ein PrintForm- oder Seiten-Objekt eingefügt werden. Eine PrtGroupbox kann wiederum eine PrtGroupbox oder Druckobjekte enthalten. Jedoch kann eine PrtGroupbox keine Seiten- oder Tabellen- Objekte aufnehmen.

PrtBarcode



Barcode drucken

[Liste](#),

[Eigenschaften](#),

Siehe [Erzeugen von](#)

[Barcodes](#)

[\(Blog\)](#)

Dieses Objekt stellt Barcodes oder Strichcodes dar. Der Barcode kann über die Eigenschaften [Caption](#) oder [DbFieldName](#) gesetzt werden. Der Typ des Barcodes wird in der Eigenschaft [TypeBarcode](#) oder als Präfix angegeben. Die Liste der unterstützten Barcodes befindet sich in der Eigenschaft [TypeBarcode](#).

Das Format wird durch die Länge der Barcodenummer und der integrierten Prüfziffern automatisch erkannt. Um ein Format zu erzwingen, kann der Barcodenummer ein Präfix vorangestellt werden: 'EAN', 'UPC', 'ISBN', 'I25', 'Code39N', 'Code39C', 'Code128B', 'Code128C' oder 'Code128X'. Der Typ kann ebenfalls in der Eigenschaft [TypeBarcode](#) angegeben werden.

Beispiele:

```
'3883224537' 'ISBN3-88322-453-7' 'Code39NABC123'
```

PrtGanttGraph



Gantt-Diagramm

Liste,

Siehe Eigenschaften,

Befehle,

GanttGraph

Dieses Objekt entspricht weitgehend dem GanttGraph-Objekt. Dieses Objekt wird verwendet, um einen GanttGraphen zu drucken.

Intervalle, Linien und Boxen können mit den gleichen Befehlen dem Objekt hinzugefügt oder entfernt werden.

Intervall-Positionen und -Größen werden wie gewohnt in Zelleinheiten angegeben. Eigenschaften die beim GanttGraph-Objekt in Pixel angegeben wurden, werden beim PrtGanttGraph-Objekt in logischen Einheiten definiert. Ein Beispiel dafür ist die Eigenschaft CellSizeHorz, die die Breite einer Zelle in Pixeln (im GanttGraph-Objekt), bzw. in logischen Einheiten (im PrtGanttGraph-Objekt) definiert.

Soll ein bestehendes GanttGraph-Objekt gedruckt werden, kann es in der Entwicklungsumgebung über die Windows-Zwischenablage aus dem Fenster-Objekt kopiert und in ein Druck-Objekt eingefügt werden. Das GanttGraph-Objekt und die untergeordneten Objekte werden automatisch in ihre entsprechenden Druck-Objekte umgewandelt.

Die Achsen des Objektes können mit dem PrtGanttGraph-Editor erstellt und verändert.

Kontakt

PrtGanttAxis



x- und y-Achse

Siehe [Liste](#),

[Eigenschaften](#)

Ein [PrtGanttGraph](#) besteht aus einer oder mehreren x- und y-Achsen. Die Achsen werden mit dem [PrtGanttGraph-Editor](#) erstellt und verändert.

Kontakt

PrtInterval



PrtInterval-Objekt

Siehe [Liste](#),

[Eigenschaften](#)

Ein [PrtGanttGraph](#) besteht aus einem oder mehreren Intervall-Objekten. Das Anlegen und Entfernen der Zeitintervalle erfolgt über die Befehle [WinGanttIvlAdd\(\)](#) und [WinGanttIvlRemove\(\)](#).

Kontakt

PrtIvlBox



PrtIvlBox-Objekt

Siehe [Liste](#),

Eigenschaften

Bei einem PrtIvlBox-Objekt handelt es sich um einen farblich oder durch eine Schraffur hervorgehobenen Bereich innerhalb eines PrtGantt-Diagramms.

Das Anlegen eines Box-Objekts geschieht durch den Befehl [WinGanttBoxAdd\(\)](#).

Kontakt

PrtIvlLine



PrtIvlLine-Objekt

Siehe [Liste](#),

[Eigenschaften](#)

Bei einem PrtLine-Objekt handelt es sich um eine Hilfslinie innerhalb eines [PrtGanttGraph-Diagramms](#).

Das Anlegen eines Line-Objekts geschieht durch den Befehl [WinGanttLineAdd\(\)](#).

Drucker-Objekte

Siehe Alle Druckformular-Objekte

PrintDevice 

Printer 

PrinterList 

PrintJob 

Kontakt

PrintDevice

PrintDevice

Liste,

Siehe Eigenschaften,

PrtDeviceOpen()

Für die Ausgabe eines Druckjobs auf einen Drucker muss ein PrintDevice-Objekt an den Prozedurbefehl PrtJobClose() übergeben werden.

Das PrintDevice-Objekt ist die Schnittstelle zum Druckertreiber auf dem die Druckausgabe erfolgen soll.

Ein Deskriptor auf ein PrintDevice-Objekt wird durch PrtDeviceOpen() zurückgegeben. Wenn das PrintDevice-Objekt nicht mehr benötigt wird, muss der Deskriptor durch PrtDeviceClose() wieder geschlossen werden.

Über die Eigenschaften des PrintDevice-Objekts können verschiedene Informationen ermittelt werden.

Der Name des Druckers, der dem Gerät zugeordnet ist kann über die Eigenschaft NamePrinter erfragt werden. Die Druckertreiber-Version liefert die Eigenschaft Version.

Das Seitenformat und die Seitenausrichtung werden über die Eigenschaften PageFormat und Orientation gesetzt oder ermittelt. Seitenhöhe und Seitenbreite können über die Eigenschaften PageWidth und PageHeight ermittelt werden.

Die Druckqualität kann über die Eigenschaften QualityX und QualityY gesetzt oder ermittelt werden.

Kontakt

Printer



Printer

Liste,

Eigenschaften,

Siehe

Druckerattribute,

Druckerstatus

Das Printer-Objekt ist die Umsetzung eines physikalisch vorhandenen System-Druckers.

Ein Deskriptor auf das Objekt kann über die Eigenschaft PrinterList ermittelt werden.

Die Eigenschaften Name, Caption und Comment sind direkt abrufbar, nachdem der Deskriptor ermittelt wurde. Alle anderen Eigenschaften liefern erst nach dem Befehl PrtPrinterRefresh() die aktuellen Werte.

PrinterList

PrinterList

Liste,
Eigenschaften,

Siehe PrtInfo(),
PrtSearch()

Dieses Objekt enthält eine Liste aller auf dem System installierter Drucker. Die Drucker können nacheinander mit dem Befehl PrtInfo() gelesen oder ein bestimmter Drucker mit dem Befehl PrtSearch() gesucht werden.

Kontakt

PrintJob

PrintJob

Liste,
Eigenschaften,
PrtJobOpen(),

Siehe

PrtPage,
Befehle,
Beispiel

Das Druckjob-Objekt dient dazu, im Designer erstellte Formular-Objekte aufzubereiten und zu drucken oder in der Druckvorschau anzuzeigen. Zu diesem Zweck erstellt das Objekt eine externe Datei, die Druckjob-Datei.

Einen Deskriptor auf ein Druckjob-Objekt liefert der Prozedurbefehl PrtJobOpen(). Mit ihm kann sowohl ein Druckjob erstellt, als auch ein bereits erzeugter Druckjob wieder eingelesen werden, um ihn anschließend zu drucken. Wenn die Arbeit mit einem Druckjob beendet ist, muss der Deskriptor über PrtJobClose() wieder geschlossen werden.

Beim Erzeugen eines Druckjobs kann angegeben werden, ob ein in der Datenbank abgelegtes PrintDoc-Objekt geladen oder ein leeres PrintDoc-Objekt angelegt werden soll.

Somit ist dem Druckjob immer ein PrintDoc-Objekt zugeordnet. Der Deskriptor des Dokuments kann ermittelt werden durch einen PrtInfo()-Aufruf mit dem Argument PrtDoc.

Mit dem Prozedurbefehl PrtJobWrite() kann eine Seite beendet, eine neue Seite gestartet werden oder ein automatischer Seitenwechsel erfolgen. Wird PrtJobWrite() mit dem Argument PrtJobPageStart oder PrtJobPageBreak aufgerufen, wird ein Deskriptor auf das neu erzeugte Seitenobjekt zurückgegeben.

Durch die Prozedurbefehle PrtAdd() und PrtAddByName() können Objekte zu einem Seiten-Objekt (PrtPage) des Druckjobs hinzugefügt werden. Die Objekte werden relativ zur aktuellen Druckposition hinzugefügt.

Über die Eigenschaft BoundAdd kann festgestellt werden, wie viel Platz bereits auf der Seite belegt ist. BoundMax gibt Auskunft darüber, wieviel Platz auf der Seite insgesamt zu Verfügung steht.

Wird der Ausdruck des Formulars vom Acrobat PDF-Writer vorgenommen, können über die Eigenschaften PdfAuthor, PdfCreator, PdfFileName und PdfTitle die Eigenschaften des Dokuments gesetzt werden.

Seiten-Objekte

Siehe Alle Druckformular-Objekte

PrtPage



PrtPageTable



PrtPageHeader



PrtPageFooter



Kontakt

PrtPage



[PrtPage](#)

[Liste](#)

Siehe [Eigenschaften](#),

[Beispiel](#)

Das PrtPage-Objekt existiert in zwei Ausprägungen: zum einen kann es im Formular-Designer des Designers einem [PrintDoc](#)-Objekt zum anderen prozedural einem [PrintJob](#)-Objekt untergeordnet werden. Die prozedurale Zuordnung erfolgt über den Befehl [PrtJobWrite\(\)](#).

Prinzipiell besteht das PrtPage-Objekt aus drei Komponenten: dem Seitenkopf ([PrtPageHeader](#)), dem Seitenfuß ([PrtPageFooter](#)) und dem Seiteninhalt. Der Seitenkopf- und Fuß können nur im Formular-Designer gesetzt werden.

[PrtPageHeader](#) und [PrtPageFooter](#) können über die Objektpalette in ein Seitenobjekt hinzugefügt werden, sofern die Objekte nicht schon in der Seite enthalten sind.

Der Seiteninhalt kann aus beliebigen Druckobjekten bestehen, die ebenfalls über die Objektpalette eingefügt werden.

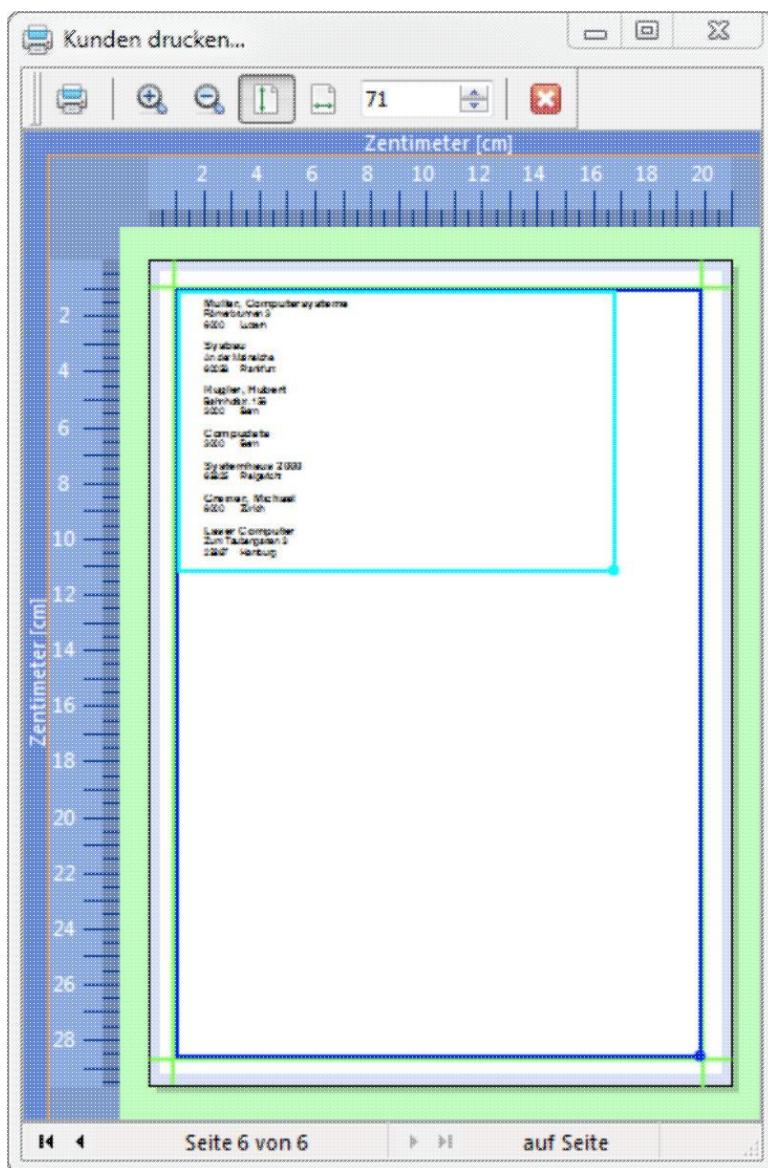
Die Hintergrundfarbe der Seite kann über die Eigenschaft [ColBkgPage](#) gesetzt werden. Die Hintergrundfarbe für untergeordnete Objekte wird dagegen über die Eigenschaft [ColBkg](#) gesetzt.

Wird das Seiten-Objekt prozedural gefüllt, kann über die Eigenschaften [BoundMax](#) der Platz, der auf der leeren Seite zur Verfügung steht (dunkelblauer Bereich) und mit [BoundAdd](#) der Platz, der bereits durch angehängte Druck-Objekte verbraucht wurde (hellblauer Bereich) ermittelt werden. In grün sind die Seitenränder angezeigt.



Das Objekt hat standardmäßig einen Rand von einem Zentimeter. Dieser kann über die Eigenschaften [AreaMarginLeft](#), [AreaMarginTop](#), [AreaMarginRight](#) und [AreaMarginBottom](#) verändert werden.

Kontakt



PrtPageTable



PrtPageTable

Siehe [Liste](#),

Eigenschaften

Das PrtPageTable-Objekt ist ein spezielles Seitenobjekt, das nur einem [PrintDocRecord](#)-Objekt (Tabellen-Dokument) hinzugefügt werden kann.

Das PrtPageTable-Objekt kann ebenfalls [PrtPageHeader](#) und [PrtPageFooter](#) enthalten, jedoch kann der Seiteninhalt nur aus maximal einem Tabellen-Objekt bestehen.

Die Hintergrundfarbe der Seite kann über die Eigenschaft [ColBkgPage](#) gesetzt werden. Die Hintergrundfarbe für untergeordnete Objekte wird dagegen über die Eigenschaft [ColBkg](#) gesetzt.

Kontakt

PrtPageHeader



PrtPageHeader

Siehe [Liste](#),

[Eigenschaften](#)

Das PrtPageHeader-Objekt kann nur in ein [PrtPage](#)- oder [PrtPageTable](#)-Objekt eingefügt werden.
Dem Objekt können beliebige Druckobjekte hinzugefügt werden.

Die Eigenschaft [AutoSize](#) bestimmt, ob sich das Objekt in der Größe an den Seitenrand der Seite anpasst oder nicht.

Ist die Eigenschaft [false](#), kann das PrtPageHeader-Objekt in der Größe verändert werden.

Kontakt

PrtPageFooter



PrtPageFooter

Siehe [Liste](#),

Eigenschaften

Das PrtPageFooter-Objekt kann nur in ein [PrtPage](#)- oder [PrtPageTable](#)-Objekt eingefügt werden.
Dem Objekt können beliebige Druckobjekte hinzugefügt werden.

Die Eigenschaft [AutoSize](#) bestimmt, ob sich das Objekt in der Größe an den Seitenrand der Seite anpasst oder nicht.

Ist die Eigenschaft [false](#), kann das PrtPageFooter-Objekt in der Größe verändert werden.

Tabellen-Objekte

Siehe [Tabellen-Bereiche, Alle Druckformular-Objekte](#)

<u>PrtTblCommon</u>	
<u>PrtTblAreaHdrTbl</u>	
<u>PrtTblAreaHdrPage</u>	
<u>PrtTblAreaHdrData</u>	
<u>PrtTblAreaData</u>	
<u>PrtTblAreaFtrData</u>	
<u>PrtTblAreaFtrPage</u>	
<u>PrtTblAreaFtrTbl</u>	

Kontakt

PrtTblCommon



PrtTblCommon

Siehe [Liste](#),

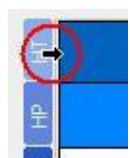
[Eigenschaften](#)

PrtTblCommon ist ein allgemeines Tabellen-Objekt. Es kann lediglich in ein leeres PrintDocRecord-Objekt eingefügt werden.

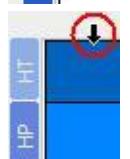
Das Tabellen-Objekt besteht aus Tabellen-Bereichen. Es gibt insgesamt sieben unterschiedliche PrtTblArea-Objekte. Eine Tabelle kann jedes dieser PrtTblArea-Objekte enthalten, jedoch maximal ein einziges desselben Typs.

Im Designer werden Fähnchen an der linken Seite der Tabelle dargestellt. Jedes Fähnchen ist genau einem Tabellen-Bereich zugeordnet.

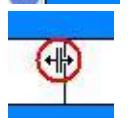
Beim Klicken auf ein Fähnchen wird der entsprechende Bereich selektiert. Mit einem Doppelklick wird die Tabelle selektiert.



Alle Zellen einer Zeile können selektiert werden, indem an den linken Rand der ersten Zelle geklickt wird. Dort verwandelt sich der Mauszeiger in einen nach rechts zeigenden Pfeil.



Alle Zellen einer Spalte können selektiert werden, indem an den oberen Rand der ersten Zelle des Tabellen-Bereiches geklickt wird. Dort verwandelt sich der Mauszeiger in einen nach unten zeigenden Pfeil.



Die Zellgröße einer Zelle kann am rechten oder unteren Rand der Zelle verändert werden. Dort verwandelt sich der Mauszeiger in einen Doppelpfeil.

Kontakt

PrtTblAreaData



PrtTblAreaData

Liste,

Siehe Eigenschaften,
Tabellen-Bereiche

Im Designer kann maximal ein Datenbereich pro Tabelle angelegt werden. Während der Druckausgabe werden Instanzen der Vorlage erstellt. Dies geschieht automatisch und für jeden Datensatz der Datensatzmenge.

Die Datensatzmenge wird im PrintDocRecord-Formular über die Eigenschaften DbFileNo, DbKeyNo und DbLinkFileNo definiert.

Kontakt

PrtTblAreaHdrTbl



PrtTblAreaHdrTbl

Liste,

Siehe Eigenschaften,

Tabellen-Bereiche

Die Ausgabe des Tabellenfuß-Bereiches erfolgt während des Druckens einmalig vor der Ausgabe

aller anderen Tabellen-Bereiche.

Kontakt

PrtTblAreaFtrTbl



PrtTblAreaFtrTbl

Liste,

Siehe Eigenschaften,

Tabellen-Bereiche

Die Ausgabe des Tabellenfuß-Bereiches erfolgt während des Druckens einmalig nach der Ausgabe aller anderen Tabellen-Bereiche.

Kontakt

PrtTblAreaHdrPage



PrtTblAreaHdrPage

Liste,

Siehe Eigenschaften,

Tabellen-Bereiche

Die Ausgabe des Seitenkopf-Bereiches erfolgt während des Druckvorgangs auf jeder Seite vor der Ausgabe des Datenkopf-Bereiches und der Datenbereiche für diese Seite. Im Designer kann maximal ein Seitenkopf-Bereich definiert werden.

Kontakt

PrtTblAreaFtrPage



PrtTblAreaFtrPage

Liste,

Siehe Eigenschaften,

Tabellen-Bereiche

Die Ausgabe des Seitenfuß-Bereiches erfolgt während des Druckvorgangs auf jeder

Seite nach der Ausgabe der Datenbereiche für die Seite und des Datenfuß-Bereiches.

Im Designer kann maximal ein Seitenfuß-Bereich definiert werden.

Kontakt

PrtTblAreaHdrData



TblAreaHdrData

Liste,

Siehe Eigenschaften,

Tabellen-Bereiche

Die Ausgabe des Datenkopf-Bereiches erfolgt während des Druckvorgangs vor der Ausgabe der Datenbereiche für die aktuelle Seite.

Kontakt

PrtTblAreaFtrData



PrtTblAreaFtrData

Liste,

Siehe Eigenschaften,
Tabellen-Bereiche

Die Ausgabe des Datenfuß-Bereiches erfolgt während des Druckvorgangs nach der Ausgabe der Datenbereiche für die aktuelle Seite.

Kontakt

PrtTblCell



PrtTblCell

Siehe [Liste](#),

[Eigenschaften](#)

Das Zell-Objekt kann nur zu einem [Tabellen-Bereich](#) hinzugefügt werden. Über die Eigenschaft [HeightMode](#) kann festgelegt werden, ob und wie sich die Höhe der Zelle an den Zellinhalt anpasst.

Die Eigenschaft [Width](#) definiert die Breite der Zelle in [logischen Einheiten](#). Die Eigenschaft [Height](#) die Höhe der Zelle in [logischen Einheiten](#). Die Höhe der Zelle kann nicht verändert werden, wenn die Eigenschaft [HeightMode](#) auf [PrtHeightContent](#) gesetzt ist.

System-Objekte

Liste der System-Objekte

- [Application](#)
- [Benutzer](#)
- [Benutzergruppe](#)
- [Binäre Objekte](#)
- [Chart-Objekt](#)
- [Dynamische Strukturen](#)
- [Elementgruppe](#)
- [HTTP-Objekt](#)
- [Locale](#)
- [Memory-Objekt](#)
- [PDF-Objekt](#)
- [System](#)
- [TapiDevice](#)
- [Validierungselemente](#)
- [Zentrale Datenobjekte](#)

Kontakt

Locale



Länderspezifische Einstellungen

Siehe [Liste](#),

Eigenschaften

Das Locale-Objekt ist ein Systemobjekt. Es wird nicht in der Entwicklungsumgebung angelegt. Das Objekt wird mit dem Befehl [LocaleLoad\(\)](#) geladen. Anschließend stehen die Eigenschaften des Objektes zur Verfügung.

In dem Locale-Objekt stehen die länderspezifischen Einstellungen zur Verfügung. Das Objekt kann dazu verwendet werden, um die Einstellungen auszulesen. Änderungen in diesem Objekt wirken sich nur an den Stellen aus, in denen das Objekt angegeben wird.

Das Objekt kann bei der [Konvertierung](#) von Datentypen von oder nach [alpha](#) angegeben werden, um länderspezifische Besonderheiten zu berücksichtigen. Ebenso kann bei einigen Eingabeobjekten der Deskriptor des Locale-Objektes in die Eigenschaft [Locale](#) eingetragen werden, um den Inhalt des Eingabeobjektes mit den geladenen Ländereinstellungen (unabhängig von den länderspezifischen Einstellungen des Betriebssystems) darzustellen.

Die Eigenschaften des Objektes können mit dem Befehl [SysPropGet\(\)](#) ausgelesen und mit [SysPropSet\(\)](#) gesetzt werden.

Soll generell eine bestimmte Einstellung verwendet werden, kann mit dem Befehl [LocaleSelect\(\)](#) ein bestimmtes Locale gesetzt werden.

Das Objekt kann mit dem Befehl [LocaleUnload\(\)](#) wieder entfernt werden.

Kontakt

System



Das System-Objekt

Siehe [Sys](#), [Liste](#),

Eigenschaften

Das System-Objekt dient zur Ermittlung von system- und laufzeitspezifischen Eigenschaften und Funktionen. Es kann nur zur Laufzeit angesprochen werden.

Der Deskriptor des System-Objektes kann durch das Schlüsselwort [Sys](#) ermittelt werden.

Kontakt

Benutzerverwaltung

Vergabe und Verwaltung von benutzerabhängigen Rechten Siehe [Blog](#)

Die Benutzerverwaltung wird im Designer über das Menü Extras / Benutzerverwaltung... aufgerufen. Im folgenden wird das Benutzersystem und der Umgang damit erläutert:

- [Benutzer](#)
- [Benutzergruppen](#)
- [Elementgruppen](#)
- [Zusammenarbeit der Objekte](#)
- [Berechtigungen](#)
- [Verwaltung](#)

Benutzer

Ausgangspunkt eines jeden Benutzersystems ist der Benutzer selbst. In CONZEPT 16 wird der Benutzer durch ein [Benutzer-Objekt](#) repräsentiert. Jedes Benutzer-Objekt verfügt über verschiedene [Eigenschaften](#) (Properties). Die wichtigsten Eigenschaften sind der Name ([UrmPropName](#)) und das Kennwort ([UrmPropPassword](#)), mit denen sich der Anwender an der Datenbank anmeldet. Der Name muss innerhalb einer Datenbank eindeutig sein. Neben den vordefinierten Eigenschaften kann das Benutzer-Objekt um zusätzliche Eigenschaften erweitert werden. Die Applikation kann dadurch zusätzliche Daten im Benutzer-Objekt speichern.

In einer neu erstellten Datenbank sind bereits die beiden Benutzer SUPERUSER und USER mit leerem Kennwort angelegt. Bei der Anmeldung als SUPERUSER reicht die Angabe von SU als Benutzer aus. Der Benutzer SUPERUSER kann nicht gelöscht werden.

Unabhängig vom verwendeten Client ist zum Öffnen einer Datenbank immer die Angabe eines Benutzers erforderlich. Alle Aktionen (Ausführen von Prozeduren, Laden von Dialogen usw.) finden somit immer in einem Benutzerkontext statt.

Die Berechtigungen eines Benutzers ergeben sich aus seiner Zugehörigkeit zu einer oder mehreren [Benutzergruppen](#), im Benutzer-Objekt selbst werden keine Rechte definiert.

Benutzergruppen

Eine Benutzergruppe fasst mehrere Benutzer zusammen, um eine einfache Vergabe von Rechten zu gewährleisten. Die Berechtigungen werden immer für eine Benutzergruppe definiert, wodurch alle Mitglieder dieser Gruppe die selben Rechte erhalten.

Ein Benutzergruppen-Objekt wird durch seinen Namen identifiziert und verfügt über eine Reihe von vordefinierten [Eigenschaften](#). Genau wie beim Benutzer-Objekt können zusätzliche Eigenschaften angelegt werden. Neben seinen Eigenschaften verfügt das Benutzergruppen-Objekt über Rechte auf mehrere [Elementgruppen](#). Ein Benutzer-Objekt kann Mitglied in mehreren Benutzergruppen sein. Die effektiven Rechte des Benutzers ergeben sich in diesem Fall durch die Addition der Rechte aller

Kontakt

Benutzergruppen, zu denen er gehört.

In einer leeren Datenbank sind bereits die beiden Benutzergruppen `_Everyone` und `_Administrators` vorhanden. Die Gruppe `_Everyone` dient als Basisgruppe für alle Benutzer, um die minimal benötigten Rechte für die Applikation zur Verfügung zu stellen. Die Gruppe `_Administrators` verfügt automatisch über alle Rechte in der Datenbank. Beide Gruppen können nicht gelöscht werden. Der Benutzer `SUPERUSER` ist Mitglied von `_Administrators` und der Benutzer `USER` Mitglied von `_Everyone`.

Elementgruppen

Genauso wie Benutzergruppen mehrere Benutzer mit den gleichen Benutzerrollen zusammenfassen, fassen Elementgruppen mehrere Applikationselemente zusammen, für die die selben Rechte benötigt werden. Die Elementgruppe kann sowohl reale Objekte aus der Datenbank (bestimmte Dialoge, einzelne Prozeduren, ein oder mehrere Datensatz-Tabellen usw.) als auch virtuelle Objekte der Applikation (Programm-Module, Funktionalitäten usw.) beinhalten. Durch die Zusammenfassung von mehreren Elementen zu Elementgruppen wird die Rechtevergabe wesentlich übersichtlicher. Gerade wenn es um die Berechtigungen eines Anwendungsbereichs geht, können alle Prozeduren, Dialoge, Tabellen usw. aus diesem Bereich in eine Elementgruppe zusammengefasst werden. Für alle Elemente werden dann gleiche Rechte vergeben. Jedes reale oder virtuelle Element kann Mitglied in bis zu 128 verschiedenen Elementgruppen sein (maximal 32 bei Texten und Prozeduren).

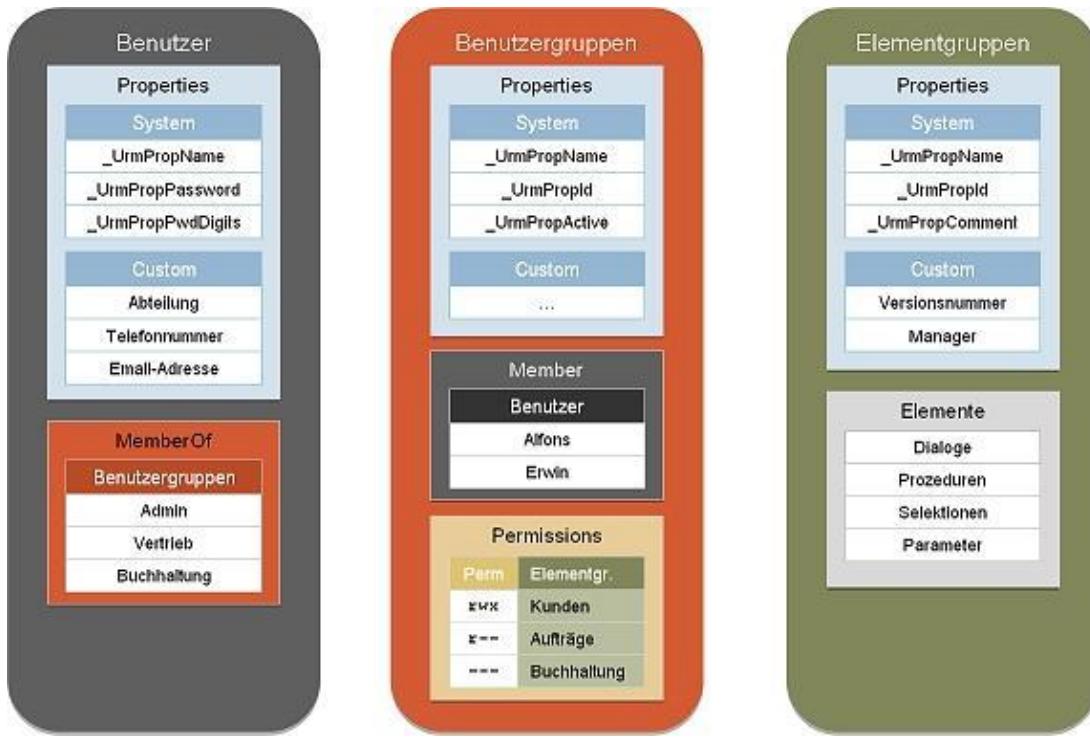
Auch ein Elementgruppen-Objekt wird durch seinen Namen identifiziert und verfügt sowohl über vordefinierte als auch über anwendungsdefinierte Eigenschaften.

In jeder Datenbank sind mehrere sogenannte Standard-Elementgruppen vordefiniert. In diesen Elementgruppen sind jeweils alle Elemente eines bestimmtem Typs enthalten. Zum Beispiel gehören alle Prozeduren automatisch zur Gruppe `_procedure`. Eine Liste aller Standard-Elementgruppen befindet sich im Abschnitt [Verwaltung der Elementgruppen](#).

Zusammenarbeit der Objekte

Die drei Objekte der Benutzerverwaltung treten über Listen miteinander in Beziehung. Innerhalb des Benutzer-Objekts gibt es eine Mitgliedsliste von Benutzergruppen. Bei den Benutzergruppen gibt es eine Liste mit den Mitgliedern (Benutzer-Objekte) und eine Liste mit Elementgruppen einschließlich deren Rechte.

Kontakt



Ob ein Benutzer ein bestimmtes Recht an einem Applikationselement besitzt, hängt also davon ab, ob er Mitglied einer Benutzergruppe ist, die das entsprechende Recht besitzt. Innerhalb der Benutzergruppe muss eine Elementgruppe mit dem entsprechenden Recht eingetragen sein, in der das Element enthalten ist. Ein Recht wird letztendlich in der Benutzergruppe definiert, wo die Elementgruppe eingetragen ist.

Berechtigungen

In CONZEPT 16 werden folgende Berechtigungen verarbeitet:

Read (R)	<u>_UrmPermRead</u>	Lesen und Anzeigen
Create (C)	<u>_UrmPermCreate</u>	Anlegen
Modify (M)	<u>_UrmPermModify</u>	Ändern
Modify Owner (MO)	<u>_UrmPermModifyOwner</u>	Ändern bei Objektbesitz
Delete (D)	<u>_UrmPermDelete</u>	Löschen
Delete Owner (DO)	<u>_UrmPermDeleteOwner</u>	Löschen bei Objektbesitz
Execute (X)	<u>_UrmPermExecute</u>	Ausführen
Config (CF)	<u>_UrmPermConfig</u>	Konfiguration

Da sich die Rechte jeweils auf die Elemente in einer Elementgruppe beziehen, gibt es für die Elementgruppe selbst drei zusätzliche Rechte:

Read Element (RE)	<u>_UrmPermElmGroupRead</u>	Berechtigung Elementgruppen lesen
Insert Element (IE)	<u>_UrmPermElmGroupInsert</u>	Berechtigung Element zur Gruppe hinzufügen

Kontakt

Delete Element

(DE)

Speziell für das Zuordnen von Benutzern zu Benutzergruppen gibt es die folgenden Berechtigungen:

Insert Member

(IM)

Delete Member

(DM)

Berechtigung Element aus Gruppe

löschen

UrmPermElmGroupRead Berechtigung Benutzer zu

Benutzergruppe hinzufügen

UrmPermElmGroupInsert Berechtigung Benutzer aus

Benutzergruppe entfernen

Eine Berechtigung für ein Element entsteht, wenn einer Benutzergruppe eine Elementgruppe hinzugefügt wird und damit gleichzeitig die Rechte auf diese Elementgruppe definiert werden. Ist ein Element in keiner Elementgruppe einer Benutzergruppe enthalten, hat diese Benutzergruppe keine Berechtigung für dieses Element. Die effektiven Rechte für einen Benutzer entstehen durch die Kombination aller Rechte aus den Benutzergruppen, in denen der Benutzer Mitglied ist.

Eine Berechtigung kann erteilt oder entzogen werden. Auf diese Weise können auch "Negativ-Gruppen" erstellt werden, die bestimmte Rechte wieder entfernen. Gibt es für ein Applikationselement sowohl das erteilte, wie das entzogene Recht, setzt sich das entzogene Recht durch.

Beispiel:

In einer Elementgruppe "ElmTblSales" sind alle Tabellen zusammengefasst, die zum Erstellen von Angeboten, Abwickeln von Aufträgen usw. notwendig sind. Darüber hinaus existieren die Elementgruppen "ElmProcCom" und "ElmProcSales" in denen alle Bibliotheken und alle Prozeduren, die zur Abwicklung von Angeboten und Aufträgen benötigt werden, zusammengefasst sind. Es werden drei Benutzergruppen erstellt: "Sales", "Developer" und "Visitor". Folgende Rechte werden vergeben:

	_tables	ElmTblSales	ElmProcSales	ElmProcCom
Visitor	M- D- CF-		C- M- D-	C- M- D-
Sales		R+ C+ M+ D+	X+	X+
Developer		R+ C+ M+ D+ X+ CF+ R+ C+ M+ D+ X+ R+ C+ M+ D+ X+		

Die Elementgruppe "_tables" wird von CONZEPT 16 zur Verfügung gestellt und enthält alle Tabellen der Datenbank. Die mit einem Minuszeichen versehenen Rechte werden der Benutzergruppe entzogen, die Rechte mit einem Pluszeichen werden erteilt.

Ein Benutzer der Benutzergruppe "Sales" hat innerhalb der relevanten Tabellen alle Rechte, die er benötigt, um Datensätze zu lesen, zu erzeugen, zu ändern und zu löschen. Die notwendigen Prozeduren darf er ausführen. Ein Einsehen der Prozeduren oder gar Ändern und neu übersetzen ist ihm nicht möglich. Da der Benutzer nur in einer Benutzergruppe Mitglied ist, bestehen seine effektiven Rechte auch nur aus den in der Benutzergruppe angegebenen Rechten.

Ein weitere Benutzer wird angelegt. Dieser Benutzer ist nur vorübergehend Gast im Verkauf der Firma. Der Benutzer wird den Benutzergruppen "Sales" und "Visitor" zugeordnet. Seine effektiven Rechte sind durch die Besucher-Gruppe eingeschränkt,

Kontakt

so kann er keine Datensätze Ändern oder löschen. Die entzogenen Rechte in bezug auf die Prozeduren wirken sich nicht weiter aus, der schon die "Sales"-Gruppe hier nur ausführende Rechte besitzt.

Die effektiven Rechte des Besuchers sehen wie folgt aus:

_tables ElmTblSales ElmProcSales ElmProcCom Effektiven Rechte
M- D- CF- R+ C+ M- D- CF- C- M- D- X+ C- M- D- X+

Bekommt die Firma einen weiteren Gast in der Entwicklungsabteilung, so bekommt dessen Benutzer die Mitgliedschaft in den Benutzergruppen "Developer" und "Visitor". Die effektiven Rechte sind dann wie folgt:

<u>_tables</u>	ElmTblSales	ElmProcSales	ElmProcCom
Effektiven Rechte	M- D- CF- R+ C+ M- D- X+ CF- R+ C- M- D- X+ R+ C- M- D- X+ Verwaltung		

Die Verwaltung des Benutzersystems kann über den Designer oder prozedural vorgenommen werden. Die Verwaltung über den Designer gliedert sich in die folgenden Bereiche:

- Verwaltung der Benutzer
- Verwaltung der Benutzergruppen
- Verwaltung der Elementgruppen

Für die prozedurale Steuerung des Benutzersystems werden die folgenden Objekte benötigt:

- User (Benutzer)
- UserGroup (Benutzergruppe)
- ElmGroup (Elementgruppe)

Über die Funktion WinUrmDialog() kann die Benutzeroberfläche in die eigene Anwendung integriert werden.

Kontakt

User

 **Benutzer-Objekt der Benutzerverwaltung**

[Liste](#),

Siehe [Eigenschaften](#),

[Benutzerbefehle](#)

Dieses Objekt entspricht einem Benutzer der Datenbank. Mit diesem Benutzer ist eine Anmeldung an der Datenbank möglich, vorausgesetzt das Kennwort ist bekannt und andere Eigenschaften ([UrmPropActive](#) und [UrmPropExpirationDate](#)) verhindern die Anmeldung nicht.

Das Objekt User ist nicht zu verwechseln mit dem angemeldeten Benutzer. In einer leeren Datenbank sind zwei Benutzer vorhanden SUPERUSER und USER. Der Benutzer USER kann über die verschiedenen Befehle der Benutzerpflege manipuliert werden. Der Benutzer SUPERUSER kann seine Eigenschaften nur selbst verändern, andere Benutzer haben dazu keine Berechtigung. Der SUPERUSER kann die Eigenschaften [UrmPropActive](#) und [UrmPropExpirationDate](#) sowie seine Zugehörigkeit zu Benutzergruppen nicht verändern. Alle anderen Benutzer können das eigene Kennwort (sofern nicht durch [UrmPropPwdLocked](#) unterbunden) und alle eigenen benutzerdefinierten Eigenschaften ändern. Zum Ändern ihrer eigenen Systemeigenschaften werden entsprechende [UrmPermModify](#)-Rechte benötigt.

Ein Benutzer kann einer oder mehreren [Benutzergruppen](#) zugeordnet werden (siehe [UrmCreate\(\)](#)). Er bekommt dadurch die dort definierten Rechte an einer oder mehreren [Elementgruppen](#). Details zur Rechtevergabe befinden sich im Abschnitt [Definition von Berechtigungen](#).

Ein Benutzer wird mit der Anweisung [UrmCreate\(0, UrmTypeUser, ...\)](#) erzeugt. Ein vorhandener Benutzer kann mit der Anweisung [UrmOpen\(0, UrmTypeUser, ...\)](#) geöffnet werden. Nach dem Öffnen des Benutzers steht ein Deskriptor zur Verfügung, mit dem auf die [Eigenschaften](#) des Benutzers zugegriffen werden kann. Die Möglichkeiten stehen zur Laufzeit und zur Entwurfszeit zur Verfügung (siehe auch [Verwaltung der Benutzer](#)).

Beispiel

```
// Aktiviert den Benutzer Sales_Temp für drei Monate
HdlUser # UrmOpen(_UrmTypeUser, _UrmLock, 'S
```

Kontakt

UserGroup

 **Objekt der Benutzergruppe in der Benutzerverwaltung**

[Liste](#),

Siehe [Eigenschaften](#),

[Benutzbefehle](#)

Dieses Objekt entspricht einer Benutzergruppe in der Datenbank. Die Benutzergruppe verfügt über eine Liste von Benutzern (User-Objekte) die Mitglieder dieser Gruppe sind. In einer Benutzergruppe können auch Rechte für Elementgruppen definiert werden. Die effektiven Rechte eines Benutzer an einer Elementgruppe entstehen aus der Kombination aller Rechte der Benutzergruppen, in denen Benutzer Mitglied ist. Details zur Rechtevergabe befinden sich im Abschnitt Definition von Berechtigungen.

In einer leeren Datenbank sind bereits die beiden Benutzergruppen _Administrators und _Everyone definiert.

Standardmäßig sind in der Benutzergruppe _Everyone alle vordefinierten Elementgruppen mit Ausnahme von _user und _usergroup mit den Rechten zum Erzeugen, Lesen, Ändern, Löschen, Ausführen und Konfigurieren eingetragen. Soll einer anderen Benutzergruppe die Ausführung aller Prozeduren erlaubt werden, muss der Benutzergruppe die Elementgruppe _procedure mit dem Recht [UrmPermExecute](#) zugewiesen werden.

Eine Benutzergruppe wird mit der Anweisung [UrmCreate\(0, _UrmTypeUserGroup, ...\)](#) erzeugt. Eine vorhandene Benutzergruppe kann mit der Anweisung [UrmOpen\(0, _UrmTypeUser, ...\)](#) geöffnet werden. Beim Öffnen der Benutzergruppe wird ein Deskriptor zurückgegeben, mit dem auf die Eigenschaften der Benutzergruppe zugegriffen werden kann. Die Rechte auf Elementgruppen können dann mit dem Befehl [UrmPermSet\(\)](#) festgelegt werden. Die Möglichkeiten stehen zur Laufzeit und zur Entwurfszeit zur Verfügung (siehe auch Verwaltung der Benutzergruppen).

Die Eigenschaften der Benutzergruppe _Administrators können mit Ausnahme der Eigenschaft [UrmPropComment](#) nicht verändert werden. Insbesondere können die Berechtigungen der Gruppe nicht geändert werden.

Beispiel

```
// Benutzergruppe erzeugenErg # UrmCreate(0, _UrmTypeUserGroup, 'UserGrpLogin');if (tErg = _ErrO
```

Kontakt

ElmGroup

 Elementgruppe der Benutzerverwaltung
[Liste](#),

Siehe [Eigenschaften](#),
[Benutzerbefehle](#)

Dieses Objekt beinhaltet eine Reihe von Einträgen. Jeder Eintrag repräsentiert entweder ein reales Element aus der Datenbank (ein Dialog, eine Prozedur, eine Datensatz-Tabelle usw.) oder ein durch den Programmierer definiertes virtuelles Objekt. Eine Elementgruppe fasst somit ein oder mehrere Datenbank- und/oder virtuelle Objekte zu einer Gruppe zusammen. Für die Gruppe können dann Rechte bei den [Benutzergruppen](#) definiert werden. Durch die Gruppenbildung erhält der Programmierer eine bessere Übersicht über das Rechtesystem.

Um einen Eintrag in eine Elementgruppe zu schreiben wird die Anweisung [UrmCreate\(\)](#) verwendet. Der Deskriptor der Elementgruppe und eine der [UrmTypeElm...-Konstanten](#) müssen zusammen mit dem Namen des Elements übergeben werden. Folgende Typen können angegeben werden:

UrmTypeElmCustom	Benutzerdefiniert / virtuelles Objekt
UrmTypeElmUser	Benutzer
UrmTypeElmUsergroup	Benutzergruppe
UrmTypeElmTable	Datei
UrmTypeElmDialog	Dialog
UrmTypeElmMenu	Menü
UrmTypeElmPicture	Bild im Bereich Picture
UrmTypeElmMetapicture	Bild im Bereich MetaPicture
UrmTypeElmPrintform	PrintForm -Objekt
UrmTypeElmPrintformlist	PrintFormList -Objekt
UrmTypeElmPrintdocument	PrintDoc -Objekt
UrmTypeElmPrintdorecord	PrintDocRecord -Objekt
UrmTypeElmProcedure	Prozedur
UrmTypeElmText	interner Text
UrmTypeElmBlob	Binäre Objekte in der BLOB-Verwaltung
UrmTypeElmTheme	Theme -Objekt

Um zum Beispiel einen Dialog in eine Elementgruppe aufzunehmen, muss ein Eintrag für den Typ (in diesem Fall [UrmTypeElmDialog](#)) in die Elementgruppe vorgenommen werden. Als (alpha2) wird der Name des Dialogs 'Beispiel1' übergeben.

Über den Typ [UrmTypeElmCustom](#) können Einträge durch den Programmierer vorgenommen werden: 'ModulArticle'. Die Rechte auf ein Element können in der Programmierung mit den Anweisungen [UrmPermElementGet\(\)](#) oder [UrmPermElementGetRaw\(\)](#) ermittelt werden.

Eine Elementgruppe wird mit der Anweisung [UrmCreate\(0, UrmTypeElmGroup, ...\)](#) erzeugt. Eine vorhandene Elementgruppe kann mit der Anweisung [UrmOpen\(0, UrmTypeElmGroup, ...\)](#) geöffnet werden. Beim Öffnen wird ein Deskriptor zurückgegeben, mit dem auf die [Eigenschaften](#) der Elementgruppe zugegriffen

Kontakt

werden kann. Ebenso können weitere Objekte der Elementgruppe hinzugefügt oder aus der Gruppe entfernt werden. Die Möglichkeiten stehen zur Laufzeit und zur Entwurfszeit zur Verfügung (siehe auch Verwaltung der Elementgruppen).

Beispiel

```
// Elementgruppe erzeugen tErg # UrmCreate(0, _UrmTypeElmGroup, 'ElmGrpLoginExamples');if (tErg ==
```

Kontakt

Storage-Objekte

Zugriff auf Applikationsressourcen

Siehe [Liste](#),
[Eigenschaften](#),

Befehle für

Die vom Programmierer angelegten Ressourcen einer CONZEPT 16-Applikation werden zusammen mit allen anderen Daten in der Datenbank abgelegt. Die Dialoge, Menüs, Druck-Objekte usw. werden als sogenannte Storage-Objekte gespeichert.

Die Objekte werden in einer Verzeichnisstruktur in der Datenbank gespeichert. Der Zugriff auf diese Objekte ist auch auf mit [DbaConnect\(\)](#) verbundenen Datenbanken möglich.

Der Befehl [StoDirOpen\(\)](#) liefert einen Deskriptor eines Verzeichnisses zurück. Dieser Deskriptor muss angegeben werden, wenn das Verzeichnis gelesen werden soll. Unterhalb des Wurzelverzeichnisses gibt es folgende Verzeichnisse:

- Dialog
Verzeichnis der Dialog-Objekte
- Menu
Verzeichnis der Menü-Objekte
- PrintForm
Verzeichnis der PrintForms
- PrintFormList Verzeichnis der Drucklisten
- PrintDocument
Verzeichnis der Druckdokumente
- PrintDocRecord Verzeichnis der Drucktabellen
- Picture
Verzeichnis der Raster- und Kachelgrafiken
- MetaPicture
Verzeichnis der Vektorgrafiken

Um eine Liste von Dialogen zu erstellen muss das entsprechende Verzeichnis geöffnet werden:

```
tHdlDir # StoDirOpen(0, 'Dialog');
```

In dem Verzeichnis kann anschließend mit der Anweisung [StoDirRead\(\)](#) gelesen werden. Wird das Verzeichnis nicht mehr benötigt, kann mit dem Befehl [StoClose\(\)](#) das Verzeichnis wieder geschlossen werden.

Kontakt

```
tHd1Dir->StoClose();
```

Über den Deskriptor können verschiedene Eigenschaften eines Verzeichnisses abgefragt werden:

ID ID des Storage-Objektes in der Datenbank

Name Name des Storage-Objekts

FullName Vollständiger Name (inkl. Pfadangabe)

Innerhalb der Verzeichnisse sind die entsprechenden Objekte abgelegt. Die Objekte können mit der Anweisung StoOpen() geöffnet werden. Anschließend können die Eigenschaften des Objekts ausgewertet werden:

ID ID des Storage-Objektes in der Datenbank

Name Name des Storage-Objekts

Type Typ des Storage-Objekts

FullName Vollständiger Name (inkl. Pfadangabe)

Created Zeitpunkt der Objekt-Erstellung

Modified Zeitpunkt der letzten Änderung des Objekts

CreatedUser Datenbank-Benutzer, der das Objekt erstellt hat

ModifiedUser Datenbank-Benutzer, der das Objekt zuletzt geändert hat

SizeDba Größe des Objektes in der Datenbank in Byte

SizeOrg Original-Größe des Objektes in Byte

Unicode Unicode-Unterstützung des Objekts

TypeFile Dateityp des Objekts

Die Eigenschaften der Storage-Objekte und Verzeichnisse können nur gelesen und nicht verändert werden.

Kontakt

Binäre Objekte

Große binäre Objekte (BLObs - Binary Large Objects)

Liste,
Eigenschaften,
Siehe Befehle für
binäre
Objekte, Blog

In binären Objekten können größere Datenmengen gespeichert werden, die nicht in einzelnen Datensätzen gespeichert werden können. Dies umfasst zum Beispiel Word-Dokumente, MPeg-Dateien oder Bilder.

Diese binären Objekte werden in einer Verzeichnisstruktur in der Datenbank gespeichert. Das Root-Verzeichnis kann nicht verändert oder gelöscht werden. Hier können neue Verzeichnisse mit dem Befehl BinDirOpen(0, <Name>, _BinCreate) angelegt werden. Dateien können nur in angelegten Verzeichnissen und nicht im Root-Verzeichnis gespeichert werden.

Der Befehl BinDirOpen() liefert einen Deskriptor eines Verzeichnisses zurück. Dieser Deskriptor muss angegeben werden, wenn ein neues binäres Objekt (Verzeichnis oder Datenobjekt) in diesem Verzeichnis angelegt werden soll.

Beispiel:

```
tHd1Dir      # BinDirOpen(0, 'Documents', _BinCreate); tHd1SubDir # tHd1Dir->BinDirOpen('Excel', _Bi
```

In diesem Beispiel wird ein neues Verzeichnis "Documents" in dem Root-Verzeichnis angelegt. In dem Verzeichnis "Documents" wird anschließend ein weiteres Verzeichnis mit dem Namen "Excel" erzeugt. Das zweite Verzeichnis könnte ebenfalls mit dem Befehl tHd1SubDir # BinDirOpen(0, 'Documents\Excel', _BinCreate); erzeugt werden.

Nicht mehr benötigte Deskriptoren können mit dem Befehl BinClose() entfernt werden. Sollten nur die oben angegebenen Verzeichnisse erzeugt werden, können jetzt die Deskriptoren freigegeben werden:

```
tHd1Dir  ->BinClose(); tHd1SubDir->BinClose();
```

Über den Deskriptor können verschiedene Eigenschaften eines Verzeichnisses abgefragt oder gesetzt werden:

- ID - Identität des Verzeichnisses
- Name - Name des Verzeichnisses
- FullName - Name und Pfad des Verzeichnisses
- Custom - Benutzerdefinierte Eigenschaft

Die Verzeichnisstruktur dient zur Ordnung der eigentlichen Datenobjekte. Ein neues Datenobjekt wird mit dem Befehl BinOpen(<Directory>, <Name>, _BinCreate) angelegt. Dieses Objekt ist zunächst leer. Mit dem zurückgegebenen Objektdeskriptor kann über den Befehl BinImport() eine externe Datei eingelesen und in der Datenbank gespeichert werden.

Beispiel:

Kontakt

```
tHdlBin # BinOpen(0, 'Documents\Excel\Tab1', _BinCreate); tHdlBin->BinImport('C:\Table\Table1.xls')
```

Bestehende Objekte können ebenfalls wieder exportiert werden.

Beispiel:

```
tHdlBin # BinOpen(0, 'Documents\Excel\Tab1', _BinSingleLock); tHdlBin->BinExport('C:\doc\Table1.xls')
```

In beiden Fällen bleibt das Original erhalten. Über den Deskriptor können verschiedene Eigenschaften eines binären Objektes abgefragt oder gesetzt werden:

- ID - Identität des Objektes
- Name - Name des Objektes
- FullName - Name und Pfad des Objektes
- Custom - Benutzerdefinierte Eigenschaft
- Created - Erstellungszeitpunkt des Objektes
- CreatedUser - Benutzer, der das Objekt erzeugt hat
- Modified - Zeitpunkt der letzten Änderung
- ModifiedUser - Benutzer, der das Objekt zuletzt geändert hat
- TypeUser - Benutzerdefinierte Typinformation
- TypeMime - MIME-Typ des Objektes
- SizeDba - Speicherverbrauch des Objektes in der Datenbank
- SizeDba64 - Speicherverbrauch des Objektes in der Datenbank (als 64-Bit-Wert)
- SizeOrg - Originalgröße des Objektes
- SizeOrg64 - Originalgröße des Objektes (als 64-Bit-Wert)
- TimeExternal - Datum und Uhrzeit der letzten Änderung der externen Datei vor Import
- Compression - Kompressionsstufe

Neben den hier gezeigten Funktionen stehen weitere Befehle zum Bearbeiten von binären Objekten zur Verfügung. Eine Liste der Befehle befindet sich im Abschnitt Befehle für binäre Objekte.

Binäre Objekte der Formate BMP, JPG und TIFF können in dem Objekt Picture dargestellt werden. In der Eigenschaft Caption wird '>0' gefolgt von dem Pfad und dem Name des binären Objektes angegeben. Entsprechendes gilt für binäre Objekte, in den Formaten, die von den Objekten MetaPicture, PrtPicture und PrtMetaPicture dargestellt werden können.

Befindet sich das binäre Objekt in einer mit DbaConnect() verbundenen CONZEPT 16-Datenbank, wird hinter dem > die Nummer des dort übergebenen Dateinummernkreises ('>2', '>3' oder '>4') angegeben.

Ist das binäre Objekt verschlüsselt in der Datenbank gespeichert, muss der Schlüssel in der Eigenschaft CryptKey angegeben werden.

Beispiele:

```
'>0\Pictures\Photo ''>2\Pictures\Archive'
```

Binäre Objekte können direkt in die folgenden Objekte geladen werden:

Kontakt

Oberflächen-Objekte Eigenschaft / Funktion

<u>DocView</u>	<u>FileName</u>
<u>Picture</u>	<u>Caption</u>
<u>MetaPicture</u>	<u>Caption</u>
<u>RtfEdit</u>	<u>StreamSource = WinStreamNameBin und FileName</u>
<u>RtfEdit</u>	<u>WinRtfPicInsertName()</u>
<u>PrtJobPreview</u>	<u>Caption</u>

Druck-Objekte

<u>PrtPicture</u>	<u>Eigenschaft</u>
<u>PrtMetaPicture</u>	<u>Caption</u>
<u>PrtPdf</u>	<u>Caption</u>
<u>PrtRtf</u>	<u>FileName</u>
	<u>StreamSource = WinStreamNameBin und FileName</u>

System-Objekte

<u>Memory</u>	<u>Funktion</u>
	<u>BinReadMem()</u>

Objekte für dynamische Strukturen

Dynamische Listen und Bäume

Liste,

Eigenschaften
der Elemente,

Siehe

Befehle für
dynamische

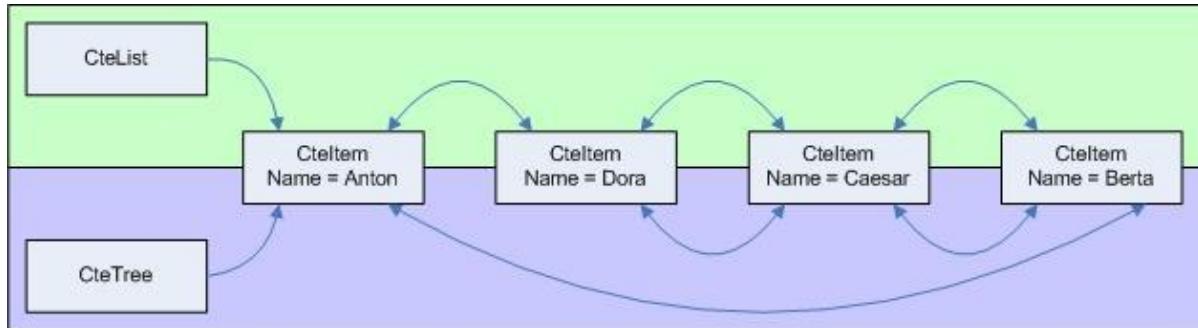
Innerhalb von CONZEPT 16 können sogenannte dynamische Strukturen erzeugt werden. Diese Strukturen werden im Arbeitsspeicher des Clients erzeugt und verarbeitet und ermöglichen so das Generieren von temporären Mengen, die effizient durchsucht oder sortiert werden können.

Für diese Strukturen stehen vier verschiedene Objekte zur Verfügung:

- CteItem-Objekt
- CteList-Objekt
- CteTree-Objekt
- CteNode-Objekt

CteItem-Objekt

CteItem-Objekte können als Informationsträger in einer Listen-Struktur in verketteten (CteList) und/oder sortierten Listen (CteTree) organisiert und verarbeitet werden:



Jedes CteItem-Objekt kann gleichzeitig in einem CteList-Objekt und einem CteTree-Objekt enthalten sein.

Das CteItem-Objekt verfügt über folgende Eigenschaften:

- Name
- ID
- Custom

CteList-Objekt

CteList-Objekte dienen zur Organisation von Elementen in einer verketteten Liste. Als Elemente können beliebige Deskriptoren (außer den Deskriptoren von Storage- und Binären Objekten) verwendet werden. Die Elemente sind nach ihrer Position sortiert.

Kontakt

Das CteList-Objekt ist für die sequenzielle Verarbeitung der darin enthaltenen Elemente ausgelegt. Die Liste kann mit der Anweisung CteRead() mit den Parametern CteFirst, CtePrev, CteNext und CteLast durchsucht werden. Es ist auch eine Suche über den Namen des Objekts möglich, kann aber in einem CteTree-Objekt wesentlich schneller durchgeführt werden.

Das CteList-Objekt verfügt über folgende Eigenschaften:

- Name
- ID
- Custom

CteTree-Objekt

CteTree-Objekte dienen zur Organisation von Elementen in einer sortierten Liste. Als Elemente können beliebige Deskriptoren (außer den Deskriptoren von Storage- und Binären Objekten) verwendet werden. Die Elemente sind nach ihrem Namen sortiert.



Innerhalb eines CteTree-Objekts darf der Name der Elemente nicht leer und muss eindeutig sein.

Das CteTree-Objekt ist für die Suche über den Namen der darin enthaltenen Elemente ausgelegt. Eine sequenzielle Verarbeitung kann in einem CteList-Objekt geringfügig schneller durchgeführt werden.

Das CteTree-Objekt verfügt über folgende Eigenschaften:

- Name
- ID
- Custom

Für das CteTree-Objekt wird der sogenannte bichromate Baumtyp verwendet (auch als rot-schwarz-Baum bezeichnet), bei dem es sich um einen ausgeglichenen binären Baum handelt.

CteNode-Objekt

CteNode-Objekte können als Informationsträger in einer Baum-Struktur organisiert und verarbeitet werden.

Jedes CteNode-Objekt kann weitere CteNode-Objekte als untergeordnete Knoten (Child) und als Attributknoten (Attrib) enthalten. Die untergeordneten Objekte werden in eigenen Listen verwaltet. Alle Kinder eines Baumknotens befinden sich in der Child-Liste, alle Attribute in der Attribute-Liste. Da alle untergeordnete Objekte wiederum CteNode-Objekte sind, haben diese ebenfalls eine Child- und eine Attribute-Liste.



CteNode-Objekte können sowohl verkettet als auch sortiert organisiert und verarbeitet werden. Dazu kann in jedem CteNode-Objekt eine verkettete Liste und/oder eine sortierte Liste für untergeordneten Knoten und Attributknoten

Kontakt

erzeugt werden.

Das CteNode-Objekt verfügt über folgende Eigenschaften:

- Name
- ID
- Custom
- Flags
- Parent
- Type
- Value...
- AttribCount
- ChildCount

Die Eigenschaften Type und Value... beinhalten die Knoteninformation, wobei die Eigenschaften Name, ID und Custom ebenfalls als Informationsträger verwendet werden können. Bei der Verwendung im Zusammenhang mit XML- und JSON-Daten wird die Eigenschaft ID zur Identifizierung des Knotentyps verwendet.

Mit der Anweisung CteRead() kann nicht die komplette Baumstruktur durchsucht werden. Die Anweisung durchsucht immer nur eine Ebene, entweder die Liste der Knoten in der nächsten Ebene (Child-List) oder die Liste der Attribute (Attribute-List) (siehe Beispiel - Durchsuchen einer XML-Struktur).

Anwendungsbeispiele

1. Temporäre Menge von Datensätzen

Soll eine noch nicht näher bestimmte Anzahl von Datensätzen zur Weiterverarbeitung markiert werden, kann die Datensatz-ID oder der eindeutige Schlüsselwert des Datensatzes in eine Liste oder einen Baum eingefügt werden. Die Verarbeitung der Datensätze kann dann über die dynamische Struktur erfolgen.

Die entsprechenden Datensätze können ebenfalls in erzeugten Datensatzpuffern (RecBufCreate()) gespeichert und in einer Liste oder einer sortierten Liste organisiert werden.

2. Beliebige temporäre Mengen

Alle Daten, die aus mehreren Elementen bestehen, können mit Hilfe der dynamischen Strukturen zu temporären Mengen zusammengefasst werden. Sind die Datenmengen in den einzelnen Elementen so groß, dass sie nicht mehr in einem CteItem-Objekt gespeichert werden können, können Deskriptoren von globalen Datenbereichen, internen Texten usw. mit Hilfe der Anweisung HdlLink() an die CteItem-Objekte angehängt oder direkt mit der Anweisung CteInsert() in eine Liste eingefügt werden.

3. Strukturierte Informationen

Liegen die Informationen in einer Baumstruktur vor, wie zum Beispiel bei einem XML-Dokument, kann diese Struktur mit Hilfe der CteNode-Objekte nachgebildet werden.

Kontakt

4. Variable Übergabe von Parametern an Funktionen

Sollen einer Funktion bei jedem Aufruf eine andere Anzahl von Argumenten übergeben werden, können diese Argumente in einer Liste zusammengefasst und nur der Deskriptor der Liste der Funktion übergeben werden. Die übergebenen Listenelemente funktionieren in diesem Fall wie var-Parameter, da Änderungen in den CteItems ebenfalls in der aufrufenden Funktion zur Verfügung stehen. Sind die Elemente in einem CteTree organisiert, kann über den Namen des Arguments auf dessen Wert zugegriffen werden.

Kontakt

Memory-Objekt



Objekt innerhalb des Speichers zur Verarbeitung beliebiger Daten

Siehe [Befehle](#),

[Eigenschaften](#)

Ein Memory-Objekt ist ein durch den Benutzer angelegter Speicherbereich im Hauptspeicher, der für beliebige binäre Daten verwendet werden kann. Der Speicherbereich besteht aus einer benutzerdefinierten Anzahl von n Bytes, wobei das erste Byte die Position 1 und das letzte Byte die Position n hat.

Das Objekt wird mit der Anweisung [MemAllocate\(\)](#) angelegt.

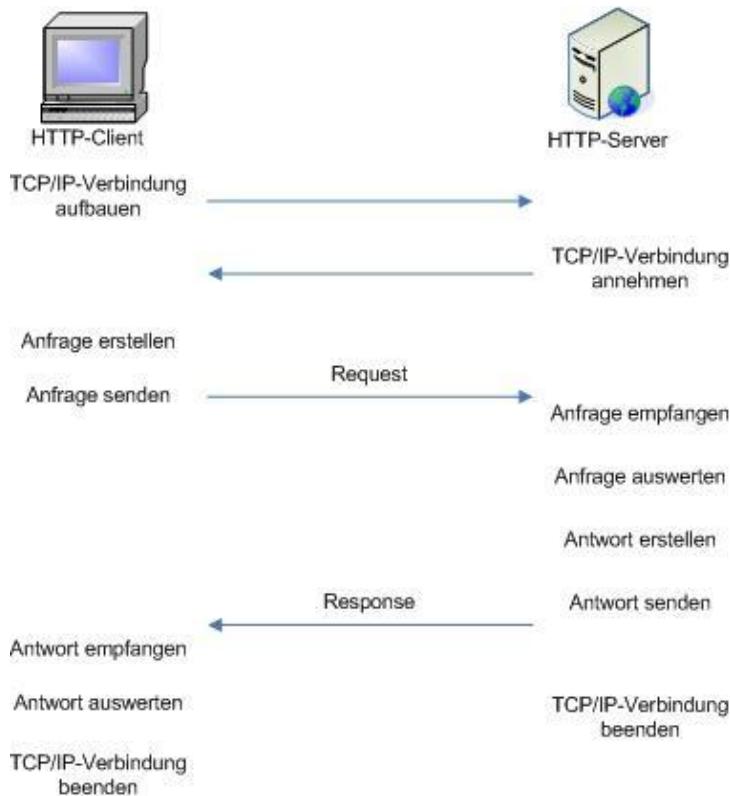
HTTP

Beschreibung des HTTP-Objekts

[Liste](#),
[Eigenschaften](#),

Siehe [HttpOpen\(\)](#),
[Beispiel](#),
[Kommunikation im Web \(Blog\)](#)

Das HTTP-Ojekt dient als Grundlage für eine einfache Nutzung des HTTP-Protokolls. Das Objekt ist sowohl auf der Clientseite als auch auf der Serverseite der HTTP-Kommunikation verwendbar. Eine einfache HTTP-Sitzung besteht aus folgenden Schritten:



Auf TCP/IP-Ebene wird die Kommunikation über Sockets ([SckConnect\(\)](#)) vorgenommen. Auf der Ebene der Anfragen und Antworten werden HTTP-Objekte eingesetzt. Diese können über den Befehl [HttpOpen\(\)](#) in vier verschiedenen Varianten erzeugt werden:

- [HttpSendRequest](#) Erstellen und Versenden einer Anfrage
- [HttpRecvRequest](#) Empfangen und Auswerten einer Anfrage
- [HttpSendResponse](#) Erstellen und Versenden einer Antwort
- [HttpRecvResponse](#) Empfangen und Auswerten einer Antwort

Die HTTP-Objekte verfügen je nach Typ über die folgenden Eigenschaften (r = nur lesen, rw = lesen und schreiben, - = nicht vorhanden):

[HttpRecvRequest](#) [HttpSendResponse](#) [HttpSendRequest](#) [HttpRecvRes](#)

Kontakt

<u>Protocol</u>	r	rw	rw	r
<u>Method</u>	r	-	rw	-
<u>StatusCode</u>	-	rw	-	r
<u>URI</u>	r	-	rw	-
<u>HostName</u>	r	-	rw	-
<u>ContentLength</u>	r	r	r	r
<u>ProxyAuthorization</u>	-	-	rw	-
<u>ProxyAuthType</u>	-	-	rw	-
<u>HttpHeader</u>	r	r	r	r
<u>HttpParameters</u>	r	r	r	r

PDF

Beschreibung des PDF-Objekts

[Liste](#),

Siehe [Eigenschaften](#),

[PdfNew\(\)](#),

[PdfOpen\(\)](#)

Mit diesem Objekt können PDF-Dokumente angelegt und bestehende PDF-Dokumente bearbeitet werden. Das Objekt wird mit der Anweisung [PdfNew\(\)](#) bzw. [PdfOpen\(\)](#) angelegt.

Alle Änderungen an dem Objekt sind nur temporär, bis das Objekt mit der Anweisung [PdfClose\(\)](#) gespeichert und geschlossen wird.

Chart



Diagramm

[Befehle](#),

Siehe [Eigenschaften](#),

[ChartData](#),

[Blog](#)

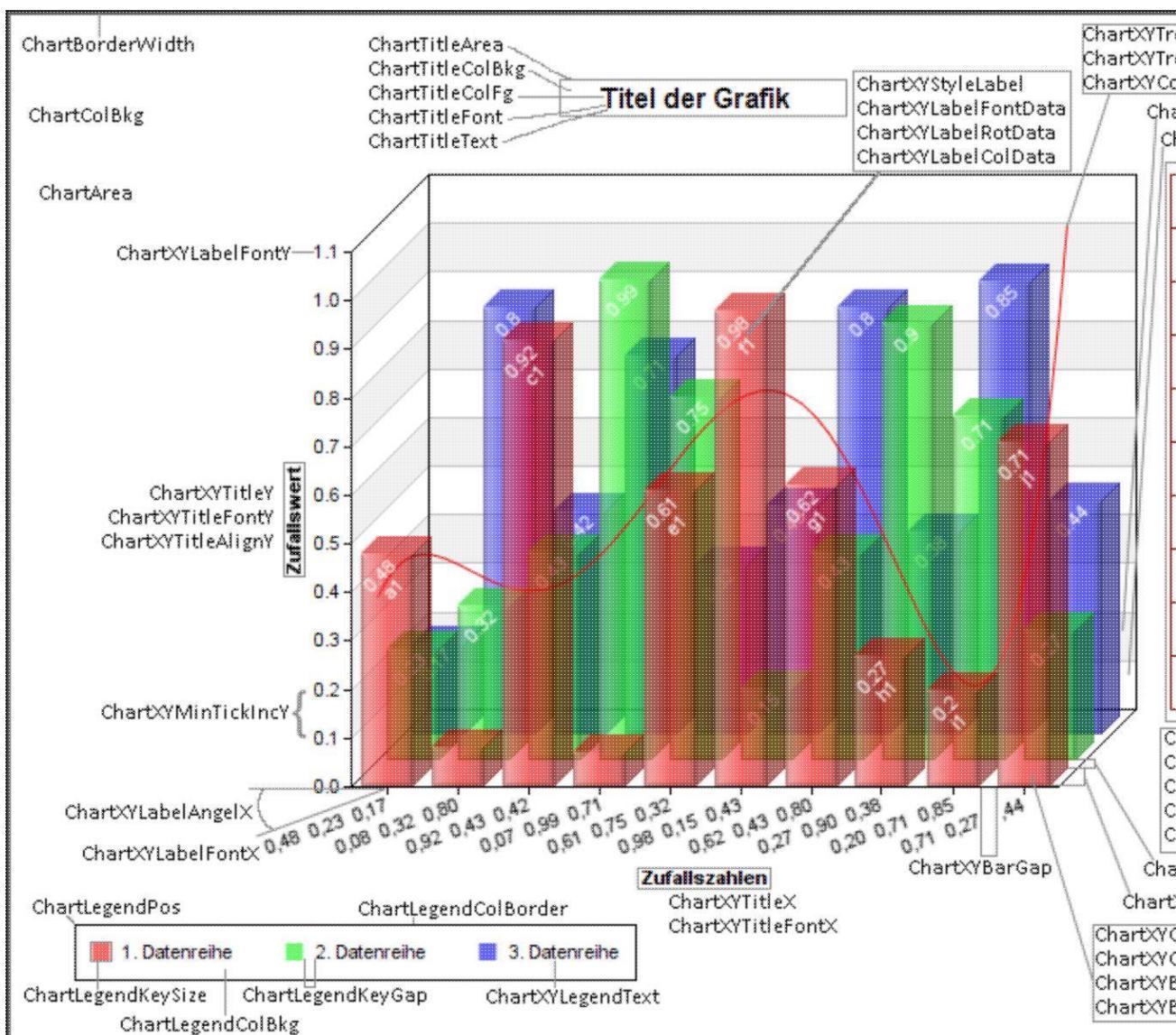
Im Chart-Objekt werden alle Informationen zur Erzeugung eines Diagramms gespeichert. Das Objekt wird mit der Anweisung [ChartOpen\(\)](#) erzeugt. Dabei wird der Diagramm-Typ, die Ausgabegröße und der Titel des Diagramms angegeben. Die Ausgabegröße und der Titel können ebenfalls über die entsprechenden Eigenschaften gesetzt werden.

Die Eigenschaften des Objekts bestimmen die Darstellung der einzelnen Datenreihen (siehe [ChartData](#)). Bei den Eigenschaften ist vermerkt, ob diese für jede Datenreihe separat gesetzt werden kann.

Die Eigenschaften des Objekts teilen sich in mehrere Bereiche auf. Die Chart...-Eigenschaften können für alle Diagramm-Typen gesetzt werden. Die ChartXY...-Eigenschaften wirken sich nur bei Koordinaten-Diagrammen ([ChartXY](#)), die ChartPie...-Eigenschaften nur bei Torten-Diagrammen ([ChartPie](#)), die ChartPyramid...-Eigenschaften nur bei Pyramiden-Diagrammen ([ChartPyramid](#)) und die ChartSurface...-Eigenschaften nur bei Oberflächen-Diagrammen ([ChartSurface](#)) aus.

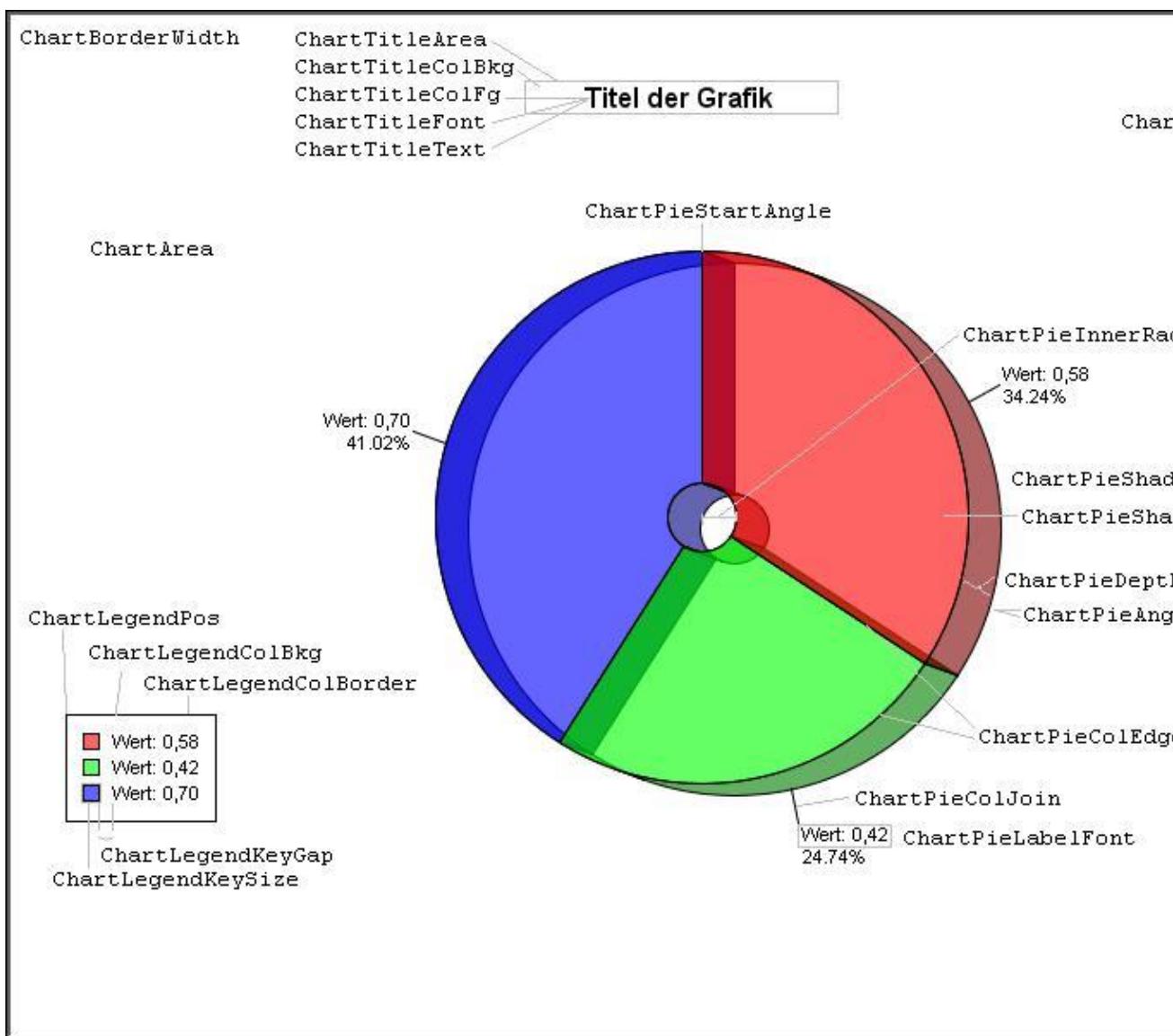
- Eigenschaften eines Koordinaten-Diagramms

Kontakt



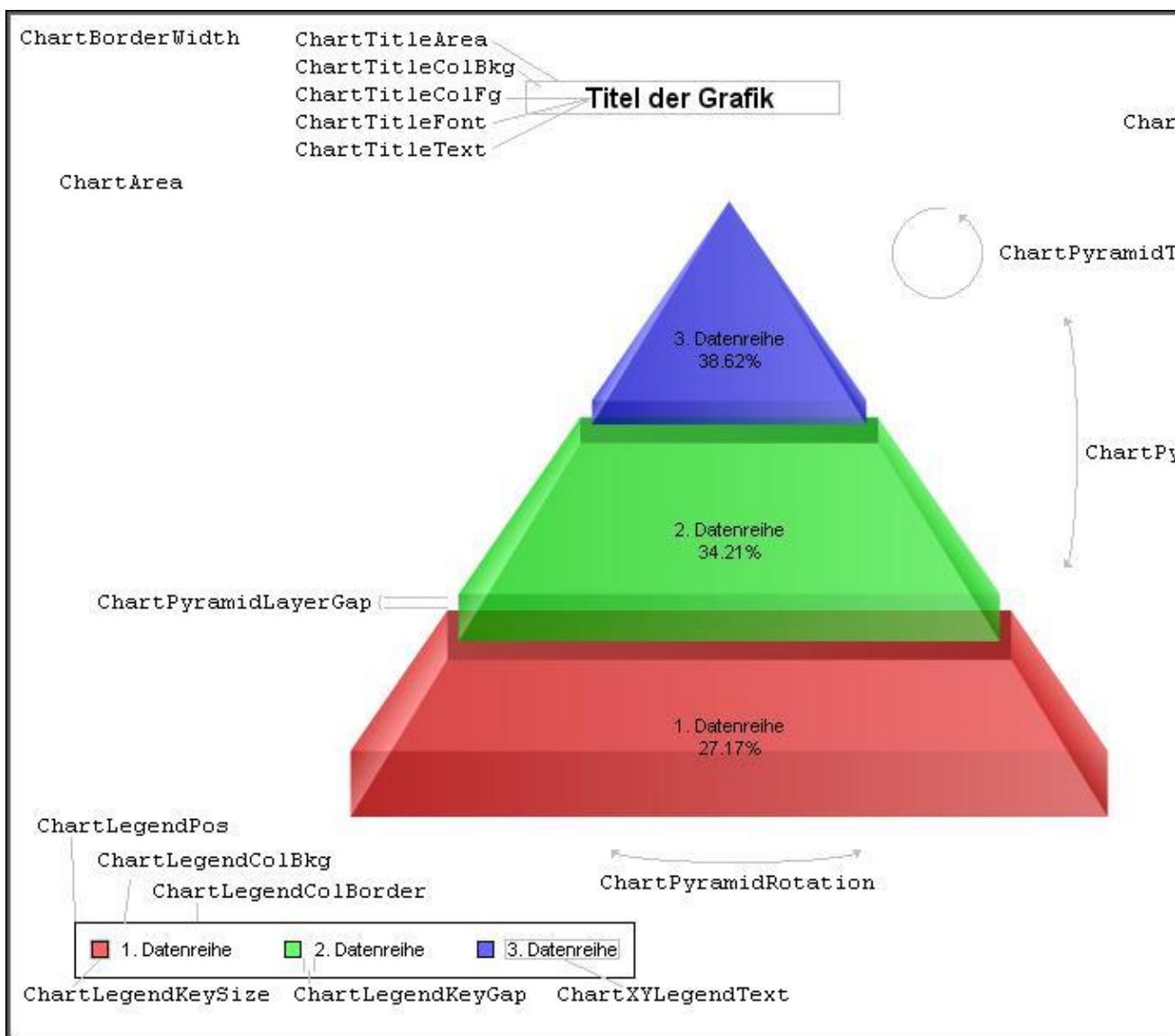
- Eigenschaften eines Tortendiagramms

Kontakt



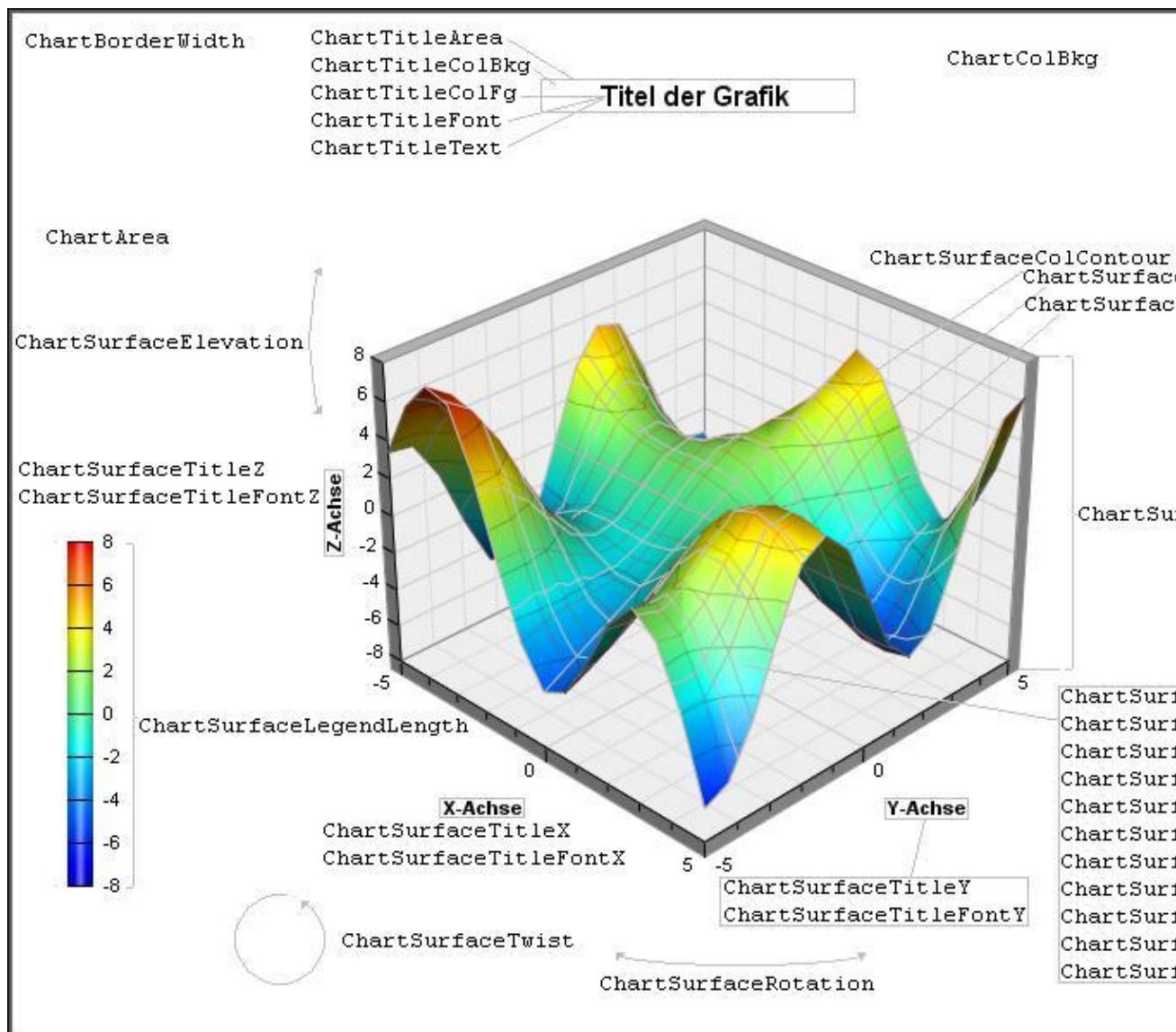
- Eigenschaften eines Pyramiden-Diagramms

Kontakt



- Eigenschaften eines Oberflächen-Diagramms

Kontakt



Das Diagramm wird erst erzeugt, wenn die Anweisung ChartSave() ausgeführt wird. Die Speicherung des Diagramms kann als externe Datei oder in einem Memory-Objekt erfolgen. Anschließend kann das Objekt mit ChartClose() geschlossen werden.

ChartData

 Darzustellende Daten in einem Diagramm

Befehle [Chart](#),

Das ChartData-Objekt enthält die Daten, die in einem [Chart](#)-Objekt (Diagramm) dargestellt werden sollen. Das Objekt wird mit einer [ChartDataOpen\(\)](#)-Anweisung geöffnet und kann anschließend mit einer oder mehreren [ChartDataAdd\(\)](#) Anweisungen mit Werten, Bezeichnern und Farben gefüllt werden.

Bei der Anzeige in einem Koordinaten-Diagramm können mehrere ChartData-Objekte erzeugt und dem [Chart](#)-Objekt mit der Anweisung [ChartDataClose\(\)](#) hinzugefügt werden. Die Daten werden dann in Reihen angezeigt. Sollen die Daten einer Reihe sortiert angezeigt werden, kann zuvor die Anweisung [ChartDataSort\(\)](#) aufgerufen werden.

Zur Anzeige der Datenreihe werden die Inhalte der Eigenschaften des [Chart](#)-Objekts verwendet. Sollen unterschiedliche Anzeigen verwendet werden (zum Beispiel unterschiedliche Farben) können die betreffenden Eigenschaften vor der Speicherung durch [ChartDataClose\(\)](#) angepasst werden. Die Eigenschaften, die geändert werden können sind mit entsprechenden Hinweisen versehen.

Das ChartData-Objekt selbst verfügt nur über die Eigenschaften [Name](#), [ID](#) und [Custom](#).

Kontakt

TapiDevice



TAPI-Gerätetreiber

Liste,

Siehe Eigenschaften,

Befehle

Bei den TAPI-Gerätetreibern handelt es sich um installierte Telefonieanbieter auf dem System. Der Befehl TapiOpen() liefert eine Liste der installierten Gerätetreiber. Der Gerätetreiber muss bei den Befehlen TapiDial() und TapiListen() angegeben werden.

Selection



Selektions-Objekt
[Liste](#),
[Eigenschaften](#),
Siehe [SelCreate\(\)](#),
[Dynamische Selektionen](#)
[\(Blog\)](#)

In einem Selection-Objekt wird eine Selektion zusammengestellt.

Eine Selektion besitzt immer eine Ausgangsdatei und mindestens ein Selektionskriterium. Das Selektionskriterium ist ein Ausdruck mit einem logischen Ergebnis. Es bestimmt, welche Datensätze der Ausgangsdatei in die Selektionsmenge übernommen werden. Neben der Ausgangsdatei können über Verknüpfungen weitere Dateien angehängt werden. Für jede dieser Anhangdateien kann wiederum ein Selektionskriterium definiert werden.

Beispiel:

Aus der Kunden-Datei sollen alle Adressen in einem bestimmten Postleitzahl-Bereich ermittelt werden. Als Ausgangsdatei wird die Kunden-Datei verwendet, das Selektionskriterium ist "Länderkennzeichen ist gleich "D" UND Postleitzahl ist im Bereich 60000 bis 69999". Mit den Befehlen für dynamische Selektionen wird die Selektion wie folgt erstellt:

```
tSel # SelCreate(tblCstCustomer, keyCstId); tSel->SelDefQuery('' , 'faCstLkz = ''D'' AND faCstPlz >
```

Nachdem die Selektion definiert wurde, kann sie ausgeführt werden. Dazu wird der Deskriptor zum Erstellen einer Selektion in einen Selektionspuffer umgewandelt (SelOpen()) und anschließend die Selektion mit der Anweisung SelRun() ausgeführt.

```
tSel # tSel->SelOpen(); tErg # tSel->SelRun(_SelDisplay | _SelWait);
```

Der Inhalt der Selektionsmenge enthält nur die Datensätze, die zum Zeitpunkt der Selektionsdurchführung das Selektionskriterium erfüllt haben. Spätere Änderungen im Datenbestand führen nicht dazu, dass Datensätze in die Menge aufgenommen oder aus der Menge entfernt werden. Dazu muss die Selektion erneut durchgeführt werden. Eine bestehende Selektion kann mit der Anweisung SelRead() in einen Selektionspuffer gelesen und anschließend durchgeführt werden.

Um auf die Datensätze einer Selektionsmenge zugreifen zu können, muss der Deskriptor des Selektionspuffers bei dem entsprechenden Befehl (zum Beispiel RecRead()) oder in einer Eigenschaft (zum Beispiel DbSelection beim RecList-Objekt) angegeben werden.

Wird die Selektionsmenge nicht mehr benötigt, kann sie geschlossen werden. Wird auch die Definition der Selektion nicht mehr benötigt, kann auch diese gelöscht werden.

```
tSel->SelClose(); SelDelete(tblCstCustomer, 'tmp.sel');
```

Kontakt

Neben Selektionen, die nur die Datensätze einer Datei untersuchen, können auch Selektionen definiert werden, die weitere Dateien durchsuchen. Existiert in der Ausgangsdatei eine Verknüpfung zu einer anderen Datei, kann diese andere Datei angehängt werden. In dieser angehängten Datei können Bedingungen für die verknüpften Datensätze angegeben werden. Sollen die Kunden weiter eingeschränkt werden auf die Kunden, die in einem bestimmten Zeitraum etwas bestellt haben, muss das Abfragekriterium entsprechend erweitert werden:

```
tSel # SelCreate(tblCstCustomer, keyCstId); tSel->SelAddLink(' ', tblOrhOrderheader, tblCstCustomer
```

In diesem Beispiel werden nur die Kunden-Datensätze in die Selektionsmenge übernommen, die in dem Postleitzahlengebiet 60000 ansässig sind und im Jahr 2007 mindestens eine Bestellung getätigten hatten. Die verknüpften Datensätze werden nicht mit in die Selektionsmenge aufgenommen. Dies kann erreicht werden, indem bei der Anweisung SelAddLink() der letzte Parameter auf SelResultSet gesetzt wird. Bei Angabe des Selektionsdeskriptors beim Befehl RecLink() werden dann auch nur Datensätze gelesen, die das Selektionskriterium erfüllen.

Weitere Informationen zum Aufbau von Selektionen befinden sich bei den entsprechenden Befehlen für dynamische Selektionen.

Kontakt

Validierungselemente

Elemente der Datenbankvalidierung

Siehe [Liste](#),

Validierungsbefehle

Mit der Datenbankvalidierung können Entwickler bei der Zertifizierung der Anwendung unterstützt werden. Hierbei können Datenbankelemente validiert werden. Die validierten Elemente sind in einer Verzeichnisstruktur nach Typ getrennt abgelegt.

Mit dem Befehl [VldDirOpen\(\)](#) kann ein Validierungsverzeichnis geöffnet werden.

Nicht mehr benötigte Deskriptoren können mit dem Befehl [VldClose\(\)](#) entfernt werden.

Über den Deskriptor können verschiedene Eigenschaften eines Verzeichnisses abgefragt oder gesetzt werden:

- ID - Identität des Verzeichnisses (nur lesbar)
- Name - Name des Verzeichnisses (nur lesbar)
- FullName - Name und Pfad des Verzeichnisses (nur lesbar)

Die Verzeichnisstruktur dient zur Ordnung der eigentlichen Validierungselemente. Ein neues Element wird mit dem Befehl [VldOpen\(<Directory>, <Name>, VldCreate | VldLock\)](#) angelegt und gesperrt. Mit dem zurückgegebenen Deskriptor können die Elementeigenschaften verändert werden. Folgende Eigenschaften sind vorhanden:

- ID - Identität des Validierungselementes (nur lesbar)
- Name - Name des Validierungselementes (nur lesbar)
- Modified - Zeitpunkt der letzten Änderung des Referenzelementes
- ModifiedUser - Benutzer, der die letzte Änderung am Referenzelement durchgeführt hat
- Validated - Zeitpunkt der letzten Änderung des Validierungselementes
- ValidatedUser - Benutzer, der die letzte Änderung am Validierungselement durchgeführt hat
- Flags - Status des Validierungselementes
- Custom - Benutzerdefinierte Eigenschaft (max. 250 Zeichen)
- Type - Typ des Validierungselementes (_VldType... / nur lesbar)
- Hash - Hashwert des Validierungselementes (max. 79 Zeichen)

Beispiel:

```
tVldDir # VldDirOpen(_VldTypeDialog);if (tVldDir > 0){ tVldElm # tVldDir->VldOpen('Frm_Main', _V
```

Neben den hier gezeigten Funktionen stehen weitere Befehle zum Bearbeiten von Validierungselementen zur Verfügung. Eine Liste der Befehle befindet sich im Abschnitt [Validierungsbefehle](#).

Kontakt

Zentrale Datenobjekte

Zentral verwaltetes Datenobjekt

System-Objekte,

Befehle für

Siehe zentrale

Datenobjekte,

Blog

Das zentrale Datenobjekt dient zur nicht persistenten Speicherung von Informationen, die von mehreren Clients benötigt werden. Die Speicherung erfolgt beim CONZEPT 16-Server. Die dort gespeicherten Daten bleiben entweder bis zur Abmeldung des Benutzers, der die Daten gespeichert hat oder bis zur Schließung der Datenbank erhalten.



Die Daten werden nicht in der Datenbank gespeichert.

Das Datenobjekt besteht aus einem Schlüssel- und einem Datenanteil. Der Schlüsselwert ist vom Typ alpha und darf maximal 2000 Zeichen lang sein. Der Datenanteil ist ebenfalls vom Typ alpha, allerdings mit einer Maximallänge von 4096 Zeichen. Die Gesamtlänge (also Schlüssel- und Datenanteil) dürfen jedoch 4601 Byte nicht überschreiten.

Es können beliebig viele Datenobjekte angelegt werden bis zu einem

Gesamtspeicherverbrauch von 64 MB.

Drag & Drop-Objekte

Beschreibung der Drag & Drop-Objekte

Drag &
Drop-Ereignisse,
Blog,

Siehe

Prozedurbeispiel in
der

Folgende Objekte werden bei Drag & Drop-Operationen verwendet:

- DataPlace
- DragData
- DragDataFormat

Die Objekte werden vom System angelegt und nach den entsprechenden Drag & Drop-Ereignissen auch wieder entfernt.

Das DragData-Objekt wird bei den Ereignissen EvtDragInit, EvtDragTerm, EvtDropEnter und EvtDrop übergeben. Je nach Ereignis muss das Objekt unterschiedlich behandelt werden:

- EvtDragInit

Das DragData-Objekt muss mit Informationen gefüllt werden. Die Informationen über den Datenbank-Benutzer und die Datenbank selber sind dabei bereits vorhanden. Die bei der Drag & Drop-Operation zu übertragenden Daten müssen allerdings noch zusammengestellt werden.

Zunächst muss das Format der Daten bestimmt und die entsprechende Eigenschaft FormatEnum gesetzt werden. Es können auch mehrere Formate gleichzeitig übertragen werden. Für jedes Format steht dann ein DragDataFormat-Objekt zur Verfügung, dass mit den Daten gefüllt werden kann.

- EvtDragTerm

Da innerhalb des DragData-Objekts die Deskriptoren der Daten-Objekte zu erreichen sind, die nach dem Abschluss der Drag & Drop-Operation unter Umständen wieder freigegeben werden müssen, wird auch in diesem Ereignis das Objekt übergeben.

- EvtDropEnter

In dem DropTarget-Objekt muss zu diesem Zeitpunkt festgestellt werden, ob das Objekt die übergebenen Daten aufnehmen kann. Hier werden also die zu übertragenden Daten untersucht. Über den Effekt-Parameter wird dann zurückgegeben, ob das Objekt eine Drag & Drop-Operation zulässt und welche Operationen zugelassen sind.

- EvtDrop

Beim EvtDrop werden die Daten in das DropTarget-Objekt übernommen. Dazu wird das DragData-Objekt ausgewertet und die Daten entsprechend eingepflegt.

Kontakt

Das DragDataFormat-Objekt wird über das DragData-Objekt zur Verfügung gestellt.

Es ist somit in den gleichen Ereignissen erreichbar.

Dieses Objekt kann für jedes Format einmal zur Verfügung stehen. Je nachdem, welches Format angegeben wurde, befindet sich ein Deskriptor auf eine Cte-Liste, Cte-Tree oder einen Textpuffer in diesem Objekt. In dieser Struktur oder dem Text befinden sich dann die Daten, die übertragen werden sollen.

Das DataPlace-Objekt wird bei dem Ereignis EvtDrop übergeben. Es enthält weiterführende Informationen über das DragSource-Objekt, die gewünschte Position der übergebenen Daten und zusätzliche Argumente.

Die zusätzlichen Argumente sind abhängig vom DropTarget-Objekt und definieren dort eine genauere Position, wo das nicht mit den Konstanten der Eigenschaft DropPlace möglich ist.

Kontakt

DataPlace



Informationen zum Ziel bei Drag & Drop-Operationen

Siehe [Liste](#),

[Eigenschaften](#)

Das Objekt wird bei den Ereignissen [EvtDragInit](#), [EvtDropOver](#) und [EvtDrop](#) übergeben.

Bei [EvtDropOver](#) und [EvtDrop](#) enthält es Informationen darüber, wo der Inhalt platziert werden soll und aus welchem Objekt die Informationen stammen. Beim [EvtDragInit](#) können Informationen über das Quell-Objekt ermittelt werden.

Die Informationen können den entsprechenden [Eigenschaften](#) entnommen werden.

DragData



Daten zur Drag & Drop-Operation

Siehe [Liste](#),

Eigenschaften

In diesem Objekt befinden sich allgemeine Daten der Drag & Drop-Operation. Die Daten können über die entsprechenden Eigenschaften ausgelesen werden. Informationen zu den Inhalten (Datei, Text usw.), die an der Operation beteiligt sind, können über die Eigenschaft [Data](#) ermittelt werden.

In diesem Objekt werden Informationen über Datenbank, Benutzer und Datenformat abgelegt.

DragDataFormat



Format der Drag & Drop-Daten

Siehe [Liste](#),

Eigenschaften

Das DragDataFormat-Objekt enthält Daten eines festgelegten Formats. Hier können Informationen über das Format und den Inhalt der über Drag & Drop übertragenen Daten ermittelt werden.

Der Deskriptor dieses Objekts ist in der Eigenschaft [Data](#) des [DragData](#)-Objekts hinterlegt.

Der Deskriptor auf die übertragenen Daten befindet sich in der Eigenschaft [Data](#). Abhängig vom Format der Daten befindet sich in der Eigenschaft ein Deskriptor auf eine [CTE-Liste](#), einen [CTE-Tree](#) oder einen Textpuffer.

Kontakt

Objekte der Mehrfachselektion

Beschreibung der Objekte der Mehrfachselektion

SelectionData,

EvtLstSelectRange,

Siehe

Befehle der

Mehrfachselektion

Die Mehrfachselektion ist in folgenden Objekten möglich:

- WinComFileOpen
- WinComFileSave
- DataList
- RecList
- StoList
- TreeView

In den Common-Dialogen (WinComFileOpen und WinComFileSave) wird eine Mehrfachselektion über die Eigenschaften FileCount und FileName dargestellt. Damit mehrere Dateien selektiert werden können, muss die Ausprägung

WinComAllowMultiSelect in der Eigenschaft Flags gesetzt werden. FileCount gibt dann die Anzahl der selektierten Dateien an, die Namen der Dateien können über das Array FileName abgefragt werden.

Bei den anderen Objekten wird die Mehrfachselektion durch Setzen der Eigenschaft MultiSelect auf true aktiviert.

Erst wenn diese Eigenschaft gesetzt ist, steht ein SelectionData-Objekt zur Verfügung aus dem die Informationen über die selektierten Objekte entnommen werden können. Die selektierten Objekte können entweder direkt aus der Cte-Liste oder über die Befehle der Mehrfachselektion verarbeitet werden. Detaillierte Informationen zu dem Objekt befinden sich im Abschnitt SelectionData.

Wird innerhalb eines RecList-Objekts eine Mehrfachselektion mit Hilfe der  -Taste durchgeführt (Selektion eines Bereichs durch Anklicken des ersten und des letzten Listeneintrags des Bereichs), wird das Ereignis EvtLstSelectRange ausgelöst.

Kontakt

SelectionData

 **Selektionsobjekt der Mehrfachauswahl**

Eigenschaften,

Befehle der

Siehe Mehrfachselektion,

MultiSelect,

SelData

Ist bei einem Objekt eine Mehrfachauswahl zulässig, kann die Mehrfachauswahl durch Setzen der Eigenschaft MultiSelect bei diesem Objekt auf true aktiviert werden.

Erst wenn diese Eigenschaft gesetzt ist, steht das SelectionData-Objekt zur Verfügung. Um die Eigenschaften dieses Objekts zu lesen, wird der Deskriptor des Objekts benötigt. Der Deskriptor kann der Eigenschaft SelData entnommen werden.

Das SelectionData-Objekt besitzt folgende Eigenschaften:

Custom Benutzerdefinierte Eigenschaft

Data Deskriptor einer Cte-Liste und eines Cte-Trees

Die beiden Strukturen unterscheiden sich lediglich in der Sortierung der Elemente. In der Cte-Liste sind die selektierten Objekte in der Reihenfolge ihrer Selektion enthalten. Im Cte-Tree werden die Objekte über ihre Identifikation (Zeilennummer, Datensatz-ID oder Objekt-Deskriptor) sortiert.

Die Deskriptoren zu den verschiedenen Strukturen werden über Parameter beim Lesen oder Schreiben der Eigenschaft Data bestimmt:

```
tCteData # tSelectionData->wpData(_WinSelDataCteTree); // Cte-Tree tCteData # tSelectionData->wpDa
```

Den Eigenschaften der einzelnen Cte-Items können anschließend Informationen zu den selektierten Objekten entnommen werden:

- Custom

Die Eigenschaft wird nicht automatisch gesetzt und steht dem Entwickler zur Verfügung.

- ID

Die ID wird automatisch vorbelegt. Je nach zugrunde liegendem Objekt befindet sich in dieser Eigenschaft die Datensatz-ID (RecList-Objekt), die Zeilennummer (DataList-Objekt) oder der Deskriptor des TreeNode-Objekts (TreeView-Objekt). Beim StoList-Objekt ist die ID nicht belegt.

- Name

In dieser Eigenschaft befindet sich der in eine Zeichenkette umgewandelte Wert aus der Eigenschaft ID. Wird innerhalb eines StoList-Objekts mehrfach selektiert, befinden sich die Namen der selektierten Objekte in dieser Eigenschaft.

Über die Befehle WinMsdInsert() und WinMsdDelete() können Elemente der Selektion hinzugefügt bzw. aus einer Selektion entfernt werden.

Kontakt

Aufgrund der Unterschiede bei den zugrundeliegenden Objekten, sind bei der Verwendung der Mehrfachselektion folgende Dinge zu beachten:

- **RecList**

Beim **RecList**-Objekt können alle sichtbaren Datensätze selektiert werden. Werden vor oder nach der Selektion Datensätze gelöscht, wird das innerhalb der Selektion nicht berücksichtigt. Die Liste der selektierten Elemente kann also Elemente enthalten, die nicht mehr vorhanden sind.

Wird zusammen mit der  -Taste ein Bereich selektiert, wird nur der Anfang und das Ende des Bereichs angegeben (siehe auch Ereignis **EvtLstSelectRange**). Dazwischen können sich auch neu angelegte Datensätze befinden, die noch nicht in der Liste angezeigt wurden.

- **DataList**

Die Liste der selektierten Elemente wird beim **DataList**-Objekt automatisch gepflegt. Werden Zeilen aus dem **DataList**-Objekt entfernt, die in der Selektion enthalten sind, wird das entsprechende Element ebenfalls aus der Cte-Liste und dem Cte-Tree entfernt.

- **StoList**

Bei der Storage-List werden in den Listen des Selektions-Objekts nur die Namen der Objekte angegeben. Die Id wird nicht besetzt.

- **TreeView**

Bei Verwendung des **TreeView**-Objekts sollten bei Änderungen der Selektion die Befehle **WinMsdInsert()** und **WinMsdDelete()** verwendet werden, da so beide Selektions-Listen des Objekts automatisch miteinander synchronisiert werden. Ansonsten wird der Befehl **WinMsdUpdate()** verwendet, um die Selektion zu aktualisieren.

Über eine Schleife können alle markierten Objekte aus dem Cte-Tree oder der Cte-Liste ermittelt und weiterverarbeitet werden.

```
tCteItem # tCteData->CteRead(_CteFirst);while (tCteItem != 0){  tCteID    # tCteItem->spID;  tCteN
```

Kontakt

Eigenschaften

Über die Eigenschaften können die Oberflächen- bzw. Druckformular-Objekte modifiziert werden.

Alphabetische

Siehe [Liste aller](#)

Eigenschaften

Die Eigenschaften können im Designer durch das Eigenschafts-Fenster und prozedural durch die Befehle [WinPropGet\(\)](#) und [WinPropSet\(\)](#) bzw. [PrtPropGet\(\)](#) und [PrtPropSet\(\)](#) gelesen und gesetzt werden.

Die Eigenschaften untergliedern sich in folgende Bereiche:

- [Base \(Basis-Eigenschaften\)](#)
- [Database \(Datenbank-Eigenschaften\)](#)
- [Placement \(Positions-Eigenschaften\)](#)
- [Grouping \(Gruppierungs-Eigenschaften\)](#)
- [State \(Status-Eigenschaften\)](#)
- [Color \(Farb-Eigenschaften\)](#)
- [Font \(Schrift-Eigenschaften\)](#)
- [Icon \(Symbol-Eigenschaften\)](#)
- [Help \(Hilfe-Eigenschaften\)](#)
- [Focus \(Eingabe-Eigenschaften\)](#)
- [Key \(Tasten-Eigenschaften\)](#)
- [Menu \(Menü-Eigenschaften\)](#)
- [Embedded \(Eigenschaften eingebundener Objekte\)](#)
- [Popup \(Eigenschaften von Popup-Objekten\)](#)
- [Special \(Objektspezifische Eigenschaften\)](#)
- [Function \(Funktions-Eigenschaften\)](#)
- [Theming \(Theme-Eigenschaften\)](#)
 - Theme-Eigenschaften:
 - ◆ [Fill Color \(Farb-Eigenschaften für Füllfarbe von Themes\)](#)
 - ◆ [Text Color \(Farb-Eigenschaften für Textfarbe von Themes\)](#)
 - ◆ [Border Color \(Farb-Eigenschaften für Rahmenfarbe von Themes\)](#)
 - ◆ [Check Color \(Farb-Eigenschaften für Markierungsfarbe von Themes\)](#)
 - ◆ [Arrow Color \(Farb-Eigenschaften für Pfeilfarbe von Themes\)](#)
 - ◆ [Separator Color \(Farb-Eigenschaften für Separatorfarbe von Themes\)](#)
 - ◆ [Splitter Color \(Farb-Eigenschaften für Splitterfarbe von Themes\)](#)
 - ◆ [Grip Color \(Farb-Eigenschaften für Ziehpunktfarbe von Themes\)](#)
 - ◆ [Border Size \(Größen-Eigenschaften für Rahmen von Themes\)](#)
 - ◆ [Responsive Size \(Größen-Eigenschaften von Themes\)](#)
 - ◆ [Spin Color \(Farb-Eigenschaften für Spin-Bereich von Themes\)](#)
 - ◆ [Popup button \(Farb-Eigenschaften für Popup-Button von Themes\)](#)
 - ◆ [Scroll button \(Farb-Eigenschaften für Scroll-Button von Themes\)](#)
 - ◆ [Editor Color \(Farb-Eigenschaften für CodeEdit-Objekt von Themes\)](#)
 - ◆ [Image Color \(Farb-Eigenschaften für Bilder von Themes\)](#)
 - ◆ [Tick Color \(Farb-Eigenschaften für Markierungsfarbe von Themes\)](#)

Kontakt

Base (**Basis-Eigenschaften**)

Eigenschaften

sortiert nach

Gruppen,

Siehe Alphabetische

Liste aller

Eigenschaften

Caption Inhalt bzw. Titel des Objektes

CaptionBigInt Inhalt des BigIntEdit-Objektes

CaptionColAuto Titel für die automatische Farbe

CaptionColor Inhalt des ColorEdit-Objektes

CaptionDate Inhalt des DateEdit-Objektes

CaptionDecimal Inhalt des DecimalEdit-Objektes

CaptionFloat Inhalt des FloatEdit-Objektes

CaptionInt Inhalt des IntEdit-Objektes

CaptionTime Inhalt des TimeEdit-Objektes

CryptKey Passwort für Verschlüsselung von PrintJob- und binären Objekten

Custom Benutzerdefinierte Eigenschaft

DecEditFlags Optionen der Darstellung

Name Name zur eindeutigen Identifikation des Objektes

StyleBorder Darstellungsart des Randes

StyleCaption Anzeige eines Funktionsergebnisses

TextLabel TextLabel

Caption

Inhalt bzw. Titel des Objekts

Typ alpha(8192)

Liste, Objekte,

Siehe WinPropGet(),
WinPropSet()

In der Caption wird der Inhalt des Objekts, der für den Benutzer sichtbar ist, abgelegt. Es können Werte vom Typ alpha zugewiesen werden. Für die anderen Datentypen gibt es entsprechende Eigenschaften (zum Beispiel CaptionInt).

Die Darstellung kann je nach verwendetem Objekt unterschiedlich sein. In folgenden Objekten kann die Caption gesetzt werden:

- Schaltflächen-Objekte
 - Button- und Toolbarbutton-Objekte
 - StatusbarButton
 - MenuItem
 - Bild-Objekte
 - Progressbar
 - PrtJobPreview
 - PrtBarcode
 - Rtf-Objekte
 - TextEdit
 - GroupSplit
 - Application
 - Chromium
- Schaltflächen-Objekte

Bei Verwendung des &-Zeichens wird das nachfolgende Zeichen mit einem Unterstrich dargestellt. Somit ist es möglich über die Kombination  + Auswahlzeichen den Button auszulösen (mnemonische Auswahl). Folgende Objekte unterstützen diese Möglichkeit:

- ♦ Button
 - ♦ MenuButton
 - ♦ Radiobutton
 - ♦ Checkbox
 - ♦ NotebookPage
 - ♦ MenuItem
 - ♦ Groupbox
 - ♦ Label (sofern die Eigenschaft ObjLink gesetzt ist)
- Button und Toolbar-Button

Bei diesen Objekten kann eine mehrzeilige Caption angegeben werden. Im Designer kann dazu durch einen Doppelklick auf den Begriff Caption ein Eingabefenster aufgerufen werden. Zur Laufzeit kann der Eigenschaft eine Zeichenkette, die mit StrChar(13) zusammengesetzt ist, zugewiesen werden.

- Statusbar-Button

Kontakt

Folgende Makrodefinitionen können angegeben werden:

```
'$(DATE [, <datums-format>])' // Datum '$(TIME [, <zeit-format>])' // Zeit
```

Zur Laufzeit wird das lokale Datum, bzw. die lokale Uhrzeit ausgegeben. Die Aktualisierung erfolgt automatisch. Die Format-Zeichenkette <datums-format> und <zeit-format> sind optional und identisch zu denen der Eigenschaft ScalaLabels.

```
'$(KEY,NUM)' // Anzeige des Toggle-Status der NumLock-Taste '$(KEY,INS)' // Anzeige des To
```

Der Status der Taste wird nur dann in der Statusbar angezeigt, wenn das Objekt, das den Fokus besitzt, ein Eingabeobjekt ist.

- MenuItem

Das Tilde-Zeichen ~ bewirkt eine rechtsbündige Darstellung des der Tilde folgenden Textes.

- Picture, MetaPicture, PrtPicture oder PrtMetaPicture

Die Angabe einer externen Datei kann mit einem absoluten oder relativen Pfad erfolgen. Bei der Angabe eines absoluten Pfades wird ein *, bei der eines relativen Pfades (relativ zum Startverzeichnis des CONZEPT 16-Clients) wird ein ! vorangestellt.

Soll in den Objekten ein binäres Objekt aus der Datenbank dargestellt werden, muss vor dem Pfad und Namen des Objektes die Zeichen >0 angegeben werden. Der Pfad eines binären Objekts beginnt immer mit "\\". Wurde das binäre Objekt verschlüsselt gespeichert, muss der Schlüssel in die Eigenschaft CryptKey eingetragen werden.

Bei Angabe des Namens ohne Präfix ('!', '*' und '>') wird eine Bilddatei in der Datenbank referenziert. Bei mit DbaConnect() verbundenen Datenbanken wird nach > der entsprechende Dateinummernbereich (z. B. '>2') angegeben.

Beispiele:

```
'!Picture\Artikel.tif'           // externe Datei relativ angegeben '*C:\c16\Picture\Artikel.'
```

Beim Picture-Objekt kann ein Bild auch aus einem Memory-Objekt angezeigt werden, dazu muss der Deskriptor des Objekts in die Eigenschaft MemObjHandle eingetragen und ImageSource auf WinImageSourceMemory gesetzt werden.

- Progress

Angabe der Einheit des Fortschrittbalkens. Die Einheit wird in dem Balken nach der Eigenschaft ProgressPos angegeben.

- PrtJobPreview

Die Angabe eines gespeicherten Druckjobs kann mit einem absoluten oder relativen Pfad erfolgen. Bei der Angabe eines absoluten Pfades wird ein *, bei der eines relativen Pfades (relativ zum Startverzeichnis des CONZEPT

Kontakt

16-Clients) wird ein ! vorangestellt. Liegt der Druckjob als binäres Objekt vor, kann es mit dem Präfix > angesprochen werden (vgl. Angaben beim Picture-Objekt). Ohne ein Präfix wird ein PrintDoc-Objekt dargestellt.

Beispiele:

```
'!PrintJobs\UserList.job'           // externe Datei relativ angegeben '*C:\c16\PrintJobs\Use
```

Soll die Anzeige des Druckjobs Informationen eines bestimmten Druckers berücksichtigen, sollte die Eigenschaft PrtDevice vor der Eigenschaft Caption gesetzt werden.

- PrtBarcode

Das Format wird durch die Länge der Barcodenummer und der integrierten Prüfziffern automatisch erkannt. Um ein Format zu erzwingen, kann der Barcodenummer ein Präfix vorangestellt oder ein Barcode in der Eigenschaft TypeBarcode ausgewählt werden. Die zur Verfügung stehenden Barcodes sind in der Eigenschaft TypeBarcode erläutert.

Beispiel:

```
'ISBN3-88322-453-7'
```

- RtfEdit oder PrtRtf

Bei diesem Objekt kann über die Eigenschaft Caption der Inhalt gesetzt werden. Der Text kann dabei entweder gesetzt oder erweitert werden. Zur Unterscheidung der Möglichkeiten stehen die Konstanten WinEditDefault, WinEditAll und WinEditMark zur Verfügung.

Beispiele:

```
tHd1RtfEdit->wpCaption(_WinEditDefault) # 'Neuer Text'; // Defaultwert setzen tHd1RtfEdit-
```

Beim Auslesen der Eigenschaft wird nur der innerhalb des Objekts markierte Bereich zurückgegeben. Eine Markierung kann in der Eigenschaft range angegeben werden.

- TextEdit

Bei diesem Objekt kann über die Eigenschaft Caption der Inhalt gesetzt werden. Der Text kann dabei entweder gesetzt oder erweitert werden. Zur Unterscheidung der Möglichkeiten stehen die Konstanten WinEditAll und WinEditMark zur Verfügung. Die Wirkungsweise entspricht der bei RtfEdit-und PrtRtf-Objekten.

Beim Auslesen der Eigenschaft wird nur der innerhalb des Objektes markierte Bereich zurückgegeben. Eine Markierung kann in der Eigenschaft range angegeben werden.

- GroupSplit

Oberhalb der GroupTile-Objekte wird der Text eingeblendet. Die Größe des Textes ist vom System vorgegeben und kann nicht über die Eigenschaft Font

Kontakt

geändert werden. Die Font-Attribute und der Zeichensatz werden berücksichtigt.

- Application

Kommt es infolge einer Verklemmung zu einem Wartezustand (siehe auch DtaBegin()), wird ein entsprechendes Benachrichtigungsfenster eingeblendet:



Der Text in der Titelzeile kann durch Setzen der Eigenschaft Caption beim Application-Objekt verändert werden. Die Hintergrundfarbe des Titels wird mit der Eigenschaft CaptionColor festgelegt.

- Chromium-Objekt

Gibt die URL des anzuzeigenden Inhaltes an. Beim Setzen der Eigenschaft werden die Ereignisse EvtCroNavigate und EvtChanged ausgelöst. Durch Weiterleitungen kann die finale URL des Inhaltes von dieser URL abweichen. Die tatsächliche URL ist über die Eigenschaft wpUrl lesbar.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

Caption...

Inhalt des Objekts

Typ var

Siehe [Liste, Objekte](#)

Inhalt des Objekts. Für diese Datentypen existiert eine eigene Eigenschaft:

- [Caption](#)
- [CaptionBigInt](#)
- [CaptionColor](#)
- [CaptionDate](#)
- [CaptionDecimal](#)
- [CaptionFloat](#)
- [CaptionInt](#)
- [CaptionLogic](#)
- [CaptionTime](#)

Nur die [Items](#) und [SubItems](#) eines [RecView-Objektes](#) verfügen über einen dynamischen Inhaltstyp.

CaptionBigInt

Inhalt des Objektes

Typ bigint

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In der CaptionBigInt wird der Inhalt des Objektes, der für den Benutzer sichtbar ist, abgelegt. Es können Werte vom Typ bigint angegeben werden.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

CaptionColAuto

Titel für automatische Farbe

Typ alpha

Liste, Objekte, ColAuto,

Siehe WinPropGet(),

WinPropSet()

In dieser Eigenschaft kann der Titel für die automatische Farbe gesetzt werden.

Der Titel für die automatische Farbe wird nur dann angezeigt, wenn die Eigenschaft ModeColor auf WinModeColorPalette gesetzt ist.

Ist die Eigenschaft leer, wird "Automatisch" angezeigt.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

CaptionColor

Inhalt des Objektes

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In der CaptionColor wird der Inhalt des Objektes, der für den Benutzer sichtbar ist, abgelegt. Es können Werte vom Typ int zugewiesen werden.

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Der Wert, der aus dieser Eigenschaft ausgelesen wird, kann ohne weitere Konvertierung einer Farbeigenschaft (zum Beispiel ColBkg) zugewiesen werden.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

CaptionDate

Inhalt des Objektes

Typ date

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In der CaptionDate wird der Inhalt des Objektes, der für den Benutzer sichtbar ist, abgelegt. Es können Werte vom Typ date angegeben werden.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

CaptionDecimal

Inhalt des Objektes

Typ decimal

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In der CaptionDecimal wird der Inhalt des Objektes, der für den Benutzer sichtbar ist, abgelegt. Es können Werte vom Typ decimal angegeben werden.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

CaptionFloat

Inhalt des Objektes

Typ float

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In der CaptionFloat wird der Inhalt des Objektes, der für den Benutzer sichtbar ist, abgelegt. Es können Werte vom Typ float angegeben werden.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

CaptionInt

Inhalt des Objektes

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In der CaptionInt wird der Inhalt des Objektes, der für den Benutzer sichtbar ist, abgelegt. Es können Werte vom Typ int angegeben werden.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

CaptionLogic

Inhalt des Objektes

Typ logic

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In der CaptionLogic wird der Inhalt des Objektes, der für den Benutzer sichtbar ist, abgelegt. Es können Werte vom Typ logic angegeben werden.

Die Eigenschaft kann über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

CaptionTime

Inhalt des Objektes

Typ time

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In der CaptionTime wird der Inhalt des Objektes, der für den Benutzer sichtbar ist, abgelegt. Es können Werte vom Typ time angegeben werden.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

CryptKey



**Passwort für Verschlüsselung von Druckjobs und binären Objekten Typ
alpha(64)**

Siehe [Liste](#), [Objekte](#), [WinPropGet\(\)](#), [WinPropSet\(\)](#)

Die Eigenschaft definiert ein Passwort, das für die Entschlüsselung von verschlüsselten Druckjob-Dateien oder binären Objekten verwendet wird. Der Inhalt kann nur dann angezeigt werden, wenn der korrekte Schlüssel angegeben ist.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle [WinPropGet\(\)](#) und [WinPropSet\(\)](#) gelesen bzw. geändert werden.

Kontakt

Custom

Benutzerdefinierte Eigenschaft

Typ alpha(8192)
Liste, Objekte, WinPropGet(),
Siehe

Die Eigenschaft dient zum Ablegen benutzerdefinierter Daten. Sie wird durch CONZEPT 16 nicht ausgewertet.

Da die Eigenschaft alphanumerischen Typs ist, muss gegebenenfalls eine Konvertierung der abzulegenden Daten erfolgen.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() / PrtPropGet() und WinPropSet() / PrtPropSet() gelesen bzw. geändert werden.

Kontakt

DecEditFlags

Optionen der Darstellung

Typ [int](#)

Siehe [Liste](#), [Objekte](#)

In dieser Eigenschaft können Darstellungsoptionen des Objekts [DecimalEdit](#) definiert werden.

Es kann nur die Option [WinDecEditDecimalUndef](#) gesetzt werden.

FlagsTitlebar

Funktionen der Titelzeile

Typ int

Siehe Objekte

Die Eigenschaft bestimmt die Funktionen, die in der Titelleiste des Frame- oder GroupTile-Objekts zur Verfügung stehen. Beim Frame-Objekt wird diese Eigenschaft nur dann ausgewertet, wenn die Eigenschaft StyleFrame auf WinWndFrameSplash gesetzt ist.

Folgende Ausprägungen stehen zur Verfügung:

- WinTitlebarCaption

Ist diese Option gesetzt, wird eine Titelzeile angezeigt.

- WinTitlebarMove

Das Objekt kann über die Titelzeile verschoben werden. Ein GroupTile-Objekt kann nur verschoben werden, wenn das andere GroupTile-Objekt ebenfalls verschoben werden kann.

- WinTitlebarClose

Das Objekt kann geschlossen werden.

- WinTitlebarMaximize

Das Objekt kann maximiert werden.



Ist die Option WinTitlebarCaption nicht gesetzt, können auch die anderen Optionen nicht gesetzt werden.

Kontakt

Name Name zur eindeutigen Identifikation des Objektes

Typ alpha(60)

Siehe Liste, Objekte, WinPropGet(), WinPropSet()

Aus Prozeduren heraus können Objekte über deren Namen referenziert werden (\$Objektname). Daher sollte auf eine eindeutige Vergabe von Objektnamen innerhalb des selben Fensters geachtet werden.

Bei der Vergabe des Namens ist folgendes zu beachten:

- Der Objektname hat eine zulässige Länge von 1 bis 60 Zeichen
- Das erste Zeichen muss ein ASCII-Zeichen im Bereich A-z sein
- Das zweite bis zum 60. Zeichen kann alphabetisch, numerisch, ein Unterstrich oder Punkt sein

<Obj-Name> := {A..z}[[{A..z,0..9,_,.}]]

StyleBorder

Darstellungsart des Randes

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet(),

ShowBorder

Diese Eigenschaft bestimmt das Aussehen des Objektrandes.

Es gibt folgende Randstile:

WinBorStandard Kein Rahmen

WinBorSunken Eingedrückt

WinBorRaised Herausgestellt

WinBorFrame Frame-Rahmen

 Der Wert WinBorFrame kann nur in Eingabe-Objekten im RecView-Objekt (siehe EvtLstEditActivate) gesetzt werden.

Wird bei einem Fenster-Objekt (Frame, AppFrame oder MdiFrame) diese Eigenschaft gesetzt, verändert sich ebenfalls die Eigenschaft StyleFrame auf WinWndNone.

Kontakt

StyleCaption

Ausgabe festlegen

Typ int

Liste, Objekte,

Siehe PrtPropGet(),

PrtPropSet()

Diese Eigenschaft bestimmt, welche Ausgabe im PrtText oder PrtTblCellText-Objekt erfolgen soll.

Folgende Konstanten sind zulässig:

PrtStyleCapNormal Der Inhalt der Eigenschaft Caption wird angezeigt

PrtStyleCapPageNo Die aktuelle Seitennummer wird angezeigt

PrtStyleCapPageCount Die Anzahl der Seiten im Druckjob wird angezeigt.

PrtStyleCapFnResult Das Ergebnis einer Funktionsberechnung wird angezeigt. Die Funktion wird in den Funktions-Eigenschaften festgelegt.
Die Ausprägung wird nur beim PrtTblCellText-Objekt ausgewertet.

Kontakt

TextLabel



Auszugebender Text

Typ alpha

Liste, Objekte,
WinPropGet(),

Siehe WinPropSet(),

PrtPropGet(),
PrtPropSet()

Mit dieser Eigenschaft wird der Text definiert, der in einem Picture- oder MetaPicture-Objekt ausgegeben wird.

Kontakt

Database (Datenbank-Eigenschaften)

Eigenschaften

sortiert nach

Gruppen,

Siehe Alphabetische

Liste aller

Eigenschaften

DbFieldName Verknüpftes Datenbankfeld

DbFileNo Nummer der zu verwendenden Datei

DbFilter Deskriptor des zu verwendenden Filters

DbKeyNo Nummer des zu verwendenden Schlüssels

DbKeyReverseSort Sortierreihenfolge umkehren

DbLinkFileNo Zielfile der zu verwendenden Verknüpfung

DbRecBuf Datensatzpuffer

DbRecID Datensatz-ID des ausgewählten Datensatzes

DbRecID64 64-Bit Datensatz-ID der ausgewählten Gruppe

DbSelection Deskriptor der zu verwendenden Selektion

DbTextBuf Deskriptor des zu verwendenden Textpuffers

DbVar Globale Datenbereiche

StoView Anzuzeigende Objekte

StoViewElmTypes Anzuzeigende Elementtypen

UrmParentName Name des Eltern-Objekts

UrmParentHandle Deskriptor des Eltern-Objekts

DbFieldName

Verknüpftes Datenbankfeld

Typ alpha(20)

Liste, Objekte,

Siehe WinPropGet(),
WinPropSet()

In dieser Eigenschaft kann ein Datenbankfeld angegeben werden. Die Angabe kann entweder durch den Namen des Feldes oder durch die Nummer der Datei, des Teildatensatzes und des Feldes (zum Beispiel 100,1,2) erfolgen. Ein Feld kann ebenfalls über die Drag & Drop-Funktionalität aus der Datenbankinfo in die Eigenschaft gezogen werden. Wird zu einem späteren Zeitpunkt die Datenstruktur verändert, sodass der Feldname nicht mehr vorhanden ist, wird der Inhalt der Eigenschaft im Designer rot dargestellt.

Nicht alle Oberflächen-Objekte können alle Datentypen darstellen. Welche Datentypen angezeigt werden können, ist bei den entsprechenden Oberflächenobjekten beschrieben.

Ist in den Einstellungen die Option "Setzen von DbFieldName setzt auch Eingabelimit" aktiviert, wird nach der Angabe eines Datenbankfeldes der zulässige Eingabebereich des Objektes an das Datenbankfeld angepasst. Dies betrifft die Eigenschaften LengthMax (bei Feldern vom Typ alpha) sowie MinInt / MinBigInt und MaxInt / MaxBigInt (bei Feldern vom Typ Ganzahlig kurz, Ganzahlig lang und Ganzahlig

(64)). Die entsprechenden Eingabelimits werden sofort umgesetzt, wirken sich also auf die Eingabemöglichkeiten in der Caption-, CaptionInt- bzw. CaptionBigInt-Eigenschaften aus.

Wurden die Eingabemöglichkeiten bereits zuvor eingeschränkt, werden nur die Eingabebeschränkungen verändert, die die Möglichkeiten des zugeordneten Datenbankfeldes überschreiten. Ist zum Beispiel der Eingabebereich auf -100 bis +100 eingeschränkt und es wird ein Datenbankfeld von Typ Ganzahlig kurz zugewiesen, verändert sich die Eingabemöglichkeit auf 0 bis +100.

Unter folgenden Bedingungen wird der Inhalt (Caption) des Eingabeobjektes automatisch in die Feldpuffer übertragen:

- Das Eingabeobjekt verliert den Eingabefokus
- Das Eingabeobjekt besitzt den Eingabefokus und eine Menüoption wurde angewählt
- Das Eingabeobjekt besitzt den Eingabefokus und ein Button bzw. Toolbarbutton wurde angewählt

Erhält das Objekt den Fokus, wird der Inhalt des Feldpuffers automatisch in das Eingabeobjekt übertragen.

Dieser Automatismus kann über die Flags WinAppObj2FldOff und WinAppFld2ObjOff nach Bedarf aktiviert bzw. deaktiviert werden.

Ein Übertragen des Inhaltes zwischen Objekt und Feldpuffer kann auch mit dem Befehl WinUpdate() unter Verwendung der Optionen WinUpdObj2Fld bzw. WinUpdFld2Obj erfolgen. Je nach dem in welche Richtung das Übertragen

Kontakt

stattfinden soll, wird eine der beiden Optionen verwendet.

Damit ein geänderter Feldpuffer, z. B. durch Lesen eines Datensatzes, in dem Objekt angezeigt wird, muss dieser über die Funktion WinUpdate() und der Option WinUpdFld2Obj in das Objekt übertragen werden.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() / PrtPropGet() und WinPropSet() / PrtPropSet() gelesen bzw. geändert werden.

Kontakt

DbFileNo

Nummer der zu verwendenden Datei

Typ int
Siehe Liste, Objekte, WinPropGet(),

Die Datensatzoperationen werden auf die in dieser Eigenschaft angegebene Datei bezogen.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() / PrtPropGet() und WinPropSet() / PrtPropSet() gelesen bzw. geändert werden.

Kontakt

DbFilter



Deskriptor des zu verwendenden Filters

Typ int

Liste, Objekte,

Siehe RecFilterCreate(),

WinPropGet(), WinPropSet()

Mit dieser Eigenschaft wird die Anzeige der Datensätze in einem RecList-Objekt auf bestimmte Datensätze, die einem Filter-Kriterium genügen, beschränkt.



Da der Deskriptor des Filters erst zur Laufzeit bekannt ist, kann diese Eigenschaft nicht im Designer gesetzt werden.

Beispiel:

```
tFltHdl # RecFilterCreate(ADR.D.Adressen, ADR.S.aPlz); tFltHdl->RecFilterAdd(1, _FltAND, _FltLike,
```

Die Eigenschaft kann bei Oberflächen-Objekten über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden. Bei Druck-Objekten werden die Befehle PrtPropGet() und PrtPropSet() verwendet.

Kontakt

DbKeyNo Nummer des zu verwendenden

Schlüssels Typ int

Siehe [Liste](#), [Objekte](#), [WinPropGet\(\)](#),

Die Reihenfolge der Datensätze innerhalb eines [RecList](#)-Objekts wird durch den in dieser Eigenschaft angegebenen Schlüssel (oder bei einer Verknüpfung, der dort angegebene Schlüssel) bestimmt. Die Reihenfolge kann durch Setzen der Eigenschaft [DbKeyReverseSort](#) umgedreht werden.

Ist in der Eigenschaft [DbLinkFileNo](#) eine Datei angegeben, erfolgt der Zugriff über die in dieser Eigenschaft angegebene [Verknüpfung](#).

Die Eigenschaften [DbKeyReverseSort](#) und [DbKeyNo](#) beim [RecList](#)-Objekt werden ignoriert, wenn die Sortierung durch Einstellungen in den Spalten vorgegeben werden. Dies wird erreicht durch Setzen der Ausprägung

[WinClmSortFlagsAutoActive](#) in der Eigenschaft [ClmSortFlags](#) und Setzen der Schlüsselnummer in [DbKeyNo](#) des Spalten-Objekts. Die Sortierung der Liste wird bestimmt durch die Einstellungen der Spalte, die die Ausprägung

[WinClmSortFlagsAutoSelected](#) in der Eigenschaft [ClmSortFlags](#) gesetzt hat.

Damit auch die korrekten Symbole neben den Spaltenüberschriften erscheinen, muss die Eigenschaft [ClmSortImage](#) auf das korrekte Symbol gesetzt werden. Das ist entweder der Pfeil nach oben, Peil nach unten oder (für die Spalten nach denen sortiert werden kann) ein Schlüsselsymbol.

Sind die Einstellungen bei den Spalten erfolgt, wird zur Laufzeit durch Anklicken der Spaltenüberschriften die Liste neu sortiert. Die Sortierung wird dabei durch den bei der Spalte angegebenen Schlüssel vorgenommen.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle [WinPropGet\(\)](#) und [WinPropSet\(\)](#) gelesen bzw. geändert werden.

DbKeyReverseSort

Sortierreihenfolge umkehren

Typ logic

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Die Reihenfolge der Datensätze innerhalb eines RecList-Objekts wird durch den in der Eigenschaft DbKeyNo angegebenen Schlüssel (oder bei einer Verknüpfung, des dort angegebenen Schlüssels) bestimmt. Ist diese Eigenschaft gesetzt, werden die Zeilen innerhalb der Liste in umgekehrter Reihenfolge angezeigt.

Im Gegensatz zum Schlüsselfeld-Attribut "absteigende Sortierung" (siehe Schlüssel bearbeiten) wird nicht die Reihenfolge in einem Schlüsselfeld, sondern die Reihenfolge in allen Schlüsselfeldern umgekehrt.

Reihenfolge ohne

DbKeyReverseSort

CH 3000

CH 4589

CH 4658

CH 6000

D 05895

D 10258

D 25598

D 63150

D70597

Reihenfolge mit

DbKeyReverseSort

D 70597

D 63150

D 25598

D 10258

D 05895 CH

6000 CH 4658

CH 4589 CH

3000

Die Eigenschaften **DbKeyReverseSort** und **DbKeyNo** beim RecList-Objekt werden ignoriert, wenn die Sortierung durch Einstellungen in den Spalten vorgegeben werden. Dies wird erreicht durch Setzen der Ausprägung WinClmSortFlagsAutoActive in der Eigenschaft ClmSortFlags des Spalten-Objekts. Die Sortierung der Liste wird bestimmt durch die Einstellungen der Spalte, die die Ausprägung WinClmSortFlagsAutoSelected in der Eigenschaft ClmSortFlags gesetzt hat.

Kontakt

DbLinkFileNo Zielfile der zu verwendenden

Verknüpfung Typ int

Siehe Liste, Objekte, WinPropGet(), WinPropSet()

In dieser Eigenschaft wird die Nummer der Zielfile angegeben, wenn in dem Objekt nur verknüpfte Datensätze angezeigt werden sollen.

In der Eigenschaft DbFileNo ist die Dateinummer der Ausgangsdatei und in DbKeyNo die Nummer der Verknüpfung anzugeben.

DbRecBuf



Datensatzpuffer

Typ handle

Liste, Objekte,
Datensatzpuffer,

Siehe Datensatzpuffer
in MDI-Frames
(Blog)



Die Abfrage der Eigenschaft kann prozedural erst im EvtCreated des Objektes oder des übergeordneten Objektes erfolgen. Davor wird immer 0 zurückgegeben.

Die Eigenschaft DbRecBuf hat je nach Objekt folgende Bedeutung:

- MDI-Anwendungsfenster

Über diese Eigenschaft kann ein Datensatzpuffer für ein oder mehrere Dateien erstellt werden. Der Datensatzpuffer wird automatisch von dem MDI-Anwendungsfenster verwaltet. Mit Hilfe der Puffer ist es möglich, dass in jedem MdiFrame-Objekt mit normalen Datensatzoperationen und Feld-Puffern gearbeitet werden kann, ohne das sich gleichartige MdiFrame-Objekte gegenseitig beeinflussen.

Ist diese Eigenschaft gesetzt, wird der Inhalt des Feldpuffers der angegebenen Dateien gesichert und zum gegebenen Zeitpunkt wieder hergestellt. Wird ein neues MDI-Fenster an ein AppFrame-Objekt angehängt, stehen erst nach dem Befehl WinAdd() oder WinAddByName() die Feldpuffer des MDI-Fensters zur Verfügung. Die Vorbelegung der Feldpuffer muss also nach dem Anhängen des MDI-Fensters erfolgen und nicht im EvtInit des MDI-Fensters.

- RecList

Wird ein RecList-Objekt neu gezeichnet (zum Beispiel durch WinUpdate()), wird ebenso der Datensatzpuffer der betreffenden Datei verändert. Die in dieser Eigenschaft angegebenen Datensatzpuffer werden vor dem Update der RecList gesichert und anschließend wieder hergestellt. Dies ist auch dann der Fall, wenn in dem Ereignis EvtLstSelect das Lesen eines Datensatzes stattfindet. Soll der Datensatzpuffer des in diesem Ereignis gelesenen Datensatzes erhalten bleiben, muss der Datensatzpuffer in den Puffer der RecList mit dem Befehl WinUpdate() und der Option WinUpdFld2Buf übertragen werden.

Die Eigenschaft kann innerhalb einer Prozedur ausgelesen werden, um den Datensatzpuffer des Objekts zu ermitteln. Dazu muss die Nummer der Datei als Parameter angegeben werden. Besitzt das Objekt keinen Datensatzpuffer der angegebenen Datei, wird 0 zurückgegeben, sonst der Deskriptor des Puffers.

Beispiel:

```
tHdlBuf # aEvt:Obj->wpDbRecBuf(tblCstCustomer);if (tHdlBuf > 0){ ... }
```

Prozedural kann ein Datensatzpuffer mit dem Wert 1 aktiviert und mit dem Wert 0 deaktiviert werden. Die Angabe eines Datensatzpuffers führt zum Laufzeitfehler ErrValueInvalid. Zusätzlich muss als Parameter der Eigenschaft die Dateinummer übergeben werden. Die Dateinummer muss im Bereich von 1 bis 1000 liegen.

Kontakt

Außerhalb des Bereichs wird ebenfalls der Laufzeitfehler [ErrValueInvalid](#) ausgelöst.

Liegt die Dateinummer innerhalb des Bereichs, existiert jedoch keine Datei mit der Nummer, kommt der Laufzeitfehler ErrFileInvalid.

Beispiele:

```
// Aktivieren des Datensatzpuffers für die Datei tblCstCustomerEvt:Obj->wpDbRecBuf(tblCstCustomerEvt)
```



Wird der Parameter der Eigenschaft weggelassen, wird der Laufzeitfehler ErrValueInvalid ausgelöst.

Ein ermittelter Datensatzpuffer kann nicht mit RecBufDestroy() entfernt werden. Die Anweisung liefert keinen Fehler zurück, sondern wird ignoriert.

Kontakt

DbRecID

RecID des ausgewählten Datensatzes

Typ **int**

Siehe [Liste](#), [Objekte](#), [WinPropGet\(\)](#)

Über diese Eigenschaft kann die **Datensatz-ID** eines in dem **RecList**-Objekt selektierten Datensatzes ermittelt werden.

Die Eigenschaft kann über den Befehl [WinPropGet\(\)](#) ermittelt werden.



Bei der **Datensatz-ID** handelt es sich um eine interne ID, die sich durch bestimmte Aktionen (z. B. Export und Import der Datensätze) ändern kann. Sie sollte daher nicht als dauerhafte Referenz auf einen Datensatz verwendet werden.

Kontakt

DbRecID64



64-Bit RecID des ausgewählten Datensatzes / der ausgewählten Gruppe

Typ **bigint**

Siehe [Liste](#), [Objekte](#), [WinPropGet\(\)](#)

Über diese Eigenschaft kann die Datensatz-ID des selektierten Datensatzes in einem RecList-Objekt oder der selektierten Gruppe in einem RecView-Objekt ermittelt werden.

Die Eigenschaft kann über den Befehl [WinPropGet\(\)](#) ermittelt werden.



Zur Zeit werden ausschließlich 32-Bit-Werte als 64-Bit-Zahl zurückgegeben.

Kontakt

DbSelection Deskriptor der zu verwendenden
Selektion Typ int

Siehe Liste, Objekte, SelOpen(), WinPropGet(),

Mit dieser Eigenschaft wird die Anzeige bzw. Ausgabe der Datensätze in einem RecList- und PrintDocRecord-Objekt auf den Inhalt einer Selektion beschränkt.

Da der Deskriptor der Selektion erst zur Laufzeit bekannt ist, kann diese Eigenschaft nicht im Designer gesetzt werden.

-  Eine Selektion, die bereits in einem RecList-Objekt angezeigt wird, darf nicht mit dem Befehl SelRun() durchgeführt werden.
-  Werden Schlüsselwerte von Datensätzen der Selektion nachträglich geändert, kann die Option SelKeyUpdate verwendet werden, um Probleme bei der Anzeige der Datensätze in der RecList zu vermeiden.

Beispiel:

```
tSelHdl # SelOpen(); tSelHdl->SelRead(Adr.D.Adressen, _SelSharedLock, 'ADR_PLZBEREICH'); $RecList->
```

Die Eigenschaft kann über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

DbTextBuf Deskriptor des zu verwendenden
Textpuffers Typ int

Siehe
Liste, Objekte, TextOpen(), WinPropGet(),
WinPropSet()

Mit dem Setzen dieser Eigenschaft wird der Inhalt des Textpuffers in das Objekt TextEdit übertragen. Mit dem Befehl WinUpdate() und der Option WinUpdBuf2Obj kann das Übertragen des Textpuffers zu jedem Zeitpunkt erfolgen. Das Übertragen des Inhaltes eines Edit-Objektes in einen Textpuffer wird über WinUpdate() mit der Option WinUpdObj2Buf vorgenommen.

Wird ein Textpuffer mit dem TextEdit-Objekt verknüpft, kann die Eigenschaft Caption nicht mehr verwendet werden. Ein Setzen dieser Eigenschaft hat keine Wirkung und das Auslesen liefert immer einen leeren alpha-String.

Das TextEdit-Objekt hat keine Limitierung der Zeilenlänge.



Da der Deskriptor des Textpuffers erst zur Laufzeit bekannt ist, kann diese Eigenschaft nicht im Designer gesetzt werden.

Die Eigenschaft kann nur über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

DbVar

Globale Datenbereiche

Typ [int](#)

Siehe [Liste, Objekte](#)

Wird die Eigenschaft mit der Maus aktiviert, wird eine Liste aller globalen Datenbereiche angezeigt.

Die Einträge, die in der Liste selektiert werden, werden automatisch beim Laden der Instanz allokiert. Somit entfällt das Anlegen und Freigeben der Datenbereiche über [VarAllocate\(\)](#) und [VarFree\(\)](#).

Die Instanzen der globalen Datenbereiche werden bei Erhalt des Eingabefokus automatisch ausgewählt. Das prozedurale Auswählen der Instanzen über [VarInstance\(\)](#) ist nicht notwendig.

StoView

Anzuzeigende Objekte

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft wird festgelegt, welche Objekte der Datenbank in dem StoList-Objekt angezeigt werden. Der Eigenschaft können eine Reihe von Konstanten zugewiesen werden:

- WinStoViewDialog
- WinStoViewElmGroup
- WinStoViewMenu
- WinStoViewMetaPicture
- WinStoViewPicture
- WinStoViewPrintDoc
- WinStoViewPrintDocRecord
- WinStoViewPrintForm
- WinStoViewPrintFormList
- WinStoViewProcedure
- WinStoViewText
- WinStoViewTheme
- WinStoViewUser
- WinStoViewUserGroup

Kontakt

StoViewElmTypes

Anzuzeigende Elementtypen

Typ int

Siehe Liste, Objekte

Diese Eigenschaft wird nur ausgewertet, wenn die Eigenschaft StoView auf den Wert WinStoViewElmGroup und die Eigenschaft UrmParentName bzw. UrmParentHandle gesetzt sind.

In der Liste werden nur die Elemente angezeigt, die den angegebenen Element-Typen entsprechen.

Folgende Konstanten können der Eigenschaft zugewiesen werden:

<u>_WinStoViewElmTypeCustom</u>	Benutzerdefinierte Objekte
<u>_WinStoViewElmTypeDialog</u>	Dialoge
<u>_WinStoViewElmTypeMenu</u>	Menüs
<u>_WinStoViewElmTypeMetaPicture</u>	Meta-Bilder
<u>_WinStoViewElmTypePicture</u>	Bilder
<u>_WinStoViewElmTypePrintDocRecord</u>	<u>PrintDocRecord-Objekte</u>
<u>_WinStoViewElmTypePrintDocument</u>	<u>PrintDoc-Objekte</u>
<u>_WinStoViewElmTypePrintForm</u>	<u>PrintForm-Objekte</u>
<u>_WinStoViewElmTypePrintFormList</u>	<u>PrintFormList-Objekte</u>
<u>_WinStoViewElmTypeProcedure</u>	Prozeduren
<u>_WinStoViewElmTypeTable</u>	Dateien
<u>_WinStoViewElmTypeText</u>	Texte
<u>_WinStoViewElmTypeTheme</u>	<u>Theme-Objekte</u>
<u>_WinStoViewElmTypeUser</u>	Benutzer
<u>_WinStoViewElmTypeUserGroup</u>	Benutzergruppen

Kontakt

UrmParentHandle Deskriptor des

Eltern-Objekts Typ handle

Siehe Liste, Objekte,

UrmParentName

Soll in dem StoList-Objekt untergeordnete Objekte angezeigt werden, kann in dieser Eigenschaft der Deskriptor des übergeordneten Objekts angegeben werden. Die Deskriptoren folgender Objekte können angegeben werden:

Objekt Angezeigte Objekte

User Benutzergruppen, in denen der Benutzer Mitglied ist.

UserGroup Mitglieder der Benutzergruppe.

ElmGroup Elemente der Elementgruppe.

Ist diese Eigenschaft gesetzt, wird die Eigenschaft StoView nicht ausgewertet.

Die Eigenschaft kann nur zur Laufzeit gesetzt werden.

Kontakt

UrmParentName



Name des Eltern-Objekts

Typ alpha

Liste, Objekte,

Siehe UrmParentHandle

Soll in dem StoList-Objekt untergeordnete Objekte angezeigt werden, kann in dieser Eigenschaft der Name des übergeordneten Objekts angegeben werden. Die Namen folgender Objekte können angegeben werden:

Objekt **Angezeigte Objekte**

User **Benutzergruppen, in denen der Benutzer Mitglied ist.**

UserGroup **Mitglieder der Benutzergruppe.**

ElmGroup **Elemente der Elementgruppe.**

Placement (Positions-Eigenschaften)

Eigenschaften

sortiert nach

Gruppen,

Siehe Alphabetische

Liste aller

Eigenschaften

Area

Position des Objektes

AreaBottom

Y-Position des Endpunktes vom Objekt

AreaClientRestore

Größe des Client-Bereichs herstellen

AreaHeight

Höhe des Objektes

AreaHeightCells

Höhe der Zellen

AreaHeightMax

Maximale Höhe des Objektes

AreaHeightMin

Mindesthöhe des Objektes

AreaId

ID des dynamischen Bereiches

AreaLeft

X-Position des Objektes

AreaMarginBottom

Unterer Abstand

AreaMarginLeft

Linker Rand

AreaMarginRight

Rechter Rand

AreaMarginTop

Oberer Rand

AreaRight

X-Position des Endpunktes vom Objekt

AreaScaling

Overlays mit Skalieren

AreaScalingFlags

Optionen zum Skalieren von Overlays

AreaScalingPageNo

Overlay an Seite verankern

AreaScreen

Bildschirmauflösung

AreaTop

Y-Position des Objektes

AreaWidth

Breite des Objektes

AreaWidthCells

Breite der Zellen

AreaWidthMax

Maximale Breite des Objektes

AreaWidthMin

Mindestbreite des Objektes

AreaWork

Arbeitsbereich

AutoSize

Automatisches Anpassen der Größe

Bound

Bound-Rechteck ermitteln

BoundAdd

Verbrauchten Platz auf Seite ermitteln

BoundMax

Maximal zu Verfügung stehenden Platz ermitteln

DialogBoxArrange

Ausrichtung der Dialogbox

DialogBoxLeft

Horizontale Position der Dialogbox

DialogBoxTop

Vertikale Position der Dialogbox

DividerSize

Höhe bzw. Breite der Splitter

DockSide

Position der Toolbar

FloatLeft

X-Koordinate der Toolbar

FloatTop

Y-Koordinate der Toolbar

FormHeight

Höhe der PrintForm

Kontakt

<u>FormWidth</u>	Breite der PrintForm
<u>Height</u>	Höhe des Objektes
<u>LockAreaSize</u>	Größe beim Verändern von Objektpositionen fixieren
<u>MarginBottom</u>	Unterer Seitenrand
<u>MarginLeft</u>	Linker Seitenrand
<u>MarginRight</u>	Rechter Seitenrand
<u>MarginTop</u>	Oberer Seitenrand
<u>Orientation</u>	Seitenformat Ausrichtung
<u>PageFormat</u>	Seitenformat
<u>PageFormatName</u>	Name des Seitenformates
<u>PageHeight</u>	Seitenhöhe
<u>PageWidth</u>	Seitenbreite
<u>Position</u>	Position des Objektes
<u>ScrollbarSize</u>	Höhe bzw. Breite der Scrollbars
<u>ScrollLeft</u>	Horizontale Scrollposition
<u>ScrollTop</u>	Vertikale Scrollposition
<u>ScrollWidth</u>	Breite des Scrollbereiches
<u>ScrollHeight</u>	Höhe des Scrollbereiches
<u>ShowPageOffset</u>	Seitenabstand anzeigen
<u>StyleGroup</u>	GroupTile-Anordnung
<u>Width</u>	Breite des Objektes

Area

Position des Objektes

Typ rect

Siehe Liste, Objekte

Über diese Eigenschaft kann die Position eines Objektes gesetzt werden. Die Position ist in einer rect-Struktur zu übergeben.

Die einzelnen Komponenten der Struktur entsprechen den folgenden Eigenschaften:

:left	<u>AreaLeft</u>	linker Rand
:top	<u>AreaTop</u>	oberer Rand
:right	<u>AreaRight</u>	rechter Rand
:bottom	<u>AreaBottom</u>	unterer Rand

Die im Designer angegebenen Eigenschaften **AreaWidth** und **AreaHeight** errechnen sich aus den angegebenen Eigenschaften und können prozedural nicht gesetzt werden.

Die Position wird relativ zum übergeordneten Objekt angegeben. Bei Oberflächen-Objekten wird die Position in Bildschirmpunkten erwartet, bei Druckformular-Objekten in logischen Einheiten.

Bei Frame-Objekten kann die Größe im normalen Status (siehe WinDialogNormal) über die Option WinAreaNormalState abgefragt und gesetzt werden.

Beispiel:

```
// Trotz minimiertem oder maximiertem Dialog die eigentliche Größe merktState # tFrame->WinInfo
```

Kontakt

AreaBottom



Y-Position des Endpunktes vom Objekt

Typ int

Siehe Liste, Objekte, AreaHeight,

Siehe LockAreaSize

Vertikale Position des Endpunktes vom Objekt.

Die Position wird relativ zur linken oberen Ecke des übergeordneten Objektes angegeben.

Der Wert in der Eigenschaft **AreaBottom** kann nicht kleiner sein als der Wert in **AreaTop**.

Soll ein Objekt nach oben verschoben werden, muss daher zunächst die Eigenschaft **AreaTop** und anschließend die Eigenschaft **AreaBottom** gesetzt werden. Ist die Eigenschaft **LockAreaSize** des **Application**-Objektes gesetzt, muss nur die Eigenschaft **AreaTop** gesetzt werden und die Eigenschaft **AreaBottom** wird automatisch angepasst.

Alternativ dazu kann die Eigenschaft **Area** auf die neuen Position gesetzt werden.

Bei Oberflächen-Objekten wird die Position in Bildschirmpunkten erwartet, bei Druckformular-Objekten in logischen Einheiten.

Kontakt

AreaClientRestore

Größe des Client-Bereichs herstellen

Typ logic
Siehe Liste, Objekte, WinPropGet(),

Frame-Objekte bestehen aus einem so genannten Non-Client-Bereich und einem Client-Bereich. Der Non-Client-Bereich enthält die Titelleiste, sowie den Fensterrand. Der Client-Bereich enthält die graphischen Objekte.

Die Area-Eigenschaften (AreaTop, AreaLeft, AreaRight und AreaBottom) geben die äußereren Abmessungen eines Fenster-Objekts an. Bei einer Veränderung z. B. der Höhe der Titelleiste über die Systemeinstellungen von Windows, verändert sich ebenfalls die Höhe des Client-Bereiches. Unter Umständen ist dies jedoch nicht gewünscht, da beispielsweise sonst Teile von Objekten die am unteren Rand stehen, abgeschnitten werden.

Beim Speichern eines Frame-Objekts im Designer, wird immer die Höhe des Client-Bereiches im Frame-Objekt gespeichert. Durch das Aktivieren der Eigenschaft **AreaClientRestore** in der Eigenschaftsliste wird die Größe des Client-Bereiches nach der Fenstererstellung wiederhergestellt (auch beim Öffnen des Frames im Designer).

Die Eigenschaft kann auch prozedural gesetzt werden, jedoch muss dies geschehen, bevor der Frame erzeugt wird, am besten direkt nach der Anweisung WinOpen() oder im Ereignis EvtInit des Frame-Objektes:

Beispiel:

```
tHdl # WinOpen(_WinOpenDialog);if (tHdl != 0){ // Die Anweisung weist das Frame-Objekt an, die G
```

AreaHeight

Höhe vom Objekt

Typ int

Liste, Objekte,

Siehe AreaBottom,

AreaWidth,

LockAreaSize

Diese Eigenschaft definiert die Höhe (siehe Position) des Objektes. Der Wert in der Eigenschaft **AreaHeight** kann nicht kleiner als 0 sein.

Bei Oberflächen-Objekten wird die Position in Bildschirmpunkten erwartet, bei

Druckformular-Objekten in logischen Einheiten.

Kontakt

AreaHeightCells |

Höhe der Zelle

Typ int

Liste, Objekte,

Siehe AreaWidthCells,

PrtPropGet(),

PrtPropSet()

Über diese Eigenschaft kann die Höhe des PrtGanttGraph- bzw. View-Objektes in Abhängigkeit von der Anzahl der Zellen (Zeilen) angegeben bzw. abgefragt werden.

Die tatsächliche Höhe ergibt sich aus den zu zeichnenden Zeilen und der Höhe der waagerechten Achsen.

Das Setzen der Eigenschaft ändert ebenfalls die Eigenschaft AreaBottom.

AreaHeightMax

Maximale Höhe des Objekts

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Über diese Eigenschaft kann die maximale Höhe des Frame- oder GroupTile-Objekts vorgegeben werden.

Bei der Angabe von -1 besteht kein Maximum. In der Entwicklungsumgebung kann durch Doppelklicken der Eigenschaft im Eigenschafts-Fenster die maximale Höhe auf die aktuelle Höhe des Fensters gesetzt werden.

AreaHeightMin

Mindesthöhe des Objekts

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Über diese Eigenschaft kann die Mindesthöhe eines Frame- oder GroupTile-Objekts vorgegeben werden.

Bei der Angabe von -1 besteht kein Minimum. In der Entwicklungsumgebung kann durch Doppelklicken der Eigenschaft im Eigenschafts-Fenster die minimale Höhe auf die aktuelle Höhe des Fensters gesetzt werden.

AreaId

ID des dynamischen Bereiches

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Diese Eigenschaft wird vom System vorgegeben und automatisch verwaltet, wenn GroupTile-Objekte zum GroupSplit-Objekt hinzugefügt oder aus diesem entfernt werden.

Mögliche Werte der Eigenschaft liegen im Bereich 0 bis zur Anzahl der aktuell vorhandenen GroupTile-Objekte.

Durch Setzen der Eigenschaft erhält das GroupTile-Objekt die Position und Größe des Bereiches, der durch die AreaId vorgegeben wird und wechselt damit seinen Platz mit dem dort befindlichen GroupTile-Objekt.

Kontakt

AreaLeft



X-Position des Objektes

Typ int

Liste, Objekte,

Siehe AreaRight,

AreaWidth,

LockAreaSize

Horizontale Position des Objektes.

Die Position wird relativ zur linken oberen Ecke des übergeordneten Objektes angegeben.

Ist die Eigenschaft LockAreaSize des Application-Objektes nicht gesetzt, kann der Wert der Eigenschaft AreaLeft nicht größer sein als der Wert in AreaRight. Soll ein Objekt nach rechts verschoben werden, muss daher zunächst die Eigenschaft AreaRight und anschließend die Eigenschaft AreaLeft gesetzt werden.

Ist die Eigenschaft LockAreaSize gesetzt, wird beim Setzen der Eigenschaft AreaLeft die Eigenschaft AreaRight ebenfalls angepasst und die Eigenschaft AreaWidth bleibt gleich. Andernfalls wird die Eigenschaft AreaWidth verändert und die Eigenschaft AreaRight bleibt gleich.

Alternativ dazu kann die Eigenschaft Area auf die neue Position gesetzt werden.

Bei Oberflächen-Objekten wird die Position in Bildschirmpunkten erwartet, bei Druckformular-Objekten in logischen Einheiten.

Kontakt

AreaMarginBottom| |

Unterer Abstand

Typ int

Liste, Objekte,

WinPropGet(),

Siehe WinPropSet(),

PrtPropGet(),

PrtPropSet()

Diese Eigenschaft ist bei einigen Oberflächen- und Druck-Objekten vorhanden. Die Bedeutung der Eigenschaft ist unterschiedlich:

- GroupSplit-Objekt

Die Eigenschaft definiert beim GroupSplit-Objekt den unteren Rand zu den untergeordneten GroupTile-Objekten. Die Angabe erfolgt in Pixeln.

- Spalte von RecView-Objekten

Bei Spalte von RecView-Objekten definiert die Eigenschaft den unteren Abstand des Inhaltes zur Zellgrenze. Die Angabe erfolgt in Pixeln.

- Druck-Objekte

Die Eigenschaft definiert den unteren Seitenrand in logischen Einheiten.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() sowie PrtPropGet() und PrtPropSet() gelesen bzw. geändert werden.

Beispiel:

```
tGroupSplit->wpAreaMarginBottom # 2;tColumn->wpAreaMarginBottom # 4;tPage->ppAreaMarginBottom # P
```

Kontakt

AreaMarginLeft | |

Linker Rand

Typ int

Liste, Objekte,

WinPropGet(),

Siehe WinPropSet(),

PrtPropGet(),

PrtPropSet()

Diese Eigenschaft ist bei einigen Oberflächen- und Druck-Objekten vorhanden. Die Bedeutung der Eigenschaft ist unterschiedlich:

- GroupSplit-Objekt

Die Eigenschaft definiert beim GroupSplit-Objekt den linken Rand zu den untergeordneten GroupTile-Objekten. Die Angabe erfolgt in Pixeln.

- Spalte von RecView-Objekten

Bei Spalte von RecView-Objekten definiert die Eigenschaft den linken Abstand des Inhaltes zur Zellgrenze. Die Angabe erfolgt in Pixeln.

- Druck-Objekte

Die Eigenschaft definiert den linken Seitenrand in logischen Einheiten.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() sowie PrtPropGet() und PrtPropSet() gelesen bzw. geändert werden.

Beispiele:

```
tGroupSplit->wpAreaMarginLeft # 2;tColumn->wpAreaMarginLeft # 4;tPage->ppAreaMarginLeft # PrtUnit
```

AreaMarginRight | |

Rechter Rand

Typ int

Liste, Objekte,

WinPropGet(),

Siehe WinPropSet(),

PrtPropGet(),

PrtPropSet()

Diese Eigenschaft ist bei einigen Oberflächen- und Druck-Objekten vorhanden. Die Bedeutung der Eigenschaft ist unterschiedlich:

- GroupSplit-Objekt

Die Eigenschaft definiert beim GroupSplit-Objekt den rechten Rand zu den untergeordneten GroupTile-Objekten. Die Angabe erfolgt in Pixeln.

- Spalte von RecView-Objekten

Bei Spalte von RecView-Objekten definiert die Eigenschaft den rechten Abstand des Inhaltes zur Zellgrenze. Die Angabe erfolgt in Pixeln.

- **Druck-Objekte**

Die Eigenschaft definiert den rechten Seitenrand in logischen Einheiten.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() sowie PrtPropGet() und PrtPropSet() gelesen bzw. geändert werden.

Beispiele:

```
tGroupSplit->wpAreaMarginRight # 2;tColumn->wpAreaMarginRight # 4;tPage->ppAreaMarginRight # PrtU
```

Kontakt

AreaMarginTop| | |

Oberer Rand

Typ int

Liste,

Objekte,

Siehe WinPropGet(),

WinPropSet(),

PrtPropGet(),

PrtPropSet()

Diese Eigenschaft ist bei einigen Oberflächen- und Druck-Objekten vorhanden. Die Bedeutung der Eigenschaft ist unterschiedlich:

- GroupSplit-Objekt

Die Eigenschaft definiert beim GroupSplit-Objekt den oberen Rand zu den untergeordneten GroupTile-Objekten. Die Angabe erfolgt in Pixeln.

- Spalte von RecView-Objekten

Bei Spalte von RecView-Objekten definiert die Eigenschaft den oberen Abstand des Inhaltes zur Zellgrenze. Die Angabe erfolgt in Pixeln.

- Druck-Objekte

Die Eigenschaft definiert den oberen Seitenrand in logischen Einheiten.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() sowie PrtPropGet() und PrtPropSet() gelesen bzw. geändert werden.

Beispiele:

```
tGroupSplit->wpAreaMarginTop # 2;tColumn->wpAreaMarginTop # 4;tPage->ppAreaMarginTop # PrtUnitLog
```

Kontakt

AreaRight



X-Position des Endpunktes vom Objekt

Typ int

Siehe Liste, Objekte, AreaWidth,

LockAreaSize

Horizontale Position des Endpunktes vom Objekt.

Die Position wird relativ zur linken oberen Ecke des übergeordneten Objektes angegeben.

Der Wert in der Eigenschaft AreaRight kann nicht kleiner sein als der Wert in AreaLeft.

Soll ein Objekt nach links verschoben werden, muss daher zunächst die Eigenschaft AreaLeft und anschließend die Eigenschaft AreaRight gesetzt werden. Ist die Eigenschaft LockAreaSize des Application-Objektes gesetzt, muss nur die Eigenschaft AreaLeft gesetzt werden und die Eigenschaft AreaRight wird automatisch angepasst.

Alternativ dazu kann die Eigenschaft Area auf die neue Position gesetzt werden.

Bei Oberflächen-Objekten wird die Position in Bildschirmpunkten erwartet, bei Druckformular-Objekten in logischen Einheiten.

AreaScaling

Overlays mit Skalieren

Typ logic
Siehe Liste, Objekte,

Diese Eigenschaft kann nur gesetzt werden, wenn das Objekt als Overlay in einem übergeordneten Objekt erzeugt wurde.

Wird das Bild oder die Seiten im übergeordneten Objekt vergrößert, verkleinert oder in seiner Position verschoben, wirkt sich das nicht auf das Overlay aus. Soll das Overlay mit skaliert oder verschoben werden, muss die Eigenschaft AreaScaling gesetzt werden. Dadurch verbleibt das Overlay an der gleichen Position innerhalb des übergeordneten Bildes oder Seite. D. h. die Position des Overlays verschiebt sich zusammen mit dem Inhalt des übergeordneten Objekts.



Beim Setzen der Eigenschaft AreaScaling wird die aktuelle Position des Overlays gegenüber dem Elternobjekt und der Zoomfaktor des Elternobjektes in internen Eigenschaften gespeichert. Wird das Overlay anschließend verschoben, hat es keine Auswirkung auf die Ausrichtung am Elternobjekt. Um die Position zu verändern muss die Eigenschaft AreaScaling zunächst wieder deaktiviert werden.

Soll das Overlay immer die gleiche Position und Größe haben, darf die Eigenschaft nicht gesetzt sein.

AreaScalingFlags

Optionen zum Skalieren von Overlays

Typ [int](#)

Siehe [Liste](#), [Objekte](#), [AreaScaling](#)

Diese Eigenschaft kann nur gesetzt werden, wenn das Objekt als Overlay in einem übergeordneten Objekt erzeugt wurde.

In dieser Eigenschaft kann nur die Ausprägung `_WinAreaScalingMoveInsidePage` gesetzt werden. Ist diese Ausprägung gesetzt, werden Overlays immer vollständig innerhalb einer Seite positioniert. Dies erfolgt automatisch, wenn die Eigenschaft [Visible](#) des [DocView](#)-Objekts auf `true` gesetzt oder das Overlay positioniert (siehe [Design](#) und [Area](#)) wird. Das Flag kann mit der Ausprägung `_WinAreaScalingNone` zurückgesetzt werden.



Die Eigenschaft wird nur ausgewertet, wenn die Eigenschaft [AreaScaling](#) gesetzt ist.

AreaScalingPageNo

Overlay an Seite verankern

Typ int

Siehe Liste, Objekte

Diese Eigenschaft kann nur gesetzt werden, wenn das Objekt als Overlay in einem übergeordneten Objekt erzeugt wurde.

Die Area-Eigenschaften geben die Position des Objekts an. Der Anker dieser Positionierung ist die linke obere Ecke des übergeordneten Objekts. Wird das Overlay in einem DocView-Objekt angezeigt, kann es auch an einer Seite des angezeigten Dokuments verankert werden. Die Seitennummer wird in dieser Eigenschaft angegeben. Wird innerhalb des Dokuments gescrollt, bewegt sich das Overlay mit der Seite mit.

AreaScreen

Bildschirmauflösung

Typ rect

Liste, Objekte,

Siehe AreaWork,

WinPropGet()

Über diese Eigenschaft kann die Größe des Primärbildschirms in Pixeln ermittelt werden. Das Resultat ist ein Wert vom Typ rect, wobei die Komponenten left und top beim Primärbildschirm immer 0 sind.

Der ermittelte Wert entspricht den Rückgabewerten von WinInfo(_WinScreenWidth) und WinInfo(_WinScreenHeight).

Optional kann der Bildschirmbereich für andere Bildschirme ermittelt werden. Hierzu wird die Nummer (≥ 0) des Monitors in Klammern angegeben. Der Primärbildschirm hat immer die Nummer 0.

Beispiel

```
// Bildschirmbereich des Primärbildschirms tRect # _App->wpAreaScreen;tRect # _App->wpAreaScreen(0
```



Die Eigenschaft kann nur ausgelesen werden.

Mögliche Laufzeitfehler:

ErrValueInvalid Ein Bildschirm mit der in Klammern angegebenen Nummer ist nicht vorhanden.

Kontakt

AreaTop



Y-Position des Objektes

Typ int

Liste, Objekte,

Siehe AreaBottom,

AreaHeight,

LockAreaSize

Vertikale Position des Objektes.

Die Position wird relativ zur linken oberen Ecke des übergeordneten Objektes angegeben.

Ist die Eigenschaft LockAreaSize des Application-Objektes nicht gesetzt, kann der Wert der Eigenschaft AreaTop nicht größer sein als der Wert in AreaBottom. Soll ein Objekt nach unten verschoben werden, muss daher zunächst die Eigenschaft AreaBottom und anschließend die Eigenschaft AreaTop gesetzt werden.

Ist die Eigenschaft LockAreaSize gesetzt, wird beim Setzen der Eigenschaft AreaTop die Eigenschaft AreaBottom ebenfalls angepasst und die Eigenschaft AreaHeight bleibt gleich. Andernfalls wird die Eigenschaft AreaHeight verändert und die Eigenschaft AreaBottom bleibt gleich.

Alternativ dazu kann die Eigenschaft Area auf die neue Position gesetzt werden.

Bei Oberflächen-Objekten wird die Position in Bildschirmpunkten erwartet, bei Druckformular-Objekten in logischen Einheiten.

AreaWidth

Breite vom Objekt

Typ int

Liste, Objekte,

Siehe AreaRight,

AreaHeight,

LockAreaSize

Diese Eigenschaft definiert die Breite (siehe Position) des Objektes. Der Wert in der Eigenschaft **AreaWidth** kann nicht kleiner als 0 sein.

Bei Oberflächen-Objekten wird die Position in Bildschirmpunkten erwartet, bei

Druckformular-Objekten in logischen Einheiten.

Kontakt

AreaWidthCells

Breite der Zelle

Typ int

Liste, Objekte,

Siehe AreaHeightCells,

PrtPropGet(),

PrtPropSet()

Über diese Eigenschaft kann die Breite des PrtGanttGraph- bzw. View-Objektes in Abhängigkeit von der Anzahl der Zellen (Spalten) angegeben bzw. abgefragt werden.

Die tatsächliche Breite ergibt sich aus den zu zeichnenden Spalten und der Breite der senkrechten Achsen.

Das Setzen der Eigenschaft ändert ebenfalls die Eigenschaft AreaRight.

AreaWidthMin

Mindestbreite des Objekts

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Über diese Eigenschaft kann die Mindestbreite eines Frame- oder GroupTile-Objekts vorgegeben werden.

Bei der Angabe von -1 besteht kein Minimum. In der Entwicklungsumgebung kann durch Doppelklicken der Eigenschaft im Eigenschafts-Fenster die minimale Breite auf die aktuelle Breite des Fensters gesetzt werden.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

AreaWidthMax

Maximale Breite des Objektes

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Über diese Eigenschaft kann die maximale Breite des Frame- oder GroupTile-Objekts vorgegeben werden.

Bei der Angabe von -1 besteht kein Maximum. In der Entwicklungsumgebung kann durch Doppelklicken der Eigenschaft im Eigenschafts-Fenster die maximale Breite auf die aktuelle Breite des Fensters gesetzt werden.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

AreaWork



Arbeitsbereich

Typ rect

Liste, Objekte,

Siehe AreaScreen,

WinPropGet()

Mit dieser Eigenschaft kann die zur Verfügung stehende Arbeitsfläche des Primärbildschirms in Pixeln ermittelt werden. Die Arbeitsfläche besteht aus der Bildschirmauflösung abzüglich der Taskleiste. Das Resultat ist ein Wert vom Typ rect, wobei die Komponenten left und top beim Primärbildschirm immer 0 sind.

Optional kann die Arbeitsfläche für andere Bildschirme ermittelt werden. Hierzu wird die Nummer (≥ 0) des Monitors in Klammern angegeben. Der Primärbildschirm hat immer die Nummer 0.

Beispiel

```
// Arbeitsbereich des Primärbildschirms tRect # _App->wpAreaWork;tRect # _App->wpAreaWork(0); // Ar
```



Die Eigenschaft kann nur ausgelesen werden.

Mögliche Laufzeitfehler:

ErrValueInvalid Ein Bildschirm mit der in Klammern angegebenen Nummer ist nicht vorhanden.

AutoSize

Automatisches Anpassen der Größe

Typ logic

Liste, Objekte, WinPropGet(),

Siehe WinPropSet(), PrtPropGet(),

PrtPropSet()

Wenn diese Eigenschaft auf true gesetzt ist, wird die Größe des Objektes an den Platzbedarf der Schrift bzw. des Objektes angepasst.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() / PrtPropGet() und WinPropSet() / PrtPropSet() gelesen bzw. geändert werden.

Kontakt

Bound



Bound-Rechteck ermitteln

Typ rect
Siehe [Liste, Objekte,](#)
[PrtPropGet\(\)](#)

Unter dem Bound-Rechteck wird das minimale Rechteck verstanden, das alle direkten untergeordneten Objekte eines gegebenen Objektes umschließt. Die Eigenschaft liefert diese Koordinaten in logischen Einheiten.

Die Eigenschaft kann über den Befehl [PrtPropGet\(\)](#) gelesen werden.

Kontakt

BoundAdd



Verbrauchten Platz auf Seite ermitteln

Typ point
Siehe Liste, Objekte, BoundMax,

PrtPropGet(), Beispiel

Die Eigenschaft liefert den bereits durch die Befehle PrtAdd() und PrtAddByName() verbrauchten Platz auf einer Seite in logischen Einheiten.

Die Eigenschaft kann über den Befehl PrtPropGet() gelesen werden.

Beispiel

```
// Bereits verbrauchter Platz auf Seite tAddSize # tPage->ppBoundAdd;// Maximal zu Verfügung stehe
```

Kontakt

BoundMax Maximal zu Verfügung stehenden Platz
ermitteln Typ point

Siehe [Liste](#), [Objekte](#), [BoundAdd](#), [PrtPropGet\(\)](#),

Beispiel

Die Eigenschaft liefert den maximal zur Verfügung stehenden Platz innerhalb eines Seitenbereichs in logischen Einheiten. Der Seitenbereich kann als Parameter angegeben werden.

Ohne weitere Angabe von Parametern wird der Platz auf der Seite zurückgegeben. Der Platz in anderen Seitenbereichen ermittelt werden, können folgende Konstanten übergeben werden:

- PrtBoundPage Druckbereich (default)
- PrtBoundHeader Seitenkopf
- PrtBoundFooter Seitenfuß

Die Eigenschaft kann über den Befehl PrtPropGet() gelesen werden.

Beispiel:

```
// Bereits verbrauchter Platz im Druckbereich der SeitenAddSize # tPage->ppBoundAdd; // Identisch
```

DialogBoxArrange

Ausrichtung der Dialogbox

Typ int

Liste, Objekte,
DialogBoxLeft,

Siehe DialogBoxTop,
WinPropGet(),
WinPropSet()

In dieser Eigenschaft wird die Ausrichtung von Dialog-Boxen nachfolgender WinDialogBox()-Aufrufe bestimmt.

Diese Eigenschaft wirkt sich nur aus, wenn die Eigenschaften DialogBoxLeft und DialogBoxTop den Wert **-1** haben.

Folgende Werte sind zulässig:

WinDialogCenter

– DialogBox zentriert zum Elternfenster anzeigen.

WinDialogCenterX

– DialogBox horizontal zentriert zum Elternfenster anzeigen.

WinDialogCenterY

– DialogBox vertikal zentriert zum Elternfenster anzeigen.

WinDialogCenterScreen

– DialogBox zentriert zum Bildschirm anzeigen.

WinDialogCenterScreenX

– DialogBox horizontal zentriert zum Bildschirm anzeigen.

WinDialogCenterScreenY

– DialogBox vertikal zum Bildschirm anzeigen.

WinDialogCenterNone

– DialogBox vom Betriebssystem positionieren lassen
(Standard).

DialogBoxLeft

Horizontale Position der Dialog-Box

Typ int
Siehe Liste, Objekte, DialogBoxTop,

In dieser Eigenschaft wird die horizontale Position von Dialog-Boxen nachfolgender WinDialogBox()-Aufrufe bestimmt.

Enthält die Eigenschaft den Wert -1, übernimmt das Betriebssystem die Positionierung der Dialogbox.

Bei einem Wert größer oder gleich 0, wird die Dialogbox entsprechend der Koordinate positioniert.

Liegt die Position außerhalb des primären Bildschirms, wird die Dialogbox entsprechend justiert.

Die vertikale Position wird in der Eigenschaft DialogBoxTop angegeben.

DialogBoxTop

Vertikale Position der Dialog-Box

Typ int
Siehe Liste, Objekte, DialogBoxLeft,

In dieser Eigenschaft wird die vertikale Position von Dialog-Boxen nachfolgender WinDialogBox()-Aufrufe bestimmt.

Enthält die Eigenschaft den Wert -1, übernimmt das Betriebssystem die Positionierung der Dialogbox.

Bei einem Wert größer oder gleich 0, wird die Dialogbox entsprechend der Koordinate positioniert.

Liegt die Position außerhalb des primären Bildschirms, wird die Dialogbox entsprechend justiert.

Die horizontale Position wird in der Eigenschaft DialogBoxLeft angegeben.

DividerSize

Höhe bzw. Breite der Splitter

Typ int

Liste, Objekte,

Siehe WinPropGet(),
WinPropSet()

Diese Eigenschaft definiert die Höhe bzw. Breite der Splitter in den folgenden Objekten:

- CodeEdit
- DataList
- DocView
- GanttGraph
- GroupSplit
- RecList
- RecView
- StoList

Folgende Werte sind möglich:

_WinDividerSizeDefault Stellt einen normalen Divider dar. Dies ist der voreingestellte Wert.

_WinDividerSizeTouch Stellt einen breiteren bzw. höheren Divider, zur besseren Interaktion bei Touch-Bedienung, dar.

Kontakt

DockSide



Position der Toolbar

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Diese Eigenschaft bestimmt die Position der Toolbar.

Es gibt folgende Positionen:

WinDockTop obere Fensterseite

WinDockRight rechte Fensterseite

WinDockBottom untere Fensterseite

WinDockBottomTop untere Fensterseite

WinDockLeft linke Fensterseite

FloatLeft

x-Koordinate der Toolbar

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Diese Eigenschaft gibt die x-Koordinate an, die die Toolbar beim Lösen aus dem ToolbarDock-Objekt einnehmen soll.

Kontakt

FloatTop



Y-Koordinate der Toolbar

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Diese Eigenschaft gibt die y-Koordinate an, die die Toolbar beim Lösen aus dem ToolbarDock-Objekt einnehmen soll.

FormHeight

Höhe der PrintForm / des Canvas

Typ int

Siehe Liste, Objekte, Beispiel

Die Eigenschaft definiert die Höhe der PrintForm bzw. des Zeichenbereiches des Canvas-Objektes in logischen Einheiten. Beim Setzen der Eigenschaft ist zu beachten, dass diese Höhe nicht kleiner werden kann, als die durch die direkt untergeordneten Objekte definierte Höhe. Die Höhe wird durch das Hinzufügen bzw. Verschieben untergeordneter Objekte automatisch erweitert.

Wird die Höhe eines PrintFormList-Objektes ausgelesen, können mit der Option

PrtLstUpdate (`tHeight # tForm->ppFormHeight(_PrtLstUpdate)`) alle PrtLine-Objekte neu angeordnet und die Höhe aller PrtLine-Objekte neu berechnet werden. Anschließend wird die neue Höhe des PrintFormList-Objektes (MarginTop + Höhe der PrtLine-Objekte) zurückgegeben. Diese Höhe kann z. B. bei der manuellen Ermittlung des Seitenumbruchs durch PrtAdd() herangezogen werden.

Kontakt

FormWidth



Breite der PrintForm / des Canvas

Typ **int**

Siehe [Liste, Objekte](#)

Die Eigenschaft definiert die Breite der [PrintForm](#) bzw. des Zeichenbereiches des [Canvas-Objektes](#) in [logischen Einheiten](#). Beim Setzen der Eigenschaft ist zu beachten, dass diese Breite nicht kleiner werden kann, als die durch die direkt untergeordneten Objekte definierte Breite. Die Breite wird durch das Hinzufügen bzw. Verschieben untergeordneter Objekte automatisch erweitert.

Kontakt

HdrHeight

Höhe der Spaltenköpfe

Typ int

HdrVerticalText,

HdrWordBreak,

Siehe Liste, Objekte,

WinPropGet(),

WinPropSet()

Der Wert der Eigenschaft bestimmt die Höhe des Spaltenkopfes in Pixel. Standardmäßig enthält die Eigenschaft den Wert 0, wodurch die Standardhöhe gesetzt wird. Beim Setzen der Eigenschaft auf negative Werte wird der Laufzeitfehler ErrValueInvalid ausgelöst.

Kontakt

Objekte von HdrHeight

Folgende Objekte unterstützen die Eigenschaft [HdrHeight](#)

[HdrHeight](#),

[Liste der](#)

Siehe [Objekte](#), [Liste](#)

[der](#)

[Eigenschaften](#)

- [DataList](#)
- [DataListPopup](#)
- [RecList](#)
- [RecListPopup](#)
- [RecView](#)
- [StoList](#)
- [StoListPopup](#)

Kontakt

Height

Höhe Typ

int

Liste,
Objekte,
Siehe WinPropGet(),
WinPropSet(),
PrtPropGet(),
PrtPropSet()

- GroupTile-Objekt

Die Eigenschaft bestimmt die Höhe des Objekts:

Ist der Wert ≥ 0 wird die prozentuale Höhe zum gesamten Darstellungsbereich in 1/100 Prozent festgelegt.

Ist der Wert < 0 wird hierdurch die absolute Höhe in Pixel angegeben. Das GroupTile-Objekt kann bei Angabe der absoluten Höhe vom Benutzer nicht mehr in der Höhe variiert werden.

Beispiele:

```
$GroupTile->wpHeight # 2000; // 20.00 Prozent$GroupTile->wpHeight # -120; // 120 Pixel
```

- PrtMetaPicture-Objekt

Beim PrtMetaPicture-Objekt kann über die Eigenschaft die Originalhöhe des Bildes in logischen Einheiten definiert werden, sofern es sich um ein "Windows Meta File" handelt.

- PrtTblCell-Objekt

Beim PrtTblCell-Objekt kann über die Eigenschaft die Zellhöhe in logischen Einheiten definiert werden.

Kontakt

LockAreaSize Größe beim Verändern von Objektpositionen
fixieren Typ logic

Siehe Liste, Objekte, AreaLeft, AreaTop

Ist diese Eigenschaft gesetzt, wird bei einer Änderung der Eigenschaften AreaLeft bzw. AreaTop von Objekten die Größe des Objektes beibehalten. Statt dessen wird AreaRight bzw. AreaBottom verändert. Ist diese Eigenschaft nicht gesetzt, ändern sich automatisch die Eigenschaften AreaWidth und AreaHeight. Standardmäßig ist diese Eigenschaft bei Prozedurausführung auf false gesetzt.

Beispiel:

```
// Objekt verschieben_App->wpLockAreaSize # true;$edObject->wpAreaLeft # 10;// => AreaRight wird
```

MarginLeft

Linker Randabstand

Typ **int**

Siehe [Liste, Objekte](#)

Diese Eigenschaft ist bei einigen Oberflächen- und Druck-Objekten vorhanden. Die Bedeutung der Eigenschaft ist unterschiedlich:

- **Eingabe-Objekte**

Die Eigenschaft legt den Abstand (in Pixel) des Inhaltes eines Eingabe-Objekts zum linken Rand fest.

Bei Angabe von 0 besteht kein Abstand zum linken Rand des Eingabe-Objekts. Bei Angabe von -1 wird der Abstand verwendet, der in dem gewählten [Font](#) als Standard definiert ist.

- **Druck-Objekte**



Diese Eigenschaft wurde bei den Druck-Objekten durch [AreaMarginLeft](#) abgelöst und sollte nicht mehr verwendet werden.

Die Eigenschaft definiert den linken Seitenrand in [logischen Einheiten](#).

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle [WinPropGet\(\)](#) / [PrtPropGet\(\)](#) und [WinPropSet\(\)](#) / [PrtPropSet\(\)](#) gelesen bzw. geändert werden.

Beispiele:

```
tEdit->wpMarginLeft # 12;
```

Kontakt

Orientation



Seitenformat Ausrichtung

Typ int

Liste, Objekte,

Siehe PrtPropGet(),

PrtPropSet()

Diese Eigenschaft bestimmt die Seitenausrichtung.

PrtOrientPortrait Hochformat

PrtOrientLandscape Querformat

Die Eigenschaft kann beim PpcObject nur ausgelesen werden.

Beispiel:

```
tJob # PrtJobOpen(_PrtDocDinA4, '', _PrtJobOpenWrite | _PrtJobOpenTemp);if (tJob > 0){ // PrintD
```

Kontakt

PageFormat



Seitenformat

Typ int

Liste, Objekte,
PageFormatName,

Siehe Benutzerdefinierte

Druckformate
(Blog)

Die Eigenschaft legt das Seitenformat des oder der Seitenobjekte im PrintDoc- oder PrintDocRecord-Objekt fest.

PrtPaperFormatUser
PrtPaperFormatLetter
PrtPaperFormatLetterSmall
PrtPaperFormatTabloid
PrtPaperFormatLedger
PrtPaperFormatLegal
PrtPaperFormatStatement
PrtPaperFormatExecutive
PrtPaperFormatA3
PrtPaperFormatA4
PrtPaperFormatA4Small
PrtPaperFormatA5
PrtPaperFormatB4
PrtPaperFormatB5
PrtPaperFormatFolio
PrtPaperFormatQuarto
PrtPaperFormat10x14
PrtPaperFormat11x17
PrtPaperFormatNote
PrtPaperFormatEnv9
PrtPaperFormatEnv10
PrtPaperFormatEnv11
PrtPaperFormatEnv12
PrtPaperFormatEnv14
PrtPaperFormatCSheet
PrtPaperFormatEnvDL
PrtPaperFormatEnvC5
PrtPaperFormatEnvC3
PrtPaperFormatEnvC4
PrtPaperFormatEnvC6
PrtPaperFormatEnvC65
PrtPaperFormatEnvB4

Benutzerdefiniertes Seitenformat

Letter (8.5 x 11 inch)

Letter Small (8.5 x 11 inch)

Tabloid (11 x 17 inch)

Ledger (17 x 11 inch)

Legal (8.5 x 14 inch)

Statement (5.5 x 8.5 inch)

Executive (7.25 x 10.5 inch)

A3 (297 x 410 mm)

A4 (210 x 297 mm)

A4 (210 x 297 mm)

A5 (148 x 210 mm)

B4 (250 x 354 mm)

B5 (182 x 257 mm)

Folio (8.5 x 13 inch)

Quarto (215 x 275 mm)

(10 x 14 inch)

(11 x 17 inch)

Note (8.5 x 11 inch)

Envelope #9 (3 7/8 x 8 7/8 inch)

Envelope #10 (4 1/8 x 9.5 inch)

Envelope #11 (4.5 x 10 3/8 inch)

Envelope #12 (4.75 x 11 inch)

Envelope #14 (5 x 11.5 inch)

C size sheet (17 x 22 inch)

Umschlag DL (110 x 220 mm)

Umschlag C5 (162 x 229 mm)

Umschlag C3 (324 x 258 mm)

Umschlag C4 (229 x 324 mm)

Umschlag C6 (114 x 162 mm)

Umschlag C65 (114 x 229 mm)

Umschlag B4 (250 x 125 mm)

Kontakt

<u>PrtPaperFormatEnvB5</u>	Umschlag B5 (176 x 125 mm)
<u>PrtPaperFormatEnvB6</u>	Umschlag B6 (176 x 125 mm)
<u>PrtPaperFormatEnvItaly</u>	Umschlag (110 x 230 mm)
<u>PrtPaperFormatEnvMonarch</u>	Umschlag Monarch (3 7/8 x 7.5 inch)
<u>PrtPaperFormatPersonal</u>	Umschlag 6 3/4 (3 5/8 x 6.5 inch)
<u>PrtPaperFormatFanfoldUS</u>	US Std Fanfold (14 7/8 x 11 inch)
<u>PrtPaperFormatFanfoldStdDE</u>	German Std Fanfold (8.5 x 12 inch)
<u>PrtPaperFormatFanfoldLglDE</u>	German Legal Fanfold (8.5 x 13 inch)

Bei den Objekten PrintDevice und PpcObject kann die Eigenschaft nur ausgelesen werden und bestimmt das beim Drucker eingestellte Papierformat bzw. das Format mit dem ein Druckjob gedruckt wurde.

Kontakt

PageFormatName

Name des Seitenformates

Typ [alpha\(8192\)](#)

Siehe [Liste, Objekte](#)

Die Eigenschaft legt das Seitenformat des Druckers ([PrintDevice](#)-Objekt) fest.



Das Setzen der Eigenschaft ändert ebenfalls die Eigenschaft [PageFormat](#).

Der Name kann zuvor mit dem Befehl [PrtInfoStr\(\)](#) ermittelt werden. Die Namen der Papierformate sind im entsprechenden Windows-Druckertreiber hinterlegt. Je nach Version und Sprache des Druckertreibers können unterschiedliche Namen vorliegen. Gerade in einer mehrsprachigen Umgebung sollten die Namen über den Befehl [PrtInfoStr\(\)](#) ermittelt werden.

Der Name des Papierformates kann mit Wildcards (*) und (?) angegeben werden.

Bei dem Objekt [PpcObject](#) kann diese Eigenschaft nur ausgelesen werden und bestimmt mit welchem Format der Druckjob gedruckt wurde.

Beispiele:

'A4' Setzt das Format A4 bei diesem Drucker.

'*Postkarte*' Setzt das Format auf das erste Format, das die Zeichenkette 'Postkarte' beinhaltet.

Kontakt

PageHeight 

Seitenhöhe

Typ [int](#)

Siehe [Liste](#),

Siehe [Objekte](#)

Die Eigenschaft definiert die Seitenhöhe in [logischen Einheiten](#).

Die maximale Seitenhöhe sind 44 inch (ca. 112 cm).



Diese Eigenschaft kann beim [PrintDevice](#) nur gelesen werden.

Kontakt

PageWidth

Seitenbreite

Typ int

Siehe Liste,

Objekte

Die Eigenschaft definiert die Seitenbreite in logischen Einheiten.

Die maximale Seitenbreite sind 22 inch (ca. 56 cm).

Beim RtfEdit-Objekt wird über diese Eigenschaft die Breite des Textes definiert. Der Text wird automatisch an der angegebenen Breite umgebrochen. Ist das Objekt kleiner als die angegebene Breite, erscheint ein horizontaler Scrollbalken, sobald Text eingegeben wurde, der breiter als das Eingabe-Objekt ist. Wird die Breite des Textes mit 0 angegeben, erfolgt der Umbruch an der Breite des Objekts. Der Umbruch beim Drucken wird durch die Breite des entsprechenden PrtRtf-Objekts bestimmt.



Diese Eigenschaft kann beim PrintDevice nur gelesen werden.

Kontakt

Position / Area-Eigenschaften

Position des Objekts

[Area](#), [AreaTop](#), [AreaLeft](#),

Siehe [AreaRight](#), [AreaBottom](#),

[AreaWidth](#), [AreaHeight](#)

Die Objektposition wird durch den Anfangspunkt ([AreaLeft](#), [AreaTop](#)) und durch den Endpunkt ([AreaRight](#), [AreaBottom](#)) angegeben. Alternativ kann die Größe über die Eigenschaften [AreaWidth](#) und [AreaHeight](#) angegeben werden.

Diese zwei Punkte sind in Bildpunkten (Pixel) relativ zu dem übergeordneten Objekt anzugeben.

Beim Anwendungsfenster sind diese Punkte die Bildschirmkoordinaten.

Der Anfangspunkt kann über die Eigenschaften [AreaLeft](#) und [AreaTop](#) gesetzt und gelesen werden.

Der Endpunkt kann über die Eigenschaften [AreaRight](#) und [AreaBottom](#) gesetzt und gelesen werden.

Die Größe kann über die Eigenschaften [AreaWidth](#) und [AreaHeight](#) gesetzt und gelesen werden.

Bei druckbaren Objekten erfolgt die Angabe von Koordinaten zwar auch relativ zum übergeordneten Container-Objekt, jedoch ist die Einheit 1/100 twips. Ein Twip entspricht einem 1/1440 inch. Die Koordinaten können mit den Anweisungen [PrtUnit\(\)](#) und [PrtUnitLog\(\)](#) umgerechnet werden.

ScrollbarSize

Höhe bzw. Breite der Scrollbars

Typ int

Siehe Liste, Objekte,

WinPropGet(), WinPropSet()

Diese Eigenschaft definiert die Höhe bzw. Breite der Scrollbars in den folgenden Objekten:

- DataList
- DataListPopup
- DocView
- GanttGraph
- RecList
- RecListPopup
- RecView
- StoList
- StoListPopup

Folgende Werte sind möglich:

_WinScrollbarSizeDefault Vom System definierte Scrollbar-Größe. Dies ist der voreingestellte Wert.

_WinScrollbarSizeTouch Stellt eine breitere bzw. höhere Scrollbar, zur besseren Interaktion bei Touch-Bedienung, dar.

ScrollLeft

Horizontale Scrollposition

Typ int

Liste, Objekte,
WinPropGet(),

Siehe WinPropSet(),

ScrollTop,
ScrollWidth,
ScrollHeight

Definiert die aktuelle horizontale Scrollposition in Pixel. Beim Setzen wird die Gültigkeit der Position überprüft und gegebenenfalls korrigiert. Die Eigenschaft ändert sich automatisch, wenn der Benutzer die Scroll-Position über die Scrollbars anpasst.

Die Eigenschaft kann über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.



Mit dem Befehl WinBoxScrollVisible() kann ein Oberflächenobjekt in den sichtbaren Bereich gescrollt werden.

ScrollTop

Vertikale Scrollposition

Typ int

Liste, Objekte,
WinPropGet(),

Siehe WinPropSet(),

ScrollLeft,
ScrollWidth,
ScrollHeight

Definiert die aktuelle vertikale Scrollposition in Pixel. Beim Setzen wird die Gültigkeit der Position überprüft und gegebenenfalls korrigiert. Die Eigenschaft ändert sich automatisch, wenn der Benutzer die Scroll-Position über die Scrollbars anpasst.

Die Eigenschaft kann über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.



Mit dem Befehl WinBoxScrollVisible() kann ein Oberflächenobjekt in den sichtbaren Bereich gescrollt werden.

ScrollWidth

Breite des Scrollbereiches

Typ int

Liste, Objekte,

WinPropGet(),

Siehe WinPropSet(),

ScrollHeight,

ScrollLeft, ScrollTop

Definiert die Breite des Scrollbereiches in Pixel. Mögliche Werte liegen im Bereich von -1 bis 32000.

Wird ein größerer oder kleinerer Wert angegeben, wird dieser entsprechend angepasst.

Wird die Eigenschaft auf den Wert -1 gesetzt, wird die Breite des Scrollbereiches auf die Breite der Scrollbox angepasst, wenn sich die Größe der Scrollbox ändert. In horizontaler Richtung kann dann nicht mehr gescrollt werden.

Die Eigenschaft kann über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

ScrollHeight

Höhe des Scrollbereiches

Typ int

Liste, Objekte,

WinPropGet(),

Siehe WinPropSet(),

ScrollWidth,

ScrollLeft, ScrollTop

Definiert die Höhe des Scrollbereiches in Pixel. Mögliche Werte liegen im Bereich von -1 bis 32000. Wird ein größerer oder kleinerer Wert angegeben, wird dieser entsprechend angepasst.

Wird die Eigenschaft auf den Wert -1 gesetzt, wird die Höhe des Scrollbereiches auf die Höhe der Scrollbox angepasst, wenn sich die Größe der Scrollbox ändert. In vertikaler Richtung kann dann nicht mehr gescrollt werden.

Die Eigenschaft kann über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

ShowPageOffset

Seitenabstand anzeigen Typ

logic

Liste, Objekte,

Siehe PrtPropGet(),

PrtPropSet()

Die Eigenschaft legt fest, ob der Seitenabstand von 1 cm zum linken oberen Rand zum PrintDocument dargestellt wird (true) oder nicht (false). Standardmäßig ist die Eigenschaft gesetzt.

StyleGroup

GroupTile-Anordnung

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Die Eigenschaft bestimmt, wie die untergeordneten GroupTile-Objekte angeordnet werden und hat nur eine Auswirkung, wenn sich drei GroupTile-Objekte im Darstellungsbereich des GroupSplit-Objekts befinden.

Die Eigenschaft verfügt über folgende Ausprägungen:

- WinStyleGroupTileBaseRB

Bei vertikaler Darstellung (siehe Eigenschaft Vertical) werden in der oberen Hälfte zwei GroupTile-Objekte dargestellt, in der unteren Hälfte ein GroupTile-Objekt. Bei horizontaler Darstellung werden in der linken Hälfte zwei GroupTile-Objekte dargestellt, und in der rechten Hälfte ein GroupTile-Objekt.

- WinStyleGroupTileBaseLT

Bei vertikaler Darstellung wird in der oberen Hälfte ein GroupTile-Objekt dargestellt und in der unteren zwei. Bei horizontaler Darstellung wird in der linken Hälfte ein GroupTile-Objekt dargestellt und in der rechten zwei.

Kontakt

Width

Breite

Typ int

Liste,

Objekte,

Siehe WinPropGet(),

WinPropSet(),

PrtPropGet(),

PrtPropSet()

- GroupTile-Objekt

Die Eigenschaft bestimmt die Breite des Objekts:

Ist der Wert ≥ 0 wird die prozentuale Breite zum gesamten Darstellungsbereich in 1/100 Prozent festgelegt.

Ist der Wert < 0 wird hierdurch die absolute Breite in Pixel angegeben. Das GroupTile-Objekt kann bei Angabe der absoluten Breite vom Benutzer nicht mehr in der Breite variiert werden.

Beispiele:

```
GroupTile->wpWidth # 2000; // 20.00 ProzentGroupTile->wpWidth # -120; // 120 Pixel
```

- PrtMetaPicture-Objekt

Beim PrtMetaPicture-Objekt kann über die Eigenschaft die Originalbreite des Bildes in logischen Einheiten definiert werden, sofern es sich um ein "Windows Meta File" handelt.

- PrtTblCell-Objekt

Beim PrtTblCell-Objekt kann über die Eigenschaft die Zellbreite in logischen Einheiten definiert werden.

- PrtFormList-Objekt

Beim PrintFormList-Objekt kann über die Eigenschaft die Breite der Tabelle in logischen Einheiten definiert werden.

Kontakt

Grouping (Gruppierungs-Eigenschaften)

<u>Eigenschaften</u>	
<u>sortiert nach</u>	
<u>Gruppen,</u>	
<u>Alphabetische</u>	
<u>Liste aller</u>	
Siehe <u>Eigenschaften</u> ,	
<u>Automatische</u>	
<u>Positionierung</u>	
<u>(Blog) Abstand</u>	
<u>und</u>	
<u>Ausrichtung</u>	
<u>(Blog)</u>	
<u>AlignGrouping</u>	Anordnung des Objekts bei automatischer Positionierung
<u>AlignGroupingBottomOrder</u>	Sichtbare Anordnung der gruppierten Objekte bei
<u>AlignGroupingLeftOrder</u>	automatischer Positionierung am unteren Rand Sichtbare Anordnung der gruppierten Objekte bei
<u>AlignGroupingRightOrder</u>	automatischer Positionierung am linken Rand Sichtbare Anordnung der gruppierten Objekte bei
<u>AlignGroupingTopOrder</u>	automatischer Positionierung am rechten Rand Sichtbare Anordnung der gruppierten Objekte bei
<u>AlignHeight</u>	automatischer Positionierung am oberen Rand Höhe des Objekts bei automatischer Anordnung
<u>AlignMarginBottom</u>	Unterer Rand des Objektes bei automatischer Anordnung
<u>AlignMarginLeft</u>	Linker Rand des Objektes bei automatischer Anordnung
<u>AlignMarginRight</u>	Rechter Rand des Objektes bei automatischer Anordnung
<u>AlignMarginTop</u>	Oberer Rand des Objektes bei automatischer Anordnung
<u>AlignWidth</u>	Breite des Objekts bei automatischer Anordnung
<u>Grouping</u>	Anordnung der untergeordneten Objekte

Kontakt

AlignGrouping  Anordnung des Objekts bei automatischer Positionierung Typ int

Liste, Objekte, AlignGroupingBottomOrder,
Siehe AlignGroupingLeftOrder, AlignGroupingRightOrder,
AlignGroupingTopOrder

Die Eigenschaft bestimmt, wie das Objekt angeordnet wird, wenn in dem übergeordneten Objekt die Eigenschaft Grouping auf einen Wert ungleich WinGroupingNone gesetzt ist. Folgende Werte können in dieser Eigenschaft angegeben werden:

- _WinAlignGroupingNone (1)

Das Objekt wird von der automatischen Anordnung nicht in seiner Position oder Größe verändert.

- _WinAlignGroupingLeft (2)

Das Objekt wird an der linken Seite des Eltern-Objekts ausgerichtet. Die Höhe des Objekts entspricht der Höhe des Eltern-Objekts, die Breite wird der Eigenschaft AlignWidth entnommen.

- _WinAlignGroupingTop (3)

Das Objekt wird an der oberen Seite des Eltern-Objekts ausgerichtet. Die Höhe des Objekts wird der Eigenschaft AlignHeight entnommen, die Breite entspricht der Breite des Eltern-Objekts.

- _WinAlignGroupingRight (4)

Das Objekt wird an der rechten Seite des Eltern-Objekts ausgerichtet. Die Höhe des Objekts entspricht der Höhe des Eltern-Objekts, die Breite wird der Eigenschaft AlignWidth entnommen.

- _WinAlignGroupingBottom (5)

Das Objekt wird an der unteren Seite des Eltern-Objekts ausgerichtet. Die Höhe des Objekts wird der Eigenschaft AlignHeight entnommen, die Breite entspricht der Breite des Eltern-Objekts.

- _WinAlignGroupingTiled (6)

Das Objekt wird abhängig von der Grouping-Eigenschaft des Eltern-Objekts horizontal (nebeneinander) oder vertikal (untereinander) ausgerichtet.

Bei dem Elternobjekt kann durch die Eigenschaften AlignGroupingBottomOrder, AlignGroupingLeftOrder, AlignGroupingRightOrder und AlignGroupingTopOrder zusätzlich die Darstellungsreihenfolge der Objekte angepasst werden.

Kontakt

AlignGroupingBottomOrder



Sichtbare Anordnung der gruppierten Objekte bei automatischer Positionierung Typ int

Liste, Objekte, AlignGroupingLeftOrder, AlignGroupingRightOrder,
AlignGroupingTopOrder
Standardmäßig werden Objekte mit der Option WinAlignGroupingBottom bei AlignGrouping in der Erstellungsreihenfolge (TabPos) am unteren Rand des übergeordneten Objektes von unten nach oben angeordnet. Folgende Werte können in dieser Eigenschaft angegeben werden:

- WinAlignGroupingOrderTopToBottom

Objekte werden in der Erstellungsreihenfolge am unteren Rand des übergeordneten Objektes von oben nach unten angeordnet.

- WinAlignGroupingOrderBottomToTop

Objekte werden in der Erstellungsreihenfolge am unteren Rand des übergeordneten Objektes von unten nach oben angeordnet.

Siehe

Kontakt

AlignGroupingLeftOrder



Sichtbare Anordnung der gruppierten Objekte bei automatischer Positionierung Typ int

Liste, Objekte, AlignGroupingRightOrder, AlignGroupingTopOrder,
AlignGroupingBottomOrder
Standardmäßig werden Objekte mit der Option WinAlignGroupingLeft bei AlignGrouping in der Erstellungsreihenfolge (TabPos) am linken Rand des übergeordneten Objektes von links nach rechts angeordnet. Folgende Werte können in dieser Eigenschaft angegeben werden:

- WinAlignGroupingOrderLeftToRight

Objekte werden in der Erstellungsreihenfolge am linken Rand des übergeordneten Objektes von links nach rechts angeordnet.

- WinAlignGroupingOrderLeftToRight

Objekte werden in der Erstellungsreihenfolge am linken Rand des übergeordneten Objektes von rechts nach links angeordnet.

Siehe

AlignGroupingRightOrder



Sichtbare Anordnung der gruppierten Objekte bei automatischer Positionierung Typ int

Liste, Objekte, AlignGroupingLeftOrder, AlignGroupingTopOrder,
AlignGroupingBottomOrder
Standardmäßig werden Objekte mit der Option WinAlignGroupingRight bei AlignGrouping in der Erstellungsreihenfolge (TabPos) am rechten Rand des übergeordneten Objektes von rechts nach links angeordnet. Folgende Werte können in dieser Eigenschaft angegeben werden:

- WinAlignGroupingOrderLeftToRight

Objekte werden in der Erstellungsreihenfolge am rechten Rand des übergeordneten Objektes von links nach rechts angeordnet.

- WinAlignGroupingOrderLeftToRight

Objekte werden in der Erstellungsreihenfolge am rechten Rand des übergeordneten Objektes von rechts nach links angeordnet.

Siehe



AlignGroupingTopOrder

Sichtbare Anordnung der gruppierten Objekte bei automatischer Positionierung Typ int

Liste, Objekte, AlignGroupingLeftOrder, AlignGroupingRightOrder,
AlignGroupingBottomOrder
Standardmäßig werden Objekte mit der Option WinAlignGroupingTop bei AlignGrouping in der Erstellungsreihenfolge (TabPos) am oberen Rand des übergeordneten Objektes von oben nach unten angeordnet. Folgende Werte können in dieser Eigenschaft angegeben werden:

- WinAlignGroupingOrderTopToBottom

Objekte werden in der Erstellungsreihenfolge am oberen Rand des übergeordneten Objektes von oben nach unten angeordnet.

- WinAlignGroupingOrderBottomToTop

Objekte werden in der Erstellungsreihenfolge am oberen Rand des übergeordneten Objektes von unten nach oben angeordnet.

Siehe

Kontakt

AlignHeight Höhe des Objekts bei automatischer Anordnung Typ int

Siehe Liste, Objekte

Diese Eigenschaft gibt die Höhe des Objekts an unter folgenden Bedingungen an:

1. Die Eigenschaft AlignGrouping ist auf WinAlignGroupingTop oder WinAlignGroupingBottom gesetzt und in dem übergeordneten Objekt ist eine automatische Ausrichtung der Objekte angegeben (siehe Grouping).
2. Die Eigenschaft AlignGrouping ist auf WinAlignGroupingTiled gesetzt und in dem übergeordneten Objekt ist die Eigenschaft Grouping auf WinGroupingCenterTileVert oder WinGroupingCenterTileHorz gesetzt.

Die Eigenschaft wird automatisch gesetzt, wenn die Eigenschaft Area, AreaTop oder AreaBottom geändert wird.

Kontakt

AlignMarginBottom



Unterer Rand des Objektes bei aktiviertem Grouping

Typ int

Siehe Liste, Objekte

Diese Eigenschaft gibt den unteren Rand des Objektes an, wenn in dem übergeordneten Objekt eine automatische Ausrichtung der Objekte angegeben ist (siehe Grouping).

Kontakt

AlignMarginLeft



Linker Rand des Objektes bei aktiviertem Grouping

Typ int

Siehe Liste, Objekte

Diese Eigenschaft gibt den linken Rand des Objektes an, wenn in dem übergeordneten Objekt eine automatische Ausrichtung der Objekte angegeben ist (siehe Grouping).

Kontakt

AlignMarginRight



Rechter Rand des Objektes bei aktiviertem Grouping

Typ **int**

Siehe **Liste, Objekte**

Diese Eigenschaft gibt den rechten Rand des Objektes an, wenn in dem übergeordneten Objekt eine automatische Ausrichtung der Objekte angegeben ist (siehe Grouping).

Kontakt

AlignMarginTop



Oberer Rand des Objektes bei aktiviertem Grouping

Typ int

Siehe Liste, Objekte

Diese Eigenschaft gibt den oberen Rand des Objektes an, wenn in dem übergeordneten Objekt eine automatische Ausrichtung der Objekte angegeben ist (siehe Grouping).

Kontakt

AlignWidth Breite des Objekts bei automatischer Anordnung Typ int

Siehe Liste, Objekte

Diese Eigenschaft gibt die Breite des Objekts an unter folgenden Bedingungen an:

1. Die Eigenschaft AlignGrouping ist auf WinAlignGroupingLeft oder WinAlignGroupingRight gesetzt und in dem übergeordneten Objekt ist eine automatische Ausrichtung der Objekte angegeben (siehe Grouping).
2. Die Eigenschaft AlignGrouping ist auf WinAlignGroupingTiled gesetzt und in dem übergeordneten Objekt ist die Eigenschaft Grouping auf WinGroupingCenterTileVert oder WinGroupingCenterTileHorz gesetzt.

Die Eigenschaft wird automatisch gesetzt, wenn die Eigenschaft Area, AreaLeft oder AreaRight geändert wird.

Kontakt

Grouping



Anordnung der untergeordneten Objekte

Typ int

Siehe Liste, Objekte, WinPropGet(),

WinPropSet()

Die Eigenschaft Grouping bestimmt die Anordnung untergeordneter Objekte. Bei den untergeordneten Objekten kann über die Eigenschaft AlignGrouping die Anordnung des Objekts weiter beeinflusst werden. In der Standardeinstellung (AlignGrouping = WinAlignGroupingTiled) nehmen die untergeordneten Objekte die gesamten Fläche des Objekts ein und werden entsprechend skaliert. Bei Änderungen der Größe des Objekts, werden alle untergeordneten Objekte neu ausgerichtet und skaliert. Folgende Konstanten können verwendet werden:

_WinGroupingNone Es findet keine automatische Anordnung statt.

_WinGroupingTileVert Die Objekte werden untereinander angeordnet.

_WinGroupingTileHorz Die Objekte werden nebeneinander angeordnet.

_WinGroupingCenterTileVert Die Objekte zentriert werden untereinander angeordnet.

_WinGroupingCenterTileHorz Die Objekte werden zentriert nebeneinander angeordnet.

Enthält ein Frame-Objekt ein FrameClient-Objekt, wirkt sich die Eigenschaft Grouping automatisch auf die untergeordneten Objekte des FrameClient-Objekts aus.

Durch Setzen der Eigenschaft AlignGrouping bei den untergeordneten Objekten, kann die Höhe (AlignHeight) bzw. die Breite (AlignWidth) der Objekte vorgegeben werden.

Ist die Ausprägung WinGroupingCenterTileVert oder WinGroupingCenterTileHorz gewählt, wird die Größe der untergeordneten Objekte von ihren Eigenschaften AlignWidth und AlignHeight bestimmt.



Die Auswirkungen werden erst zur Laufzeit sichtbar.

Kontakt

State (Status-Eigenschaften)

Eigenschaften

sortiert nach

Gruppen,

Siehe Alphabetische

Liste aller

Eigenschaften

AutoUpdate

Automatisches Neuzeichnen des Objektes

Changed

Änderung der Caption durch den Benutzer

ChangedChild

Änderung der Caption in einem untergeordneten Objekt

ChangedTrigger

Steuerungsflags der Eigenschaft Changed

CheckState

Vorgabewert bzw. Status

ClickedTrigger

Auslöser für das Ereignis EvtClicked

Design

Verschieben des Objektes zur Laufzeit

DesignFlags

Parameter des Design-Modus

Disabled

Aktivität des Objektes

Error

Expliziter Fehlerzustand des Eingabeobjektes

Fixed

Splitter trotz statischer Größenangabe anzeigen

Floatable

Lösen der Toolbar

MultiSelect

Mehrfachselektion aktivieren

OleDropMode

Drag & Drop-Berechtigungen

OleDropModeRelaxed OleDropMode während der Durchführung von Drag &

Drop-Operationen (EvtDragTerm, EvtDropEnter, EvtDropOver,
EvtDrop) setzen

Password

Eingabeobjekt als Passwort-Eingabeobjekt

PasswordChar

Zeichen, das bei der Passwort-Darstellung verwendet wird

ReadOnly

Eingabe nicht möglich

ScrollbarVisible

Anzeige der Scrollbar

Toggled

Zustand der Schaltfläche

Unicode

Objekt mit Unicode-Unterstützung

UseLogSize

Logische Auflösung berücksichtigen

Visible

Sichtbarkeit des Objektes

VisiblePrint

Sichtbarkeit des Objektes beim Drucken

Kontakt

AutoUpdate Automatisches Neuzeichnen des
Objektes Typ logic

Siehe Liste, Objekte

Wenn diese Eigenschaft auf true gesetzt ist, wird bei jeder Änderung einer Eigenschaft das Objekt neu gezeichnet.

Diese Eigenschaft sollte auf false gesetzt werden, wenn an einem Objekt mehrere Eigenschaften geändert werden, die ein Neuzeichnen erzwingen. Nachdem die Änderungen vorgenommen wurden, können durch das Setzen der Eigenschaft auf true die Änderungen auf einmal gezeichnet werden.

Beim Objekt View kann diese Eigenschaft nur dann auf false gesetzt werden, wenn die Eigenschaft beim GanttGraph-Objekt auf true gesetzt wurde. Die Eigenschaft mindestens eines der beiden Objekte muss auf true gesetzt werden.

Beim App-Objekt wirkt sich die Eigenschaft AutoUpdate auf die Eigenschaften TileTheme, TileSize und ColTheme von allen sichtbaren Fenstern der Applikation aus.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden. Die Eigenschaft wird ebenfalls durch den Befehl WinUpdate() gesetzt.

Kontakt

Changed Änderung der Caption durch den Benutzer Typ logic

Siehe Liste, Objekte, WinPropGet(),

Die Eigenschaft zeigt bei Eingabe-Objekten an, ob zwischen dem Betreten (EvtFocusInit) und dem Verlassen (EvtFocusTerm) die Caption des Eingabeobjektes durch den Benutzer verändert wurde.

Die Eigenschaft wird, bevor das Objekt den Fokus erhält, auf false gesetzt. Nach der ersten Änderung durch eine Eingabe über Tastatur, wird die Eigenschaft auf true gesetzt. Die Eigenschaft bleibt solange erhalten, bis der Eingabefokus das Objekt verlassen hat. Die Eigenschaft kann im Ereignis EvtFocusTerm ausgewertet werden.

Durch Setzen der Eigenschaft ChangedTrigger beim übergeordneten Frame- oder MDI-Frame-Objekt kann das Setzen der Eigenschaft Changed auch beim prozeduralen Ändern der Caption-Eigenschaft oder beim Übertragen der Feldpuffer in das Objekt aktiviert werden.

Kontakt

ChangedChild



Änderung der Caption in einem untergeordneten Objekt Typ logic

Siehe Liste, Objekte

Die Eigenschaft zeigt bei einem Fenster-Objekt an, ob von einem untergeordneten Objekt die Caption durch den Benutzer verändert wurde.

Durch Setzen der Eigenschaft ChangedTrigger kann das Setzen der Eigenschaft ChangedChild auch beim prozeduralen Ändern der Caption-Eigenschaft oder beim Übertragen der Feldpuffer in das Objekt aktiviert werden.

Die Eigenschaft kann zu jedem Zeitpunkt zurückgesetzt werden.

Kontakt

ChangedTrigger



Steuerungsflags der Eigenschaft Changed und ChangedChild Typ int

Siehe Liste, Objekte, WinPropGet(), WinPropSet()

Mit dieser Eigenschaft wird das Auftreten der Ereignisse EvtChanged / EvtChangedChild und das Verhalten der Eigenschaften Changed / ChangedChild gesteuert.

Ist keine der möglichen Optionen gesetzt, erfolgt das Ereignis EvtChanged sobald der Benutzer eine Eingabe in einem Eingabe-Objekt tätigt. Wird zum Beispiel ein vier-buchstabiges Wort in ein Edit-Objekt eingetragen, erfolgt das Ereignis vier mal. Die Eigenschaft Changed des Objektes wird, bevor das Objekt den Fokus erhält, auf false gesetzt. Nach der ersten Änderung durch eine Eingabe über Tastatur, wird die Eigenschaft auf true gesetzt. Die Eigenschaft bleibt solange erhalten, bis der Eingabefokus das Objekt verlassen hat. Die Eigenschaft kann im Ereignis EvtFocusTerm ausgewertet werden.

In der Eigenschaft ChangedTrigger können folgende Optionen gesetzt werden:

- WinChgTriggerCaption

Ist dieses Flag gesetzt, wird die Eigenschaft Changed auch dann gesetzt, wenn der Inhalt des Eingabeobjektes durch Änderung der entsprechenden Caption-Eigenschaft oder beim Checkbox-Objekt durch Änderung der CheckState-Eigenschaft verändert wurde. Ist dieses Flag gesetzt wird ebenfalls das Ereignis EvtChanged ausgelöst.

- WinChgTriggerFldBuf

Ist dieses Flag gesetzt, wird die Eigenschaft Changed auch dann gesetzt, wenn der Inhalt der Feldpuffer in das Eingabeobjekt übertragen wurde. Dies erfolgt nur, wenn in der Eigenschaft DbFieldName ein gültiges Datenbankfeld eingetragen ist. Ist dieses Flag gesetzt, wird ebenfalls das Ereignis EvtChanged ausgelöst.

Die Übertragung des Inhaltes der Feldpuffer in das Eingabeobjekt erfolgt, wenn das Objekt den Eingabefokus erhält, der Befehl WinUpdate() mit dem Parameter WinUpdFld2Obj ausgeführt wird oder über das Objekt RecNavigator ein neuer Datensatz gelesen wird.

- WinChgTriggerNoEvent

Ist dieses Flag gesetzt, wird das Ereignis EvtChanged nicht ausgelöst.

- WinChgTriggerChild

Ist dieses Flag gesetzt, wird nach dem Ereignis EvtChanged des Eingabe-Objekts das Ereignis EvtChangedChild des übergeordneten Fenster-Objekts ausgelöst. Dabei wird ebenfalls die Eigenschaft ChangedChild bei dem Fenster-Objekt gesetzt. Welche Änderungen im Objekt das Ereignis auslösen, wird durch die anderen Ausprägungen definiert.

Die Optionen können wie folgt überprüft und gesetzt werden:

Kontakt

```
// Option auswerten:if (tFrame->wpChangedTrigger & _WinChgTriggerCaption > 0) { ... } // Flag s
```

Kontakt

CheckState

Vorgabewert bzw. Status

Typ int

Liste, Objekte,

Siehe WinPropGet(),
WinPropSet()

Über diese Eigenschaft kann der Vorgabewert einer Checkbox bzw. der Zustand eines Radiobuttons ermittelt und gesetzt werden.

WinStateChkChecked angehakt bzw. ausgewählt

WinStateChkUnchecked nicht angehakt bzw. nicht ausgewählt

Bei dem Setzen des Wertes WinStateChkChecked bei einem Radiobutton werden Radiobuttons mit dem gleichen Elternobjekt automatisch auf WinStateChkUnchecked gesetzt.

Wird die Eigenschaft eines Radiobuttons in dem Ereignis EvtFocusInit oder EvtFocusTerm auf WinStateChkChecked gesetzt, erhält dieses Objekt beim nächsten

Objektwechsel mit der -Taste den Fokus. Dies kann mit der Option

WinFlagNoFocusSet unterbunden werden.

Beispiel:

```
// Setzen der Radiobuttons $RadioButton1->wpCheckstate # _WinStateChkChecked; $RadioButton2->wpChec
```

Kontakt

ClickedTrigger

Auslöser für das Ereignis EvtClicked

Typ int
Siehe Liste, Objekte, WinPropGet(),

Die Eigenschaft bestimmt unter welchen Umständen das EvtClicked Ereignis ausgelöst wird.

Standardmäßig wird das EvtClicked-Ereignis auch dann ausgelöst, wenn der Button bereits markiert ist und der Benutzer beispielsweise auf den Button klickt. Wird die Konstante WinClkTriggerChanged gesetzt, wird EvtClicked nur dann ausgelöst, wenn der Radiobutton aktiviert wurde und zuvor nicht markiert war (CheckState = WinStateChkUnchecked).

Kontakt

Design

Verschieben des Objekts zur Laufzeit

Typ [logic](#)

Siehe [Liste, Objekte](#)

Wird diese Eigenschaft bei einem verschiebbaren Fenster-Objekt gesetzt, erhält das Objekt zur Laufzeit einen Selektions-Rahmen und kann vom Benutzer verschoben und in der Größe geändert werden (sofern das Objekt dies zulässt). Zum Verändern werden auf dem Selektionsrahmen an den Ecken und in der Mitte der Linien Vierecke angezeigt, an denen das Objekt verschoben (oben/mitte) und in der Größe verändert werden kann. Wird die Eigenschaft zurückgesetzt, verschwindet der Selektions-Rahmen wieder.

Ist bei einem Objekt die Eigenschaft [AutoSize](#) gesetzt, kann nur die Position des Objekts geändert werden. Objekte, die deaktiviert sind (Eigenschaft [Disabled](#)) können nicht verändert werden.

Nach der Änderung des Objekts wird das Ereignis [EvtChangedDesign](#) aufgerufen.

Die Speicherung der veränderten Position der Objekte kann mit der Anweisung [WinSave\(\)](#) erfolgen, wenn der Dialog zuvor mit [WinOpen\(..., WinOpenLock\)](#) geöffnet wurde.

DesignFlags

Parameter des Design-Modus

Typ `int`

Siehe [Liste](#), [Objekte](#)

In dieser Eigenschaft werden die Parameter des Design-Modus (siehe [Design](#)) angegeben. Folgende Parameter werden unterstützt:

- `_WinDesignDefault (0)`

Dies ist das Standardverhalten im Design-Modus. Es wird ein Selektionsrahmen gezeichnet mit den entsprechenden Anker zum Vergrößern und Verschieben des Objekts gezeichnet.

- `_WinDesignDragInterior (1)`

Der Benutzer kann das Objekt auch dann verschieben, wenn er in den Innenbereich des Selektionsrahmens klickt. Am oberen Rand wird kein Bewegungs-Anker gezeichnet.

Die Eigenschaft kann nur zur Laufzeit verändert werden.

Disabled

Aktivität des Objektes

Typ [logic](#)
[Liste, Objekte](#),

Siehe [ReadOnly](#),
[WinPropGet\(\)](#),
[WinPropSet\(\)](#)

Bei true ist das Objekt inaktiv. In diesem Fall ist es ist nicht mehr selektierbar und wird grau dargestellt.



Die Eigenschaft darf nicht für das Objekt auf true gesetzt werden, das zu diesem Zeitpunkt den Fokus besitzt.

EditorBraceHighlight

Zugehörige Klammer markieren

Typ [logic](#)

Siehe [Liste, Objekte](#)

Wenn diese Eigenschaft auf `true` gesetzt ist, wird von einer markierten Klammer die zugehörige öffnende / schließende Klammer markiert. Standardmäßig ist diese Eigenschaft gesetzt.



Diese Eigenschaft kann frühestens im [EvtCreated](#) des Elternfensters gesetzt werden.

Indizes:

Das Setzen der Eigenschaft ist prozedural über Indizes pro View des [CodeEdit](#)-Objektes möglich. Wird kein Index, oder der Index 0 angegeben, wird die Eigenschaft in allen Views verändert. Beim Ermitteln wird das erste View abgefragt. Mit den Indizes 1 bis 4 können die einzelnen Views angesprochen werden.

Beispiel:

```
// Eigenschaft bei allen Views des Objektes setzen$CodeEdit->wpEditorBraceHighlight # true;// Eig
```

Kontakt

EditorFold

Folding im CodeEdit-Objekt aktivieren

Typ logic

Siehe Liste, Objekte

Wenn diese Eigenschaft auf true gesetzt ist, können im CodeEdit-Objekt Prozeduren an Klammern zu- und aufgeklappt werden. Standardmäßig ist diese Eigenschaft gesetzt.

Ist die Eigenschaft aktiviert, kann das Folding mit folgenden Tastenkombinationen gesteuert werden:

+ +	Aktuellen Block zuklappen
+ +	Aktuellen Block aufklappen
+ +	Alle Blöcke zuklappen
+ +	Alle Blöcke aufklappen

Diese Eigenschaft kann frühestens im EvtCreated des Elternfensters gesetzt werden.

Indizes:

Das Setzen der Eigenschaft ist prozedural über Indizes pro View des CodeEdit-Objektes möglich. Wird kein Index, oder der Index 0 angegeben, wird die Eigenschaft in allen Views verändert. Beim Ermitteln wird das erste View abgefragt. Mit den Indizes 1 bis 4 können die einzelnen Views angesprochen werden.

Beispiel:

```
// Eigenschaft bei allen Views des Objektes setzen$CodeEdit->wpEditorFold # true;// Eigenschaft v
```

Kontakt

EditorReadOnly

Nur-Lese-Modus im CodeEdit-Objekt

Typ logic

Siehe Liste, Objekte

Ist der Wert der Eigenschaft auf true gesetzt, kann der Benutzer keine Eingaben in diesem CodeEdit-Objekt t  igen. Standardm  ig ist diese Eigenschaft auf false gesetzt.



Diese Eigenschaft kann fr  hestens im EvtCreated des Elternfensters gesetzt werden.

Indizes:

Das Setzen der Eigenschaft ist prozedural   ber Indizes pro View des CodeEdit-Objektes m  glich. Wird kein Index, oder der Index 0 angegeben, wird die Eigenschaft in allen Views ver  ndert. Beim Ermitteln wird das erste View abgefragt. Mit den Indizes 1 bis 4 k  nnen die einzelnen Views angesprochen werden.



Wird diese Eigenschaft auf einem einzelnen View gesetzt, wird die ebenfalls f  r alle Views   bernommen, bei denen die Eigenschaften FileName und EditorTextType auf den gleichen Wert gesetzt ist.

Beispiel:

```
// Eigenschaft bei allen Views des Objektes setzen$CodeEdit->wpEditorReadOnly # true;// Eigenschaft
```

Kontakt

Error Expliziter Fehlerzustand des Eingabeobjektes

Typ logic

Siehe [Liste](#), [Objekte](#), [WinPropGet\(\)](#),

Wird diese Eigenschaft gesetzt, wird das Eingabeobjekt als fehlerhaft angezeigt. Die Eigenschaft wird nicht ausgewertet, wenn ein Wert eines Eingabeobjektes außerhalb des gültigen Bereichs liegt (z. B. kleiner als MinInt oder größer als MaxInt bei einem IntEdit). Standardmäßig ist diese Eigenschaft auf false gesetzt.

Beispiel:

```
// Fehlerzustand setzen$Edit->wpError # true;
```

Kontakt

Fixed Splitter trotz statischer Größenangabe anzeigen

Typ logic

Siehe Liste, Objekte, WinPropGet(), WinPropSet() GroupTile-Objekte können über einen Splitter vom Anwender in der Größe verändert werden. Hierzu muss die Eigenschaft Width bzw. Height jedoch auf einen positiven Wert gesetzt werden (prozentuale Angabe). Bei Veränderung der Größe des übergeordneten GroupSplit-Objektes verändert sich dann die Größe des GroupTile-Objektes entsprechend.

Ist dies nicht erwünscht, kann die Größe des GroupTile-Objektes auf einen negativen Wert (Pixelangabe) gesetzt werden. Das GroupTile behält dann seine Größe auch bei, wenn das übergeordnete GroupSplit-Objekt in der Größe verändert wird. Jedoch wird in diesem Fall kein Splitter dargestellt. Somit kann das GroupTile-Objekt nicht vom Anwender in der Größe verändert werden.

Mit dieser Eigenschaft kann unter den folgenden Bedingungen auch in diesem Fall ein Splitter angezeigt werden:

- Eines der jeweils zwei aneinander liegenden Objekte hat eine prozentuale Größenangabe. Das andere hat eine Pixelangabe.
- Bei dem GroupTile mit Pixelangabe ist die Eigenschaft Fixed gesetzt.

Haben die zwei aneinander liegenden Objekte eine prozentuale Angabe, wird die Eigenschaft ignoriert und der Splitter dennoch angezeigt.



Beim Verschieben eines GroupTile auf den Platz eines anderen (mit der Maus über den Caption) behält das GroupTile den Wert für Fixed bei. Dadurch kann sich die Sichtbarkeit der Splitter im GroupSplit-Objekt ändern.

Floatable

Lösen der Toolbar

Typ logic
Liste, Objekte,
Siehe WinPropGet(),
WinPropSet()

Wird die Eigenschaft auf true gesetzt, kann die Toolbar während der Laufzeit aus dem ToolbarDock-Objekt herausgezogen und in ein anderes ToolbarDock-Objekt hineingezogen werden.

Es ist auch möglich die Toolbar freischwebend auf dem Bildschirm zu platzieren. Wird eine freischwebende Toolbar über die Schließen-Schaltfläche oder + abgeschaltet, kann die abgeschaltete Toolbar über die Eigenschaft Visible wieder dargestellt werden.

Zum Lösen der Toolbar gibt es zwei Möglichkeiten:

- Doppelklick auf den Toolbar-Grip
- Ziehen mit der Maus am Toolbar-Grip oder an einem freien Platz (zum Beispiel Separator) in der Toolbar

Kontakt

MultiSelect



Mehrfachselektion aktivieren

Typ logic

Liste, Objekte, SelData,

Siehe WinPropGet(),

WinPropSet()

Ist diese Eigenschaft auf true gesetzt, ist bei dem Objekt eine Mehrfachselektion

möglich. Mehrere Objekte können dann durch gedrückt halten der -Taste mit der Maus selektiert werden. Für jedes der selektierten Objekte wird ein "Select"-Ereignis (EvtLstSelect bzw. EvtNodeSelect) ausgelöst.

Die selektierten Objekte können über ein SelectionData-Objekt ermittelt werden. Der Deskriptor dieses Objekts befindet sich in der Eigenschaft SelData.

Kontakt

OleDropMode



Drag & Drop-Berechtigungen

Typ int

Siehe Liste, Objekte,

OleDropModeRelaxed

Diese Eigenschaft bestimmt welche OLE-Fähigkeiten des Objekts unterstützt werden.

Folgende Berechtigungs-Modi gibt es:

WinOleNone

Der Inhalt des Objekts kann nicht kopiert und der Inhalt eines anderen Objekts kann nicht eingefügt werden.

WinOleSource

Der Inhalt des Objekts kann kopiert, der Inhalt eines anderen Objekts kann aber nicht eingefügt werden.

WinOleTarget

Der Inhalt eines anderen Objekts kann in das Objekt eingefügt, aber nicht aus dem Objekt kopiert werden.

WinOleBoth

Der Inhalt kann aus dem Objekt kopiert und in das Objekt eingefügt werden.

WinOleDynamic Für das Objekt werden die dynamische Drag & Drop-Funktionen aktiviert.



Die Eigenschaft OleDropMode kann während Drag & Drop-Operationen nur (EvtDragTerm, EvtDropEnter, EvtDropOver, EvtDrop) verändert werden, wenn zuvor die Eigenschaft OleDropModeRelaxed des App-Objektes gesetzt wurde.

Kontakt

OleDropModeRelaxed



Eigenschaft OleDropMode während Drag & Drop-Operationen änderbar Typ logic

Siehe Liste, Objekte, OleDropMode, WinPropGet()

Die Eigenschaft OleDropMode von Oberflächen-Objekten kann während der Durchführung einer Drag & Drop-Operation nicht gesetzt werden (EvtDragTerm, EvtDropEnter, EvtDropOver, EvtDrop). Unter Umständen kann dies jedoch notwendig sein. Wird die Eigenschaft OleDropModeRelaxed auf true gesetzt, dann kann auch während der Durchführung von Drag & Drop-Operationen die Eigenschaft OleDropMode verändert werden.



Die Eigenschaft OleDropModeRelaxed kann nicht während der Durchführung einer Drag & Drop-Operation gesetzt werden.

Kontakt

Password Eingabeobjekt als Passwort-
Eingabeobjekt Typ logic

Siehe Liste, Objekte, WinPropGet(),

Wenn diese Eigenschaft auf true gesetzt wird, erscheinen bei der Eingabe nicht mehr die eingegebenen Zeichen, sondern nur das Zeichen, das in der Eigenschaft PasswordChar angegeben wurde.

Kontakt

PasswordChar



Zeichen, das bei der Passwort-Darstellung verwendet wird Typ alpha

Siehe Liste, Objekte, WinPropGet(), WinPropSet()

Dieses Zeichen wird zur Darstellung des Inhaltes verwendet, wenn die Eigenschaft Password auf true gesetzt ist.

Um das betriebssystemabhängige Zeichen zu verwenden, muss '***' angegeben werden. Weiterhin muss die Eigenschaft StyleTheme des Application-Objektes auf WinStyleThemeSystem gesetzt werden. Das betriebssystemabhängige Zeichen wird nicht von allen Schriftarten unterstützt. In der ab Windows XP mitgelieferten Schriftart Tahoma ist das Zeichen enthalten. Weiterhin kann das Zeichen erst ab Windows XP verwendet werden.

ReadOnly

Eingabe nicht möglich

Typ [logic](#)

Siehe [Liste](#), [Objekte](#)

Ist der Wert der Eigenschaft auf true gesetzt, kann der Benutzer keine Eingaben in diesem Eingabeobjekt t tigen.

Wird diese Eigenschaft bei einem DataListPopup-Objekt gesetzt, kann die selektierte Zeile durch Benutzerinteraktionen nicht mehr ver ndert werden. Dadurch kann in Verbindung mit der **ReadOnly-Eigenschaft des Eingabe-Objekts verhindert werden, dass der Benutzer den Inhalt des Eingabe-Objekts ver ndert, indem er in der Popup-Liste eine andere Zeile selektiert.**

Soll trotzdem das Kontextmen  aufgerufen werden k nnen, muss in der Eigenschaft Flags des App-Objekts die Auspr gung WinAppEditTextROContextMenu gesetzt werden. Ist diese Auspr gung gesetzt, kann zudem der Inhalt des Eingabe- bzw. des RtfEdit-Objekts mit der Tastatur oder der Maus selektiert werden.

Beim RecView-Objekt kann die Eigenschaft sowohl 脿ber den RecView-Editor als auch im Ereignis EvtLstGroupInit ver ndert werden. Ist diese Eigenschaft gesetzt, kann das Item nicht editiert werden.

Beim Canvas-Objekt steuert die Eigenschaft, ob eine Interaktion des Anwenders mit dem Canvas-Objekt zul ssig ist. Standardm  ig ist dies der Fall. Sofern die Eigenschaft auf true gesetzt wird, ist es nicht mehr m glich Graphikobjekte (CanvasGraphic) auszuw hlen oder die vorhandene Auswahl zu ver ndern.

ScrollBarVisible

Anzeige der Scrollbar

Typ logic

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Über diese Eigenschaft wird festgelegt, ob das Objekt einen horizontalen und vertikalen Rollbalken erhält (true).

Je nach Objekt ist folgendes zu beachten:

- Picture

Bei true erhält das Objekt, falls die Grafik größer ist als der dafür reservierte Raum, einen horizontalen und vertikalen Rollbalken. Andernfalls werden keine dargestellt.

- TextEdit

Bei true werden immer Rollbalken angezeigt. Wenn kein Text vorhanden ist oder der Text vollständig in dem Objekt dargestellt werden kann, werden die Rollbalken automatisch deaktiviert.

Kontakt

Toggled



Zustand der Schaltfläche

Typ logic

Siehe Liste, Objekte

Ist diese Eigenschaft gesetzt, wird die Schaltfläche gedrückt dargestellt.

Unicode

Objekt mit Unicode-Unterstützung

Typ [logic](#)

Siehe [Liste, Objekte](#)

Über diese Eigenschaft kann ermittelt werden, ob ein Objekt über **Unicode-Unterstützung** verfügt ([true](#)), oder nicht ([false](#)). Die Eigenschaft kann nur gelesen werden.

UseLogSize

Logische Auflösung berücksichtigen

Typ [logic](#)

Siehe [Liste, Objekte](#)

Mit dieser Eigenschaft wird angegeben, ob die logische Auflösung eines Bildes berücksichtigt werden soll ([true](#)) oder nicht ([false](#)). Standardmäßig wird die logische Auflösung nicht berücksichtigt.

Normalerweise spielt die Auflösung für die Anzeige keine Rolle, da die horizontale und vertikale Auflösung gleich ist. Unterscheiden sich diese Auflösungen voneinander, wird das Bild nicht mit dem erwarteten Seitenverhältnis angezeigt.

Beispiel:

Ein Bild hat eine Breite von 1728 Pixel mit einer Auflösung von 200 pixel/inch. Die Höhe ist 1113 Pixel mit 100 pixel/inch. Wird die logische Auflösung nicht berücksichtigt (UseLogSize = [false](#)), ist das Bild breiter als hoch (1728 x 1113 pixel, Querformat). Wird die logische Auflösung berücksichtigt (UseLogSize = [true](#)), ist das Bild höher als breit ($1728 / 200 = 8,64$ inch = 21,95 cm breit und $1113 / 100 = 11,13$ inch = 28,27 cm hoch, Hochformat).

Die Informationen der horizontalen und vertikalen logischen Auflösung werden dem Bildformat entnommen.

Visible

Sichtbarkeit des Objektes

Typ logic

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Wenn diese Eigenschaft den Wert false enthält, ist das Objekt nicht sichtbar. Bei dem Objekt Application bezieht sich die Eigenschaft auf den CONZEPT 16-Client und den von ihm erzeugten Splash-Screen beim Starten.



Die Eigenschaft darf nicht für das Objekt auf false gesetzt werden, das zu diesem Zeitpunkt den Fokus besitzt.

Kontakt

VisiblePrint



Sichtbarkeit des Objektes beim Drucken

Typ int
Siehe Liste, Objekte, PrtPropGet(),
 PrtPropSet()

Diese Eigenschaft bestimmt die Sichtbarkeit eines Objektes bei der Druckaufbereitung:

- PrtVisiblePrintPreview
das Objekt ist im Preview-Dialog sichtbar
- PrtVisiblePrintJob
das Objekt ist beim Druck sichtbar
- PrtVisiblePrintLine
Das Objekt wird gedruckt.

Die Optionen können kombiniert werden.

Die Optionen sind z. B. sinnvoll, wenn es einen Seitenkopf gibt, der im Preview sichtbar sein soll, jedoch nicht gedruckt werden soll, da auf dem verwendeten Papier bereits ein entsprechender Seitenkopf vorgedruckt ist.

Soll der Inhalt der Eigenschaft zurückgesetzt werden, kann der Wert PrtVisiblePrintNone verwendet werden.



Die Sichtbarkeit, die durch die Konstanten PrtVisiblePrintPreview und PrtVisiblePrintJob festgelegt wird, hat keinen Einfluss auf die Sichtbarkeitskriterien eines PrtLine-Objektes.

Beispiel:

```
// Das Objekt ist nicht im Preview-Dialog sichtbarPrtRtf->ppVisiblePrint # tPrtRtf->ppVisiblePri
```

Kontakt

Color (Farb-Eigenschaften)

Eigenschaften

sortiert nach

Gruppen,

Siehe Alphabetische

Liste aller

Eigenschaften

BorderColFg

Rahmenfarbe einer Column, SubColumn, eines Items, SubItems _____
oder eines CtxDocEdit-Objektes

ClmColBkg

Hintergrundfarbe einer Spalte

ClmColFg

Vordergrundfarbe einer Spalte

ClmColFocusBkg

Hintergrundfarbe des Fokus

ClmColFocusFg

Vordergrundfarbe des Fokus

ClmColFocusOffBkg Hintergrundfarbe des Fokus, wenn das Objekt nicht fokussiert ist

ClmColFocusOffFg Vordergrundfarbe des Fokus, wenn das Objekt nicht fokussiert

ist

Automatische Farbe

ColBkg

Hintergrundfarbe

ColBkgApp

Hintergrundfarbe der Applikation

ColBkgBase

Basis Füllfarbe setzen

ColBkgOffPrint

Farbe des nicht druckbaren Bereiches

ColBkgPage

Farbe der Seite

ColBkgTabArea

Hintergrundfarbe der Fläche hinter den Registerreitern

ColBkgTabSel

Hintergrundfarbe des ausgewählten Registerreiters

ColBoundary

Farbe des Begrenzungsrechtecks

ColCellActive

Farbe für die Zellmarkierungen

ColCellActiveHorz

Farbe für die Zellmarkierungen

ColDateFg

Vordergrundfarbe

ColDateMonth

Hintergrundfarbe des Monats

ColDateTitleBkg

Hintergrundfarbe des Monatstitels

ColDateTitleFg

Vordergrundfarbe des Monatstitels

ColDateTrailing

Trailing-Farbe

ColDesign

Farbe im Design-Modus

ColDisabledBkg

Hintergrundfarbe des selektierten Datensatzes

ColDisabledFg

Vordergrundfarbe des selektierten Datensatzes

ColErrRangeBkg

Hintergrundfarbe bei fehlerhafter Eingabe

ColErrRangeFg

Vordergrundfarbe bei fehlerhafter Eingabe

ColFg

Vordergrundfarbe

ColFgTabSel

Vordergrundfarbe des selektierten Registerreiters

ColFillEven

Farbe der geraden Markierung

ColFillOdd

Farbe der ungeraden Markierung

ColFocusBkg

Hintergrundfarbe, wenn das Objekt fokussiert ist

ColFocusFg

Vordergrundfarbe , wenn das Objekt fokussiert ist

Kontakt

<u>ColFocusOffBkg</u>	Hintergrundfarbe des Fokus, wenn das Objekt nicht fokussiert ist
<u>ColFocusOffFg</u>	Vordergrundfarbe des Fokus, wenn das Objekt nicht fokussiert ist
<u>ColGrid</u>	Farbe des Gitters
<u>ColHover</u>	Farbe, wenn der Benutzer mit der Maus auf dem Objekt steht
<u>ColIvlArrows</u>	Farbe der Intervall-Pfeile
<u>ColIvlBorder</u>	Rahmenfarbe für Intervalle
<u>ColLink</u>	Vordergrundfarbe
<u>ColorMode</u>	ColorMode-Modus
<u>ColParent</u>	Elternfarbe setzen
<u>ColSeparator</u>	Farbe des Trenners zwischen fixierten und nicht fixierten Spalten
<u>ColVisited</u>	Farbe, nachdem der Hyperlink ausgeführt wurde
<u>GroupColBkg</u>	Hintergrundfarbe einer Gruppe
<u>GroupColFg</u>	Vordergrundfarbe einer Gruppe
<u>HdrColFg</u>	Vordergrundfarbe des Spaltenkopfes
<u>HdrShadeCol1</u>	Füllfarbe des Spaltenkopfes
<u>HdrShadeCol2</u>	Farbe des Spaltenkopfes für Farbverlauf
<u>HdrShadeMode</u>	Modus des Farbverlaufs des Spaltenkopfes
<u>ScalaColorBkg</u>	Hintergrundfarbe der Skala
<u>ScalaColorBkgBase</u>	Hintergrundfarbe der Skala (ungerade Zeilen/Spalten)
<u>ScalaColorEdge</u>	Farbe der Trennung zwischen Überschrift und Anzeigebereich
<u>ScalaColorEnd</u>	Farbe des ungenutzten Skala-Bereichs
<u>ScalaColorText</u>	Farbe des Textes in der Skala
<u>ShadeCol1</u>	Erste Farbe des Farbverlaufs
<u>ShadeCol2</u>	Letzte Farbe des Farbverlaufs
<u>ShadeMode</u>	Modus des Farbverlaufs im sichtbaren Bereich des Views
<u>TitleColorBkg</u>	Hintergrundfarbe des Achsentitels
<u>TitleColorFg</u>	Vordergrundfarbe des Achsentitels

Kontakt



BorderColFg

Rahmenfarbe einer Column, SubColumn, eines Items, SubItems, RecView- oder eines CtxDocEdit-Objektes

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Die Konstante setzt die Rahmenfarbe des jeweiligen Objektes.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Wird diese Eigenschaft im Ereignis EvtLstGroupInit gesetzt, wirkt sich das nur auf das aktuelle Item oder SubItem aus. Die Eigenschaft wird nicht für die gesamte Gruppe geändert.

Wird die Eigenschaft beim RecView-Objekt gesetzt, erhalten alle Spalten und Gruppen diese Rahmenfarbe.

Wird die Eigenschaft bei einem CtxDocEdit-Objekt auf WinColUndefined gesetzt, wird die Standard-Rahmenfarbe des CtxDocEdit-Objektes verwendet. Andernfalls wird ein Rahmen in der jeweils angegebenen Farbe gezeichnet.

Beispiel:

```
aGroup->wpBorderColFg # _WinColBlue;$CtxDocEdit->wpBorderColFg # _WinColUndefined;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

ClmColBkg



Hintergrundfarbe einer Spalte

Typ [int](#)

Siehe [Liste](#), [Objekte](#), [Beispiel](#)

In dieser Eigenschaft können [Farbkonstanten](#) gesetzt werden.

Die Konstante setzt die Hintergrundfarbe der Spalte.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe [Beispiel](#)).

Wird diese Eigenschaft im Ereignis [EvtLstDataInit](#) gesetzt, wirkt sich das nur auf die aktuelle Zelle aus. Die Eigenschaft wird nicht für die gesamte Spalte geändert.

Beispiel:

```
aEvt:Obj->wpClmColBkg # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle [WinPropGet\(\)](#) und [WinPropSet\(\)](#) gelesen bzw. geändert werden.



Im [Modern Theme Style](#) wirken sich Änderungen der Eigenschaft im [EvtLstDataInit](#) aus, sofern sich der gesetzte Wert von dem gespeicherten Wert in der Spalte unterscheidet. Wird der Wert in [EvtLstDataInit](#) auf den gespeicherten Wert gesetzt oder nicht verändert, wird die jeweilige Eigenschaft von dem Theme-Objekt verwendet.

ClmColFg

Vordergrundfarbe einer Spalte

Typ [int](#)

Siehe [Liste](#), [Objekte](#), [Beispiel](#)

In dieser Eigenschaft können [Farbkonstanten](#) gesetzt werden. Die Konstante setzt die Vordergrundfarbe der Spalte.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe [Beispiel](#)).

Wird diese Eigenschaft im Ereignis [EvtLstDataInit](#) gesetzt, wirkt sich das nur auf die aktuelle Zelle aus. Die Eigenschaft wird nicht für die gesamte Spalte geändert.

Beispiel:

```
aEvt:Obj->wpClmColFg # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle [WinPropGet\(\)](#) und [WinPropSet\(\)](#) gelesen bzw. geändert werden.



Im [Modern Theme Style](#) wirken sich Änderungen der Eigenschaft im [EvtLstDataInit](#) aus, sofern sich der gesetzte Wert von dem gespeicherten Wert in der Spalte unterscheidet. Wird der Wert in [EvtLstDataInit](#) auf den gespeicherten Wert gesetzt oder nicht verändert, wird die jeweilige Eigenschaft von dem Theme-Objekt verwendet.

ClmColFocusBkg

Hintergrundfarbe des Fokus

Typ int

Siehe Liste, Objekte, Beispiel

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Die Konstante setzt die Hintergrundfarbe der Spalte.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Wird diese Eigenschaft im Ereignis EvtLstDataInit gesetzt, wirkt sich das nur auf die aktuelle Zelle aus. Die Eigenschaft wird nicht für die gesamte Spalte geändert.

Beispiel:

```
aEvt:Obj->wpClmColFocusBkg # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.



Im Modern Theme Style wirken sich Änderungen der Eigenschaft im EvtLstDataInit aus, sofern sich der gesetzte Wert von dem gespeicherten Wert in der Spalte unterscheidet. Wird der Wert in EvtLstDataInit auf den gespeicherten Wert gesetzt oder nicht verändert, wird die jeweilige Eigenschaft von dem Theme-Objekt verwendet.

ClmColFocusFg

Vordergrundfarbe des Fokus

Typ [int](#)

Siehe [Liste](#), [Objekte](#), [Beispiel](#)

In dieser Eigenschaft können [Farbkonstanten](#) gesetzt werden. Die Konstante setzt die Vordergrundfarbe Spalte.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe [Beispiel](#)).

Wird diese Eigenschaft im Ereignis [EvtLstDataInit](#) gesetzt, wirkt sich das nur auf die aktuelle Zelle aus. Die Eigenschaft wird nicht für die gesamte Spalte geändert.

Beispiel:

```
aEvt:Obj->wpClmColFocusFg # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle [WinPropGet\(\)](#) und [WinPropSet\(\)](#) gelesen bzw. geändert werden.



Im [Modern Theme Style](#) wirken sich Änderungen der Eigenschaft im [EvtLstDataInit](#) aus, sofern sich der gesetzte Wert von dem gespeicherten Wert in der Spalte unterscheidet. Wird der Wert in [EvtLstDataInit](#) auf den gespeicherten Wert gesetzt oder nicht verändert, wird die jeweilige Eigenschaft von dem Theme-Objekt verwendet.

Kontakt

ClmColFocusOffBkg



Hintergrundfarbe des Fokus, wenn das Objekt nicht fokussiert ist Typ int

Siehe Liste, Objekte, Beispiel

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Die Konstante setzt die Hintergrundfarbe des Fokus innerhalb des Objektes, wenn der Eingabefokus nicht in diesem Objekt steht.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Wird diese Eigenschaft im Ereignis EvtLstDataInit gesetzt, wirkt sich das nur auf die aktuelle Zelle aus. Die Eigenschaft wird nicht für die gesamte Spalte geändert.

Beispiel:

```
aEvt:Obj->wpClmColFocusOffBkg # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.



Im Modern Theme Style wirken sich Änderungen der Eigenschaft im EvtLstDataInit aus, sofern sich der gesetzte Wert von dem gespeicherten Wert in der Spalte unterscheidet. Wird der Wert in EvtLstDataInit auf den gespeicherten Wert gesetzt oder nicht verändert, wird die jeweilige Eigenschaft von dem Theme-Objekt verwendet.

Kontakt

ClmColFocusOffFg



Vordergrundfarbe des Fokus, wenn das Objekt nicht fokussiert ist Typ int

Siehe Liste, Objekte, Beispiel

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Die Konstante setzt die Vordergrundfarbe des Fokus des Objekts, wenn der Eingabefokus nicht in diesem Objekt steht.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Wird diese Eigenschaft im Ereignis EvtLstDataInit gesetzt, wirkt sich das nur auf die aktuelle Zelle aus. Die Eigenschaft wird nicht für die gesamte Spalte geändert.

Beispiel:

```
aEvt:Obj->wpClmColFocusOffFg # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.



Im Modern Theme Style wirken sich Änderungen der Eigenschaft im EvtLstDataInit aus, sofern sich der gesetzte Wert von dem gespeicherten Wert in der Spalte unterscheidet. Wird der Wert in EvtLstDataInit auf den gespeicherten Wert gesetzt oder nicht verändert, wird die jeweilige Eigenschaft von dem Theme-Objekt verwendet.

ColAuto

Automatische Farbe

Typ [int](#)

[Liste, Objekte,](#)

Siehe [CaptionColAuto](#),

[WinPropGet\(\)](#),

[WinPropSet\(\)](#)

In dieser Eigenschaft können **Farbkonstanten** gesetzt werden. Die Konstante setzt die automatische Farbe für dieses Objekt.

Die automatische Farbe kann nur dann ausgewählt werden, wenn die Eigenschaft [ModeColor](#) auf [WinModeColorPalette](#) gesetzt ist.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe [Beispiel](#)).

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle [WinPropGet\(\)](#) und [WinPropSet\(\)](#) gelesen bzw. geändert werden.

ColBkg

Hintergrundfarbe

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet(),

Beispiel

In dieser Eigenschaft können **Farbkonstanten** gesetzt werden. Die Konstante setzt die Farbe des Hintergrundes für dieses Objekt.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beim Objekt RtfEdit kann der Eigenschaft ein Parameter übergeben werden. Der Parameter bestimmt, welcher Textabschnitt von der neuen Farbe betroffen ist. Folgende Parameter können angegeben werden:

- WinEditAll

Der gesamte Text wird in der neuen Farbe angezeigt.

- WinEditMark

Nur der markierte Text wird in der neuen Farbe angezeigt. Die Markierung wird über die Eigenschaft Range bestimmt.

- WinEditWord

Nur das Wort, in dem der Cursor steht, wird in der neuen Farbe angezeigt. Die Position des Cursors wird über die Eigenschaft range bestimmt.



Bei den Druck-Objekten PrtText, PrtRtf, PrtMetaPicture und PrtLine kann als Hintergrundfarbe WinColTransparent angegeben werden. Diese Objekte können dann über andere Objekte (zum Beispiel ein Bild) gedruckt werden.

Bei Interval- und Box-Objekten kann die Hintergrundfarbe Transparenzen enthalten. Ein Farbwert mit Transparenzen wird mit dem Befehl WinColorOpacitySet() erzeugt.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Beispiele:

```
// Hintergrundfarbe BlauaEvt:Obj->wpColBkg # _WinColBlue;// Die ersten 10 Zeichen im Text gelb hi
```

Kontakt

ColBkgApp Hintergrundfarbe der Applikation oder des Objekts Typ int

Beispiel Siehe
Liste, Objekte, WinPropGet(), WinPropSet(),

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Die Konstante setzt die Hintergrundfarbe für den Ausgabebereich.



Bei Scrollbox-Objekten definiert diese Eigenschaft die Farbe des Scrollbereiches und die Eigenschaft ColBkg die Hintergrundfarbe des Ausgabebereiches.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->wpColBkgApp # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

ColBkgBase

Basis Füllfarbe setzen

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet(),

Beispiel

Über die Eigenschaft ColBkgBase kann die Basis-Füllfarbe des GanttGraph- bzw. PrtGanttGraph- Objektes festgelegt werden. Dies ist die Farbe, wo kein Inhalt des Gantt-Graphen ausgegeben wird, z. B. die linke obere Ecke und rechts unten zwischen den Scrollbalken.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

ColBkgOffPrint

Farbe des nicht druckbaren Bereiches

Typ int

Siehe Liste, Objekte, WinPropGet(),

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Die Konstante setzt die Farbe des nicht druckbaren Bereiches auf der Seite.

Soll der nicht druckbare Bereich nicht angezeigt werden, kann diese Eigenschaft auf den Wert WinColTransparent gesetzt werden.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->wpColBkgOffPrint # _WinColBlue;
```

Die Eigenschaft kann nur über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

ColBkgPage



Farbe der Seite

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Die Eigenschaft definiert die Farbe für das Seiten-Objekt. Die Farbe der untergeordneten Objekte der Seite wird über die Eigenschaft ColBkg zugewiesen.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

ColBkgTabArea Hintergrundfarbe der Fläche hinter den Registerreitern Typ int

Siehe Liste, Objekte, WinPropGet(), WinPropSet(), Beispiel

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Die Konstante setzt die Hintergrundfarbe der Fläche der Registerreiters.

Wird WinColUndefined eingestellt, wird die Fläche schattiert dargestellt (ausgehend von der Hintergrundfarbe, die in der Eigenschaft ColBkg eingestellt ist).

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->wpColBkgTabArea # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

ColBkgTabSel



Hintergrundfarbe des ausgewählten Registerreiters

Typ [int](#)

[Liste](#), [Objekte](#), [WinPropGet\(\)](#), [WinPropSet\(\)](#),

Siehe [Beispiel](#)

In dieser Eigenschaft können [Farbkonstanten](#) gesetzt werden.

Die Konstante setzt die Hintergrundfarbe des ausgewählten Registerreiters. Bei Verwendung von [WinColParent](#) wird die Hintergrundfarbe des übergeordneten Fenster-Objektes benutzt.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe [Beispiel](#)).

Beispiel:

```
aEvt:Obj->wpColBkgTabSel # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle [WinPropGet\(\)](#) und [WinPropSet\(\)](#) gelesen bzw. geändert werden.

Kontakt

ColBoundary



Farbe des Begrenzungsrechtecks

Typ int

Siehe Liste, Objekte, Beispiel

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Die Eigenschaft wird in den Objekten unterschiedlich verwendet:

- DocView-Objekt

Innerhalb des Objekts wird jede Seite mit einem Rand angezeigt. Die Farbe des Randes wird in dieser Eigenschaft gesetzt.

- GanttGraph, PrtGantGraph und Canvas

Die Eigenschaft setzt die Farbe für das Begrenzungsrechteck in den Objekten.

Standardmäßig ist diese hellrot. Bei WinColUndefined wird kein Begrenzungsrechteck gezeichnet.

- GroupTile

Die Eigenschaft setzt die Farbe für den Rand des Objekts. Die Breite des Randes wird beim GroupSplit-Objekt in der Eigenschaft TileBorder angegeben.

- Label

Die Eigenschaft setzt die Farbe für die Schriftbegrenzung in dem Objekt. Bei WinColTransparent und WinColUndefined wird keine Schriftbegrenzung gezeichnet. Diese Eigenschaft wirkt sich nur auf Labels aus, die in einem Frame mit dem Stil WinWndFrameShape in der ersten Ebene sind. Durch Verwendung dieser Eigenschaft, kann man Schriftzüge hervorheben, die durch ähnliche Hintergrundfarbe, wie die Textfarbe nicht sichtbar wären.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->wpColBoundary # _WinColBlue;
```

Kontakt

ColCellActive



Farbe für die Zellmarkierungen

Typ int

Siehe Liste, Objekte, CellActiveHorz,

CellActiveVert, Beispiel

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Die Konstante legt die Farbe für die Zeilenmarkierung fest. Bei WinColUndefined wird keine Markierung gezeichnet. Soll für die Spalten- und die Zeilenmarkierung unterschiedliche Farben verwendet werden, muss zusätzlich die Eigenschaft ColCellActiveHorz gesetzt werden.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->wpColCellActive # _WinColBlue;
```

Eine Markierung wird nur dann gezeichnet, wenn in mindestens einer der Eigenschaften CellActiveHorz oder CellActiveVert ein Wert größer -1 eingetragen wurde.

Kontakt

ColCellActiveHorz



Farbe für die Zellmarkierungen

Typ int

Siehe Liste, Objekte, CellActiveHorz,

CellActiveVert, Beispiel

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Die Konstante legt die Farbe für die Spaltenmarkierung fest. Bei WinColUndefined wird die Farbeinstellung der Eigenschaft ColCellActive verwendet. Soll keine Spaltenmarkierung gezeichnet werden, kann die Farbe auf WinColTransparent gesetzt werden.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->wpColCellActiveHorz # _WinColBlue;
```

Eine Markierung wird nur dann gezeichnet, wenn in mindestens einer der Eigenschaften CellActiveHorz oder CellActiveVert ein Wert größer -1 eingetragen wurde.

Kontakt

ColDateFg

Vordergrundfarbe

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet(),

Beispiel

In dieser Eigenschaft können **Farbkonstanten** gesetzt werden. Die Konstante setzt die Farbe des Vordergrundes in einem Kalenderobjekt. Es werden alle Zahlen des angezeigten Monates in dieser Farbe dargestellt.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->wpColDateFg # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Um die Eigenschaft prozedural im Calendar eines DateEdit-Objekts zu setzen, wird diese beim DateEdit selbst eingetragen. Beim Öffnen des Popups durch den Anwender wird der Wert automatisch an das Calendar-Objekt übergeben.

ColDateMonth

Hintergrundfarbe eines Monats

Typ int

Siehe Liste, Objekte,

Siehe WinPropGet(), WinPropSet()

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Die Konstante setzt die Farbe des Hintergrundes für dieses Objekt.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->wpColDateMonth # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Um die Eigenschaft prozedural im Calendar eines DateEdit-Objekts zu setzen, wird diese beim DateEdit selbst eingetragen. Beim Öffnen des Popups durch den Anwender wird der Wert automatisch an das Calendar-Objekt übergeben.

ColDateTitleBkg

Hintergrundfarbe des Monatstitels

Typ int
Siehe Liste, Objekte, WinPropGet(),

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Die Konstante setzt die Farbe des Hintergrundes für den Titel dieses Objektes.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->wpColDateTitleBkg # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Um die Eigenschaft prozedural im Calendar eines DateEdit-Objekts zu setzen, wird diese beim DateEdit selbst eingetragen. Beim Öffnen des Popups durch den Anwender wird der Wert automatisch an das Calendar-Objekt übergeben.

ColDateTitleFg

Vordergrundfarbe des Monatstitels

Typ int
Siehe Liste, Objekte, WinPropGet(),

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Die Konstante setzt die Farbe des Vordergrundes für den Titel dieses Objektes.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->wpColDateTitleFg # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Um die Eigenschaft prozedural im Calendar eines DateEdit-Objekts zu setzen, wird diese beim DateEdit selbst eingetragen. Beim Öffnen des Popups durch den Anwender wird der Wert automatisch an das Calendar-Objekt übergeben.

ColDateTrailing

Trailing-Farbe

Typ int

Liste,

Objekte,

Siehe WinPropGet(),

WinPropSet(),

Beispiel

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Die Konstante setzt die Farbe des Vordergrundes in einem Kalenderobjekt. Es werden alle Zahlen des vorigen und folgenden Monats in dieser Farbe dargestellt.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->wpColDateTrailing # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Um die Eigenschaft prozedural im Calendar eines DateEdit-Objekts zu setzen, wird diese beim DateEdit selbst eingetragen. Beim Öffnen des Popups durch den Anwender wird der Wert automatisch an das Calendar-Objekt übergeben.

ColDesign

Farbe im Design-Modus

Typ int
Siehe Liste, Objekte,

Beispiel

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Das Objekt bekommt einen Rand in der angegebenen Farbe, wenn die Eigenschaft Design gesetzt wird.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->wpColDesign # _WinColBlue;aEvt:Obj->wpDesign # true;
```

Kontakt

ColDisabledBkg Hintergrundfarbe des selektierten

Datensatzes Typ int

Siehe [Liste](#), [Objekte](#), [ColDisabledFg](#),

[WinPropGet\(\)](#), [WinPropSet\(\)](#), [Beispiel](#)

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Die Konstante setzt die Farbe des Hintergrundes für den selektierten Datensatz, wenn das Objekt deaktiviert ist.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe [Beispiel](#)).

Beispiel:

```
aEvt:Obj->wpColDisabledBkg # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle [WinPropGet\(\)](#) und [WinPropSet\(\)](#) gelesen bzw. geändert werden.

Kontakt

ColDisabledFg



Vordergrundfarbe des selektierten Datensatzes

Typ int

Liste, Objekte, ColDisabledBkg,

Siehe WinPropGet(), WinPropSet(),
Beispiel

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Die Konstante setzt die Farbe des Vordergrundes für den selektierten Datensatz, wenn das Objekt deaktiviert ist.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->wpColDisabledFg # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

ColEdit



Hintergrundfarbe der Gruppe eines RecViews im Bearbeitungsmodus Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Die Konstante setzt die Hintergrundfarbe der aktiven Gruppe im Bearbeitungsmodus des RecViews.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->wpColEdit # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.



Die Eigenschaft kann in den Ereignissen EvtLstViewInit, EvtLstGroupInit, EvtLstEditStartGroup, EvtLstEditStartItem, EvtLstEditActivate und EvtLstEditEndItem nicht verändert werden.

Kontakt

ColErrRangeBkg Hintergrundfarbe bei
fehlerhafter Eingabe Typ int

Siehe [Liste](#), [Objekte](#), [ColErrRangeFg](#),
[WinPropGet\(\)](#), [WinPropSet\(\)](#), [Beispiel](#)

In dieser Eigenschaft können **Farbkonstanten** gesetzt werden. Durch diese Eigenschaft kann eine farbliche Hervorhebung des Eingabefeldes erfolgen, wenn der Inhalt den zulässigen Bereich unter- oder überschreitet.

Der zulässige Bereich ist durch die Eigenschaften **MinInt** und **MaxInt** (für **IntEdit**-Objekte) bzw. **MinFloat** und **MaxFloat** (für **FloatEdit**-Objekte) definiert. Werte außerhalb dieses Bereiches werden in der Vordergrundfarbe **ColErrRangeFg** und der Hintergrundfarbe **ColErrRangeBkg** hervorgehoben. Ist diese Eigenschaft auf WinColUndefined gesetzt, findet keine farbliche Hervorhebung statt.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe [Beispiel](#)).

Beispiel:

```
aEvt:Obj->wpColErrRangeBkg # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle [WinPropGet\(\)](#) und [WinPropSet\(\)](#) gelesen bzw. geändert werden.

Kontakt

ColErrRangeFg  Vordergrundfarbe bei fehlerhafter
Eingabe Typ int

Siehe [Liste](#), [Objekte](#), [ColErrRangeBkg](#),
[WinPropGet\(\)](#), [WinPropSet\(\)](#), [Beispiel](#)

In dieser Eigenschaft können **Farbkonstanten** gesetzt werden. Durch diese Eigenschaft kann eine farbliche Hervorhebung des Eingabefeldes erfolgen, wenn der Inhalt den zulässigen Bereich unter- oder überschreitet.

Der zulässige Bereich ist durch die Eigenschaften **MinInt** und **MaxInt** (für **IntEdit**-Objekte) bzw. **MinFloat** und **MaxFloat** (für **FloatEdit**-Objekte) definiert. Werte außerhalb dieses Bereiches werden in der Vordergrundfarbe **ColErrRangeFg** und der Hintergrundfarbe **ColErrRangeBkg** hervorgehoben. Ist diese Eigenschaft auf WinColUndefined gesetzt, findet keine farbliche Hervorhebung statt.

Wird ein neues Eingabeobjekt erzeugt, wird diese Eigenschaft auf WinColLightRed gesetzt.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe [Beispiel](#)).

Beispiel:

```
aEvt:Obj->wpColErrRangeFg # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle [WinPropGet\(\)](#) und [WinPropSet\(\)](#) gelesen bzw. geändert werden.

ColFg

Vordergrundfarbe

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet(),

Beispiel

In dieser Eigenschaft können **Farbkonstanten** gesetzt werden. Die Konstante setzt die Farbe des Vordergrundes für dieses Objekt.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->wpColFg # _WinColBlue;
```

Wird diese Eigenschaft beim Systemdialog für **Schriftarten** gesetzt, können nur die 16 Grundfarben angegeben werden, da das Betriebssystem nur diese Farben für Schriften unterstützt.

In dem Systemdialog kann nur dann eine Farbe angezeigt oder ausgewählt werden, wenn in der Eigenschaft Flags der Wert WinComFontEffects gesetzt wurde.

Beim Objekt RtfEdit kann der Eigenschaft ein Parameter übergeben werden. Der Parameter bestimmt, welcher Textabschnitt von der neuen Farbe betroffen ist. Folgende Parameter können angegeben werden:

- WinEditAll

Der gesamte Text wird in der neuen Farbe angezeigt.

- WinEditMark

Nur der markierte Text wird in der neuen Farbe angezeigt. Die Markierung wird über die Eigenschaft range bestimmt.

- WinEditWord

Nur das Wort, in dem der Cursor steht, wird in der neuen Farbe angezeigt. Die Position des Cursors wird über die Eigenschaft range bestimmt.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

ColFgTabSel



Vordergrundfarbe des selektierten Registerreiters

Typ int

Liste, Objekte, WinPropGet(), WinPropSet(),

Siehe Beispiel

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Die Konstante setzt die Vordergrundfarbe des ausgewählten Registerreiters. Bei Verwendung von WinColParent wird die Hintergrundfarbe des übergeordneten Fenster-Objektes benutzt.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->wpColFgTabSel # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

ColFillEven

Farbe der geraden Markierung

Typ int

Liste, Objekte,

Siehe PrtPropGet(), PrtPropSet(),

ColFillOdd

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Ist bei diesem Objekt die Eigenschaft StylePen auf PrtStylePenBroken, PrtStylePenDotted oder PrtStylePenCircles gesetzt, werden alle geraden Begrenzungen in dieser Farbe dargestellt.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->ppColFillEven # _WinColLightBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle PrtPropGet() und PrtPropSet() gelesen bzw. geändert werden.

ColFillOdd

Farbe der ungeraden Markierungen

Typ int

Siehe Liste, Objekte, PrtPropGet(),

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Ist bei diesem Objekt die Eigenschaft StylePen auf PrtStylePenBroken, PrtStylePenDotted oder PrtStylePenCircles gesetzt, werden alle ungeraden Begrenzungen in dieser Farbe dargestellt.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->ppColFillOdd # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle PrtPropGet() und PrtPropSet() gelesen bzw. geändert werden.

Kontakt

ColFocusBkg



Hintergrundfarbe, wenn das Objekt fokussiert ist

Typ int
Siehe Liste, Objekte, ColFocusFg,

WinPropGet(), WinPropSet(), Beispiel

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Die Konstante setzt die Hintergrundfarbe des Objektes, wenn der Eingabefokus in diesem Objekt steht.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->wpColFocusBkg # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

ColFocusFg Vordergrundfarbe, wenn das Objekt fokussiert
ist Typ int

Siehe
[Liste](#), [Objekte](#), [ColFocusBkg](#), [WinPropGet\(\)](#),
[WinPropSet\(\)](#), [Beispiel](#)

In dieser Eigenschaft können **Farbkonstanten** gesetzt werden. Die Konstante setzt die Vordergrundfarbe des Objektes, wenn der Eingabefokus in diesem Objekt steht.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe [Beispiel](#)).

Beispiel:

```
aEvt:Obj->wpColFocusFg # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle [WinPropGet\(\)](#) und [WinPropSet\(\)](#) gelesen bzw. geändert werden.



ColFocusOffBkg

Hintergrundfarbe des Fokus, wenn das Objekt nicht fokussiert ist Typ int

Siehe
Liste, Objekte, ColFocusBkg, ColFocusOffFg, WinPropGet(),
WinPropSet(), Beispiel
In dieser Eigenschaft können Farbkonstanten gesetzt werden. Die Konstante setzt die Hintergrundfarbe des Fokus innerhalb des Objektes, wenn der Eingabefokus nicht in diesem Objekt steht. Die Eigenschaft kann in allen Objekten gesetzt werden, in denen Inhalt markiert werden kann und diese Markierung auch dann noch angezeigt wird, wenn der Fokus nicht mehr in diesem Objekt ist.

Die angegebene Farbe wird zur Darstellung des markierten Inhaltes verwendet.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->wpColFocusOffBkg # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

ColFocusOffFg



Vordergrundfarbe des Fokus, wenn das Objekt nicht fokussiert ist Typ int

Liste, Objekte, ColFocusFg, ColFocusOffBkg, WinPropGet(),
WinPropSet(), Beispiel Siehe
In dieser Eigenschaft können Farbkonstanten gesetzt werden. Die Konstante setzt die Vordergrundfarbe des Fokus des Objektes, wenn der Eingabefokus nicht in diesem Objekt steht. Die Eigenschaft kann in allen Objekten gesetzt werden, in denen Inhalt markiert werden kann und diese Markierung auch dann noch angezeigt wird, wenn der Fokus nicht mehr in diesem Objekt ist.

Die angegebene Farbe wird zur Darstellung des markierten Inhaltes verwendet.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden.

Beispiel:

```
aEvt:Obj->wpColFocusOffFg # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehl WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

ColGrid

Farbe des Gitters

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet(),

Beispiel

In dieser Eigenschaft können **Farbkonstanten** gesetzt werden. Die Konstante setzt die Farbe des Gitters innerhalb des Objektes. Mit der Farbe WinColUndefined kann das Gitter ausgeblendet werden.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->wpColGrid # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

ColHover



Farbe, wenn der Benutzer mit der Maus auf dem Objekt steht Typ int

Siehe Liste, Objekte, WinPropGet(), WinPropSet(), Beispiel

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Diese Farbe bekommt der Hyperlink, wenn der Benutzer mit der Maus auf dem Hyperlink steht.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->wpColHover # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

ColIvlArrows

Farbe der Intervall-Pfeile

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet(), Beispiel

Über die Eigenschaft ColIvlArrows kann die Pfeilfarbe der Intervalle eingestellt werden. Der Farbwert WinColTransparent schaltet die Pfeilausgabe ab. Durch WinColUndefined werden die Pfeile in der Standard-Farbe gezeichnet.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

ColIvlBorder

Rahmenfarbe für Intervalle

Typ int

Liste, Objekte,

Siehe WinPropGet(),
WinPropSet(), Beispiel

Über die Eigenschaft ColIvlBorder kann die Rahmenfarbe für Intervalle eingestellt werden. Die Farbkonstante WinColTransparent setzt die Rahmenfarbe auf die Hintergrundfarbe des Intervalls. Durch WinColUndefined wird der Rahmen in der Standard-Farbe gezeichnet.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

ColLink



Farbe, bevor der Hyperlink ausgeführt wurde

Typ int

Siehe Liste, Objekte, WinPropGet(),

WinPropSet(), Beispiel

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->wpColLink # WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

ColorMode

ColorMode-Modus

Typ int

Liste, Objekte,

Siehe PrtPropGet(),

PrtPropSet()

Die Eigenschaft definiert den Farbmodus beim Drucken (Farbe oder monochrom).

PrtColModeNone Farbmodus unbekannt

PrtColModeMono Monochromdruck

PrtColModeColor Farbdruck

Die Eigenschaft kann beim Objekt PpcObject nur gelesen werden.

ColParent

Elternfarbe setzen

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet(),

Beispiel

Diese Eigenschaft definiert die Eltern-Farbe. Sie wird nur dann ausgewertet, wenn die Eigenschaft ModeColor auf den Wert WinModeColorFull gesetzt ist. Nur in diesem Fall kann die schraffierte Schaltfläche (WinColParent) ausgewählt werden.

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Die Konstante setzt die Farbe des Hintergrundes für dieses Objekt.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Enthält die Eigenschaft ColParent den Wert WinColParent gibt es unterschiedliche Fälle für die angezeigte und ausgewählte Farbe:

1. Der ColorButton hat die ID = -1302 (= Hintergrundfarbe für das RtfEdit-Objekt)

Befindet sich der ColorButton in einer ToolbarRtf, wird die Hintergrundfarbe (Eigenschaft ColBkg) des zugeordneten RtfEdit-Objektes ausgewählt. Ansonsten wird die Hintergrundfarbe (Eigenschaft ColBkg) des ColorButton ausgewählt.

2. Der ColorButton hat die ID = -1301 (= Textfarbe für das RtfEdit-Objekt)

Befindet sich der ColorButton in einer ToolbarRtf wird die Vordergrundfarbe (Eigenschaft ColFg) des zugeordneten RtfEdit-Objektes ausgewählt. Ansonsten wird die Vordergrundfarbe (Eigenschaft ColFg) des ColorButton ausgewählt.

3. Der ColorButton hat nicht die ID -1301 oder -1302

In diesem Fall kann keine gültige Farbe ermittelt werden, denn die Unterscheidung ob es sich um eine Hintergrund- oder Vordergrundfarbe handelt ist nicht gegeben. Es wird eine schraffierte Fläche angezeigt.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

ColSeparator



Farbe des Trenners zwischen fixierten und nicht fixierten Spalten Typ int

Siehe Liste, Objekte, ClmFixed, Beispiel

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Die Konstante setzt die Farbe der Gitterlinie zwischen den fixierten (siehe ClmFixed) und nicht fixierten Spalten der Liste. Mit der Farbe WinColUndefined kann das Gitter ausgeblendet werden.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->wpColSeparator # _WinColBlue;
```

Kontakt

ColVisited Farbe, nachdem der Hyperlink ausgeführt wurde Typ int

Liste, Objekte, WinPropGet(), WinPropSet(), Siehe
Beispiel

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Diese Farbe bekommt der Hyperlink, wenn er von einem Benutzer mit einen Klick ausgeführt wurde.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->wpColVisited # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

GroupColBkg

Hintergrundfarbe einer Gruppe

Typ int

Siehe Liste, Objekte, Beispiel

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Die Konstante setzt die Hintergrundfarbe der Gruppe. Die gesetzte Hintergrundfarbe wirkt sich auf alle Items aus, bei denen die Eigenschaft ClmColBkg auf WinColParent gesetzt ist.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Diese Eigenschaft wirkt sich nur auf die Gruppe aus und kann im Ereignis

EvtLstGroupInit gesetzt werden.

Beispiel:

```
aGroup->wpGroupColBkg # _WinColBlue;
```

Die Eigenschaft kann über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

GroupColFg

Vordergrundfarbe einer Gruppe

Typ int

Siehe Liste, Objekte, Beispiel

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Die Konstante setzt die Vordergrundfarbe der Gruppe. Die gesetzte Vordergrundfarbe wirkt sich auf alle Items aus, bei denen die Eigenschaft ClmColFg auf WinColParent gesetzt ist.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Diese Eigenschaft wirkt sich nur auf die Gruppe aus und kann im Ereignis

EvtLstGroupInit gesetzt werden.

Beispiel:

```
aGroup->wpGroupColFg # _WinColBlue;
```

Die Eigenschaft kann über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

HdrColFg



Vordergrundfarbe des Spaltenkopfes

Typ **int**

[Liste](#), [Objekte](#), [WinPropGet\(\)](#),

Siehe [WinPropSet\(\)](#), [HdrShadeCol1](#),
[HdrShadeCol2](#), [HdrShadeMode](#)

Diese Eigenschaft bestimmt die Textfarbe des Spaltenkopfes sowie die Farbe des Trenners zwischen den einzelnen Spalten. Standardmäßig steht dieser Wert aus Kompatibilitätsgründen auf [WinColParent](#).

HdrShadeCol1

Füllfarbe des Spaltenkopfes

Typ int

Liste, Objekte,
WinPropGet(),

Siehe WinPropSet(),

HdrColFg,
HdrShadeCol2,
HdrShadeMode

Diese Eigenschaft bestimmt die Füllfarbe des Spaltenkopfes. Standardmäßig hat die Eigenschaft aus Kompatibilitätsgründen den Wert WinColParent. In Verbindung mit den Eigenschaften HdrShadeCol2 und HdrShadeMode können auch Farbverläufe dargestellt werden.

Kontakt

HdrShadeCol2



Farbe des Spaltenkopfes für Farbverlauf

Typ int

Liste, Objekte, WinPropGet(),

Siehe WinPropSet(), HdrColFg,

HdrShadeCol1, HdrShadeMode

Die Eigenschaft ist die zweite Farbe für die Darstellung eines Farbverlaufs in Zusammenhang mit den Eigenschaften HdrShadeCol1 und HdrShadeMode. Hat die Eigenschaft den Wert WinColParent, wird die Farbe WinColBtnFace verwendet.

Kontakt

HdrShadeMode



Modus des Farbverlaufs des Spaltenkopfes

Typ int

Liste, Objekte, WinPropGet(),

Siehe WinPropSet(), HdrColFg,

HdrShadeCol1, HdrShadeCol2

Diese Eigenschaft bestimmt den Modus des Farbverlaufs des Spaltenkopfes. Sie kann folgende Werte annehmen:

WinShadeSolid Der Spaltenkopf wird mit der Farbe HdrShadeCol1 gefüllt.

WinShadeTopBottom Der Farbverlauf wird von oben (HdrShadeCol1) nach unten (HdrShadeCol2) gezeichnet.

WinShadeBottomTop Der Farbverlauf wird von unten nach oben gezeichnet.

WinShadeLeftRight Der Farbverlauf wird von links nach rechts gezeichnet.

WinShadeRightLeft Der Farbverlauf wird von rechts nach links gezeichnet.

Kontakt

ScalaColorBkg



Hintergrundfarbe der Skala Typ

int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Mit dieser Eigenschaft wird die Hintergrundfarbe der Skala festgelegt. Abhängig von der Eigenschaft SubDivisions der Achse, wird die Hintergrundfarbe um 25% aufgehellt (bei maximaler Intensität abgedunkelt) um einen Abschnittswechsel anzuzeigen.

Kontakt

ScalaColorBkgBase | | Hintergrundfarbe der Skala (ungerade Zeilen/Spalten) Typ int

Siehe
Liste, Objekte, ScalaColorBkg, WinPropGet(),
WinPropSet()

Mit dieser Eigenschaft wird die Hintergrundfarbe der Skala für ungerade Zeilen bzw. Spalten festgelegt.

Ist die Eigenschaft nicht gesetzt, wird abhängig von der Eigenschaft SubDivisions der Achse, die Hintergrundfarbe um 25% aufgehellt (bei maximaler Intensität abgedunkelt) um einen Abschnittswechsel anzuzeigen.

Kontakt

ScalaColorEdge



Farbe der Trennung zwischen Überschrift und Anzeigebereich Typ int

Siehe Liste, Objekte, WinPropGet(), WinPropSet()

Mit dieser Eigenschaft kann die Farbe einer Trennlinie eingestellt werden. Diese Linie wird am unteren (horizontale Achse) bzw. rechten Rand (vertikale Achse) der Skala dargestellt.

Enthält die Eigenschaft den Wert WinColUndefined (default) wird keine Trennlinie dargestellt.

Kontakt

ScalaColorEnd | | Farbe des ungenutzten Skala-Bereichs Typ int

Siehe Liste, Objekte

Mit dieser Eigenschaft wird die Farbe des nicht verwendeten Skala-Bereichs festgelegt. Die Skala muss in der Eigenschaft ScalaLabels der Achse mit '~' abgeschlossen werden. Der verbleibende Platz des Hintergrundes wird dann mit der hier angegebenen Farbe gefüllt.

Kontakt

ScalaColorText

Farbe des Textes in der Skala

Typ int

Siehe Liste, Objekte,

WinPropGet(), WinPropSet()

Mit dieser Eigenschaft wird die Farbe des Textes in der Skala festgelegt.

ShadeCol1

Erste Farbe des Farbverlaufs

Typ int
Siehe Liste, Objekte,

ShadeMode

In dieser Eigenschaft können Farbkonstanten gesetzt werden. Die Konstante setzt die erste Farbe des Farbverlaufs der Gruppen bzw. die Hintergrundfarbe.

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe Beispiel).

Beispiel:

```
aEvt:Obj->wpShadeCol1 # _WinColBlue;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.



Die Eigenschaft kann in den Ereignissen EvtLstViewInit, EvtLstGroupInit, EvtLstEditStartGroup, EvtLstEditStartItem, EvtLstEditActivate und EvtLstEditEndItem nicht verändert werden.

ShadeCol2

Letzte Farbe des Farbverlaufs

Typ [int](#)

Siehe [Liste](#), [Objekte](#), [ShadeMode](#)

In dieser Eigenschaft können [Farbkonstanten](#) gesetzt werden. Die Konstante setzt die letzte Farbe des Farbverlaufs der [Gruppen](#).

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe [Beispiel](#)).

Beispiel:

```
aEvt:Obj->wpShadeCol2 # _WinColRed;
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle [WinPropGet\(\)](#) und [WinPropSet\(\)](#) gelesen bzw. geändert werden.



Die Eigenschaft kann in den Ereignissen [EvtLstViewInit](#), [EvtLstGroupInit](#), [EvtLstEditStartGroup](#), [EvtLstEditStartItem](#), [EvtLstEditActivate](#) und [EvtLstEditEndItem](#) nicht verändert werden.

Kontakt

ShadeMode



Modus des Farbverlaufs im sichtbaren Bereich des Views Typ int

Siehe Liste, Objekte

Diese Eigenschaft bestimmt den Modus des Farbverlaufs im sichtbaren Bereich des Views. Sie kann folgende Werte annehmen:

WinShadeSolid Der Ausgabebereich wird mit der Farbe ShadeCol1 gefüllt.

WinShadeTopBottom Der Farbverlauf wird von oben (ShadeCol1) nach unten (ShadeCol2) gezeichnet.

WinShadeBottomTop Der Farbverlauf wird von unten nach oben gezeichnet.

WinShadeLeftRight Der Farbverlauf wird von links nach rechts gezeichnet.

WinShadeRightLeft Der Farbverlauf wird von rechts nach links gezeichnet.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.



Die Eigenschaft kann in den Ereignissen EvtLstViewInit, EvtLstGroupInit, EvtLstEditStartGroup, EvtLstEditStartItem, EvtLstEditActivate und EvtLstEditEndItem nicht verändert werden.

Kontakt

TitleColorBkg

Hintergrundfarbe des Achsentitels

Typ int
Siehe Liste, Objekte, WinPropGet(),

Mit dieser Eigenschaft wird die Hintergrundfarbe des Achsentitels festgelegt.

Kontakt

TitleColorFg

Vordergrundfarbe des Achsentitels

Typ int
Siehe Liste, Objekte, WinPropGet(),

Mit dieser Eigenschaft wird die Vordergrundfarbe des Achsentitels festgelegt.

Kontakt

Font (Schrift-Eigenschaften)

Eigenschaften

sortiert nach

Gruppen,

Siehe Alphabetische

Liste aller

Eigenschaften

<u>Font</u>	Schriftart des Objektes
<u>FontAttr</u>	Schriftartattribute
<u>FontAttrSel</u>	Schriftartattribute
<u>FontBox</u>	Schriftart des Box-Objekts
<u>FontIvl</u>	Schriftart des Intervalls
<u>FontName</u>	Referenz-Schriftart
<u>FontParent</u>	Übernehmen der Schriftart des übergeordneten Objektes
<u>FontParentScala</u>	Schriftart der Skala
<u>FontParentTitle</u>	Schriftart der Achse
<u>FontScala</u>	Schriftart der Skala
<u>FontSize</u>	Größe der Schriftart
<u>FontSizeTwips</u>	Zeichensatzgröße in Twips
<u>FontTitle</u>	Schriftart des Titels
<u>IndentLeft</u>	linker Absatzeinzug
<u>IndentOfs</u>	Einzug ab der zweiten Zeile
<u>IndentRight</u>	rechter Absatzeinzug
<u>LineSpacingTwips</u>	Zeilenabstand
<u>RtfAlign</u>	Textausrichtung
<u>RtfEffect</u>	Schriftattribute
<u>ThemeFontBox</u>	Schriftart des Themes für Box-Objekte
<u>ThemeFontCaption</u>	Schriftart des Themes der Titelzeile der Achse
<u>ThemeFontBox</u>	Schriftart des Themes für Interval-Objekte
<u>ThemeFontText</u>	Schriftart des Themes
<u>ThemeFontTextHdr</u>	Schriftart des Themes für Spaltenköpfe

DialogBoxFont

Schriftart der Dialogbox

Typ font

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Die Eigenschaft bestimmt die Schriftart zur Anzeige des Meldungstextes in Nachrichtenfenstern (WinDialogBox()), sofern die Option WinDialogBoxUseFont der Eigenschaft DialogBoxFlags beim App-Objekt gesetzt ist. Die Schriftart wird auch für Schaltflächen verwendet, sofern die option WinDialogBoxUseFontButton nicht gesetzt ist.



Diese Eigenschaft wirkt sich nicht aus, wenn das Elternfenster der Dialogbox im Modern Theme Style dargestellt (Eigenschaft StyleTheme = WinStyleThemeModern) wird.

Kontakt

Objekte von DialogBoxFont

Folgende Objekte unterstützen die Eigenschaft [DialogBoxFont](#)

[DialogBoxFont](#),

[Liste der](#)

Siehe [Objekte](#), [Liste](#)

[der](#)

[Eigenschaften](#)

- [Application](#)

Kontakt

DialogBoxFontButton Schriftart der Schaltflächen
der Dialogbox Typ font

Siehe [Liste](#), [Objekte](#), [WinPropGet\(\)](#),

Die Eigenschaft bestimmt die Schriftart zur Anzeige der Schaltflächen in Nachrichtenfenstern ([WinDialogBox\(\)](#)), sofern die Option [WinDialogBoxUseFontButton](#) der Eigenschaft [DialogBoxFlags](#) beim [App](#)-Objekt gesetzt ist.



Diese Eigenschaft wirkt sich nicht aus, wenn das Elternfenster der Dialogbox im [Modern Theme Style](#) dargestellt (Eigenschaft [StyleTheme](#) = [WinStyleThemeModern](#)) wird.

Kontakt

Objekte von DialogBoxFontButton

Folgende Objekte unterstützen die Eigenschaft [DialogBoxFontButton](#)

[DialogBoxFontButton](#),

[Liste der Objekte](#),

[Liste der](#)

[Eigenschaften](#)

- [Application](#)

Siehe

Font

Schriftart des Objekts

Typ font

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In dieser Eigenschaft wird die Schriftart des Objektes eingetragen. Der Font besteht neben der Schriftart aus dem Schriftgrad und verschiedenen Attributen (Fett, Kursiv, ...). Durch Ändern dieser Eigenschaft wird automatisch die Eigenschaft FontParent auf false gesetzt.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden. Zum Lesen und Setzen einer Schriftart wird eine Variable vom Typ font benötigt.

Beispiel:

```
// Font-Attribut "Fett" setzen oder entfernenFont # aEvt:Obj->wpFont;if (tFont:Attributes & _Win
```

FontAttr

Schriftartattribute

Typ int

Siehe Liste, Objekte

Mit dieser Eigenschaft kann eine Hervorhebung des Spalteninhaltes durch Fontattribute erzeugt werden.

Folgende Attribute können vergeben werden:

<u>WinFontAttrN</u> / <u>PrtFontAttrNormal</u>
<u>WinFontAttrB</u> / <u>PrtFontAttrBold</u>
<u>WinFontAttrI</u> / <u>PrtFontAttrItalic</u>
<u>WinFontAttrU</u> / <u>PrtFontAttrUnderline</u>
<u>WinFontAttrX</u> / <u>PrtFontAttrStrikeout</u>
<u>WinFontAttrBI</u>
<u>WinFontAttrIU</u>

Normal
Fett
Kursiv
Unterstrichen
Durchstreichen
Fett & Kursiv
Kursiv &
Unterstrichen

Wird diese Eigenschaft im Ereignis EvtLstDataInit gesetzt, wirkt sich das nur auf die aktuelle Zelle aus. Die Eigenschaft wird nicht für die gesamte Spalte geändert.



Im Modern Theme Style wirken sich Änderungen der Eigenschaft im EvtLstDataInit aus, sofern sich der gesetzte Wert von dem gespeicherten Wert in der Spalte unterscheidet. Wird der Wert in EvtLstDataInit auf den gespeicherten Wert gesetzt oder nicht verändert, wird die jeweilige Eigenschaft von dem Theme-Objekt verwendet.

FontAttrSel

Schriftartattribute

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Hierüber kann das Font-Attribut des selektierten Objekts ausgewählt werden.

Folgende Attribute können vergeben werden:

WinFontAttrN Normal

WinFontAttrB Fett

WinFontAttrI Kursiv

WinFontAttrU Unterstreichen

WinFontAttrX Durchstreichen

WinFontAttrBI Fett & Kursiv

WinFontAttrIU Kursiv & Unterstreichen

FontBox

Font für Box-Objekte

Typ font

Liste, Objekte,
WinPropGet(),

Siehe WinPropSet(),
PrtPropGet(),
PrtPropSet()

In dieser Eigenschaft wird der Zeichensatz der Beschriftung von Box-Objekten in einem GanttGraph- oder PrtGanttGraph-Objekt eingetragen. Der Font besteht neben der Schriftart aus dem Schriftgrad und verschiedenen Attributen (Fett, Kursiv, ...). Die Eigenschaft FontParent wird für diese Eigenschaft nicht ausgewertet.

Standardmäßig enthält die Eigenschaft den Font "Arial" ohne Schriftartattribute (Normal).

Enthält die Eigenschaft FontAttr eines Box-Objekts die Angabe WinFontAttrN (default), wird das Schriftart-Attribut der Eigenschaft FontBox verwendet.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() bzw. PrtPropGet() und PrtPropSet() gelesen und geändert werden. Zum Lesen und Setzen einer Schriftart wird eine Variable vom Typ font benötigt.

Beispiel:

```
// Font-Attribut "Fett" setzen oder entfernen tFont # tBox->wpFontAttr;if (tFont:Attributes & _Wi
```

FontIvl

Schriftart des Objekts

Typ font

Liste, Objekte,

Siehe WinPropGet(),
WinPropSet()

In dieser Eigenschaft wird die Schriftart von Intervallen in einem GanttGraph- oder PrtGanttGraph-Objekt eingetragen. Der Font besteht neben der Schriftart aus dem Schriftgrad und verschiedenen Attributen (Fett, Kursiv, ...). Die Eigenschaft FontParent darf nicht gesetzt sein.

Standardmäßig enthält die Eigenschaft den Font "Arial" ohne Schriftartattribute (Normal).

Enthält die Eigenschaft FontAttr eines Intervalls die Angabe WinFontAttrN (default), wird das Schriftart-Attribut der Eigenschaft FontIvl verwendet.

Der Schriftgrad in FontIvl wird ignoriert, da eine automatische Skalierung erfolgt.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden. Zum Lesen und Setzen einer Schriftart wird eine Variable vom Typ font benötigt.

Beispiel:

```
// Font-Attribut "Fett" setzen oder entfernenFont # tIvl->wpFontAttr;if (tFont:Attributes & _Win
```

FontName

Referenz-Schriftart

Typ alpha

Liste, Objekte,

Siehe FontSize,

WinPropGet(),

WinPropSet()

In dieser Eigenschaft wird die Schriftart eingetragen. Die eingetragene Schriftart dient dem Objekt als Referenz für andere Eigenschaften. So werden zum Beispiel im Objekt FontSizeEdit nur die Schriftgrößen dargestellt, die in dieser Schriftart definiert sind.

Beim Objekt RtfEdit kann der Eigenschaft ein Parameter übergeben werden. Der Parameter bestimmt, welcher Textabschnitt von der neuen Schrift betroffen ist. Folgende Parameter können angegeben werden:

- WinEditAll

Der gesamte Text wird in der neuen Schrift angezeigt.

- WinEditMark

Nur der markierte Text wird in der neuen Schrift angezeigt. Die Markierung wird über die Eigenschaft Range bestimmt.

- WinEditWord

Nur das Wort, in dem der Cursor steht, wird in der neuen Schrift angezeigt. Die Position des Cursors wird über die Eigenschaft Range bestimmt.

Wird ein unformatierter Text in das RtfEdit-Objekt gelesen, kann mit diesen Optionen eine Formatierung des Textes vorgenommen werden.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Beispiele:

```
$RtfEdit->wpFontName # 'Arial'; // Arial verwenden$RtfEdit->wpFontName(_WinEditAll) # 'Arial'; //
```

Kontakt

FontParent  Übernehmen der Schriftart vom übergeordneten Objekt. Typ logic

Siehe Liste, Objekte

Wenn die Eigenschaft FontParent auf true gesetzt ist, wird die Schriftart (Eigenschaft Font) vom übergeordneten Objekt übernommen.

Wird diese Eigenschaft im Ereignis EvtLstDataInit gesetzt, wirkt sich das nur auf die aktuelle Zelle aus. Die Eigenschaft wird nicht für die gesamte Spalte geändert.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.



Ist bei Objekten, die einem PrintForm-Objekt untergeordnet sind, diese Eigenschaft auf true gesetzt, wird der Font vom PrintForm-Objekt nur verwendet, wenn auch bei diesem die Eigenschaft ebenfalls auf true gesetzt ist. Sonst wird der Font von der Job-Seite (siehe PrtJobWrite() bei PrtAdd() verwendet.

Kontakt

FontParentScala

Schriftart der Skala

Typ logic

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Durch das Setzen dieser Eigenschaft wird ein Font für den Titel der Skala definiert. Sofern es sich um eine Skala einer Unterachse handelt und FontParentTitle true ist, wird die nächste übergeordnete Achse gesucht, bei der FontParentTitle false gesetzt ist. Weisen alle übergeordneten Achsen den Wert true auf, wird im GanttGraph mit der Suche des Fonts fortgefahrene.

Kontakt

FontParentTitle

Schriftart der Achse

Typ logic

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Durch das Setzen dieser Eigenschaft wird ein Font für den Titel der Achse definiert. Sofern es sich bei der Achse um eine Unterachse handelt und FontParentTitle true ist, wird die nächste übergeordnete Achse gesucht, bei der FontParentTitle false gesetzt ist. Weisen alle übergeordneten Achsen den Wert true auf, wird im GanttGraph mit der Suche des Fonts fortgefahrene.

FontScala

Schriftart der Skala

Typ font

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In dieser Eigenschaft wird die Schriftart der Skala eingetragen. Der Font besteht neben der Schriftart aus dem Schriftgrad und verschiedenen Attributen (Fett, Kursiv, ...).

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden. Die Funktionen benötigen eine Variable vom Typ font.

Kontakt

FontSize

Größe der Schriftart

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet(),

FontName

In dieser Eigenschaft wird die Schriftgröße eingetragen, in der innerhalb des Objekts der Inhalt dargestellt werden soll. Die Schriftart wird in der Eigenschaft FontName eingetragen.

Bei der Eigenschaft kann ein Parameter angegeben werden. Der Parameter bestimmt, welcher Textabschnitt von der neuen Textgröße betroffen ist. Folgende Parameter können angegeben werden:

- WinEditAll

Der gesamte Text wird in der neuen Größe angezeigt.

- WinEditMark

Nur der markierte Text wird in der neuen Größe angezeigt. Die Markierung wird über die Eigenschaft Range bestimmt.

- WinEditWord

Nur das Wort, in dem der Cursor steht, wird in der neuen Größe angezeigt. Die Position des Cursors wird über die Eigenschaft Range bestimmt.

Wird ein unformatierter Text in das RtfEdit-Objekt gelesen, kann mit diesen Optionen eine Formatierung des Textes vorgenommen werden.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Beispiele:

```
$RtfEdit->wpFontSize # 12; // 12 Punkt großer Font$RtfEdit->wpFontSize(_WinEditAll) # 12; // Gesam
```

FontSizeTwips

Zeichensatzgröße in Twips

Typ int

Liste, Objekte, FontSize,

Siehe WinPropGet(),

WinPropSet()

In dieser Eigenschaft kann die Fontgröße in Twips angegeben werden. 20 Twips entsprechen dabei einem Punkt. Wird der Font mit einer Größe von 200 Twips angegeben, entspricht das einer 10 Punkten großen Schrift.

Beim RtfEdit-Objekt kann bei der Eigenschaft ein Parameter angegeben werden. Der Parameter bestimmt, welcher Textabschnitt von der neuen Textgröße betroffen ist. Folgende Parameter können angegeben werden:

- WinEditAll

Der gesamte Text wird in der neuen Größe angezeigt.

- WinEditMark

Nur der markierte Text wird in der neuen Größe angezeigt. Die Markierung wird über die Eigenschaft Range bestimmt.

- WinEditWord

Nur das Wort, in dem der Cursor steht, wird in der neuen Größe angezeigt. Die Position des Cursors wird über die Eigenschaft Range bestimmt.

Wird ein unformatierter Text in das RtfEdit-Objekt gelesen, kann mit diesen Optionen eine Formatierung des Textes vorgenommen werden.



Beim PrtRtf-Objekt wirkt sich eine Änderung der Eigenschaft auf den gesamten RTF-Text aus. Standardmäßig steht hier die Eigenschaft auf -1. Bei diesem Wert, wird sie nicht ausgewertet.

Änderungen in dieser Eigenschaft ändert automatisch die Eigenschaft FontSize mit. Der Wert wird dabei kaufmännisch gerundet. Ebenso wird bei Änderungen der FontSize diese Eigenschaft geändert.

Beispiele:

```
$RtfEdit->wpFontSizeTwips # 240; // 12 Punkt großer Font$RtfEdit->wpFontSizeTwips (_WinEditAll) #
```

FontTitle

Schriftart des Titels

Typ font

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In dieser Eigenschaft wird die Schriftart des Titels eingetragen. Der Font besteht neben der Schriftart aus dem Schriftgrad und verschiedenen Attributen (Fett, Kursiv, ...).

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden. Die Funktionen benötigen eine Variable vom Typ font.

IndentLeft

linker Absatzeinzug

Typ int

Liste, Objekte,

IndentRight,

Siehe IndentOfs,

WinPropGet(),

WinPropSet()

In dieser Eigenschaft kann der linke Einzug einer Markierung (Absatzweise) oder des gesamten Dokumentes gesetzt werden.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden. Erfolgt das Setzen der Eigenschaft prozedural, kann die Option WinEditAll zum objektweiten oder WinEditMark zum markierungsweiten Setzen verwendet werden.

Beispiel:

```
$RtfEdit->wpIndentLeft(_WinEditAll) # PrtUnitLog(1.0, _PrtUnitCentimetres); $RtfEdit->wpIndentLe
```

IndentOfs

Einzug ab der zweiten Zeile

Typ int

Liste, Objekte, IndentLeft,

Siehe IndentRight,

WinPropGet(),

WinPropSet()

In dieser Eigenschaft kann der Einzug einer Markierung (Absatzweise) oder des gesamten Dokumentes gesetzt werden. Hier wird der Abstand ab der zweiten Zeile von dem linken Einzug angegeben. Dieser Einzug kann negativ sein.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Erfolgt das Setzen der Eigenschaft prozedural, kann die Option WinEditAll zum objektweiten oder WinEditMark zum markierungsweiten Setzen verwendet werden.

Beispiel:

```
$RtfEdit->wpIndentOfs(_WinEditAll) # PrtUnitLog(-1.0, _PrtUnitCentimetres); $RtfEdit->wpIndentOf
```

IndentRight

rechter Absatzeinzug

Typ int

Liste, Objekte,

IndentLeft,

Siehe IndentOfs,

WinPropGet(),

WinPropSet()

In dieser Eigenschaft kann der rechte Einzug einer Markierung (Absatzweise) oder des gesamten Dokumentes gesetzt werden.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Erfolgt das Setzen der Eigenschaft prozedural, kann die Option WinEditAll zum objektweiten oder WinEditMark zum markierungsweiten Setzen verwendet werden.

Beispiel:

```
$RtfEdit->wpIndentRight(_WinEditAll) # PrtUnitLog(1.0, _PrtUnitCentimetres); $RtfEdit->wpIndentR
```

Kontakt

LineSpacingTwips Höhe der

Zeile in Twips Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet(),

PrtUnitLog(), PrtUnit()

In dieser Eigenschaft wird der Zeilenabstand angegeben. Der Zeilenabstand kann entweder in Twips oder mit Hilfe der folgenden Konstanten angegeben werden:

Konstante	Wert	Zeilenabstand
<u>WinRtfLineSpacingSingle /</u> <u>PrtRtfLineSpacingSingle</u>	0	1-zeiliger Abstand
<u>WinRtfLineSpacingOneHalf /</u> <u>PrtRtfLineSpacingOneHalf</u>	-1	1 1/2-zeiliger Abstand
<u>WinRtfLineSpacingDouble /</u> <u>PrtRtfLineSpacingDouble</u>	-2	2-zeiliger Abstand
<u>PrtRtfLineSpacingCurrent</u>	-3	Zeilenabstand nicht verändern (nur PrtRtf)

Der Zeilenabstand kann über die Anweisungen PrtUnitLog() und PrtUnit() in Twips umgerechnet werden.

Kontakt

RtfAlign



Textausrichtung

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In dieser Eigenschaft wird die Ausrichtung des Textes eingetragen. Folgende Ausrichtungen stehen zur Verfügung:

WinRtfAlignLeft Linksbündiger Text

WinRtfAlignCenter Zentrierter Text

WinRtfAlignRight Rechtsbündiger Text

WinRtfAlignJustify Blocksatz

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

RtfEffect



Schriftattribute

Typ int

[Liste, Objekte,](#)

Siehe [WinPropGet\(\)](#),

[WinPropSet\(\)](#)

Diese Eigenschaft dient zur Vorbelegung des RtfEdit-Objekts. Es können folgende Schriftattribute gesetzt werden:

[WinRtfEffectBold](#) Schriftattribut Fett

[WinRtfEffectItalic](#) Schriftattribut Kursiv

[WinRtfEffectUnderline](#) Schriftattribut Unterstrichen

[WinRtfEffectStrikeout](#) Schriftattribut Durchgestrichen

[WinRtfEffectSubscript](#) Schriftattribut Tiefgestellt

[WinRtfEffectSuperscript](#) Schriftattribut Hochgestellt

[WinRtfEffectProtected](#) Text vor Editieren und Löschen schützen

[WinRtfEffectLink](#) Text wird als Link dargestellt

Die verschiedenen Effekte können kombiniert werden. Eine Änderung der Attribute kann über das Objekt ToolbarRtf erfolgen.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden. Erfolgt das Setzen der Eigenschaft prozedural, kann die Option WinEditAll zum objektweiten oder WinEditMark zum markierungsweiten Setzen verwendet werden.

Das zu setzende Attribut muss mit WinRtfEffectSet kombiniert werden, um die angegebenen Attribute zu setzen. Nicht angegebene Attribute bleiben unberührt. Um bestehende Attribute zurückzusetzen muss das Attribut mit der Konstanten WinRtfEffectClear kombiniert werden.

Beispiel:

```
$RtfEdit->wpRtfEffect(_WinEditAll) # _WinRtfEffectBold | _WinRtfEffectSet; $RtfEdit->wpRtfEffect
```

ThemeFontBox

Schriftart für Box-Objekte

Typ font

Siehe Liste, Objekte

Diese Eigenschaft definiert in einem Theme-Objekt die Schriftart für Box-Objekte in einem GanttGraph. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeGanttGraph

Beispiele:

```
tTheme->wpThemeFontBox(_WinTypeGanttGraph) # tFont;tFont # tTheme->wpThemeFontBox(_WinTypeGanttGr
```

ThemeFontCaption

Schriftart für Titelzeile

Typ [font](#)

Siehe [Liste, Objekte](#)

Diese Eigenschaft definiert in einem [Theme](#)-Objekt die Schriftart für die Titelzeile von [Axis](#)-Objekten in einem [GanttGraph](#). Das [Theme](#)-Objekt muss zuvor mit dem Befehl [WinThemeOpen\(\)](#) geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- [WinTypeGanttAxis](#)

Beispiele:

```
tTheme->wpThemeFontCaption(_WinTypeGanttAxis) # tFont;tFont # tTheme->wpThemeFontCaption(_WinType
```

ThemeFontIvl

Schriftart für Intervall-Objekte

Typ font

Siehe Liste, Objekte

Diese Eigenschaft definiert in einem Theme-Objekt die Schriftart für Interval-Objekte in einem GanttGraph. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeGanttGraph

Beispiele:

```
tTheme->wpThemeFontIvl(_WinTypeGanttGraph) # tFont;tFont # tTheme->wpThemeFontIvl(_WinTypeGanttGr
```

ThemeFontText

Schriftart

Typ font
Liste,

Siehe Objekte

Diese Eigenschaft definiert in einem Theme-Objekt die Schriftart. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeAppFrame
- WinTypeBarcode
- WinTypeBigIntEdit
- WinTypeButton
- WinTypeCheckBox
- WinTypeCodeEdit
- WinTypeCodeEditListColumn
- WinTypeColorEdit
- WinTypeDataListColumn
- WinTypeDataListPopupColumn
- WinTypeDateEdit
- WinTypeDecimalEdit
- WinTypeEdit
- WinTypeFloatEdit
- WinTypeFontNameEdit
- WinTypeFontSizeEdit
- WinTypeFrame
- WinTypeGanttAxis
- WinTypeGroupBox
- WinTypeGroupSplit
- WinTypeGroupTile
- WinTypeHelpTip
- WinTypeHyperLink
- WinTypeIntEdit
- WinTypeLabel
- WinTypeMdiFrame
- WinTypeMenuButton
- WinTypeMenuItem
- WinTypeMetaPicture
- WinTypeNotebook
- WinTypePicture
- WinTypeProgress
- WinTypePrtPreviewDlg
- WinTypeRadioButton
- WinTypeRecListColumn
- WinTypeRecListPopupColumn
- WinTypeSlider
- WinTypeStatusbarButton
- WinTypeStoListColumn

Kontakt

- WinTypeStoListPopupColumn
- WinTypeTextEdit
- WinTypeTimeEdit
- WinTypeToolbarButton
- WinTypeTreeNode

Beispiele:

```
tTheme->wpThemeFontText(_WinTypeButton) # tFont;tFont # tTheme->wpThemeFontText(_WinTypeMenuButto
```

ThemeFontTextHdr

Schriftart für Spaltenköpfe

Typ [font](#)

Siehe [Liste](#), [Objekte](#)

Diese Eigenschaft definiert in einem [Theme](#)-Objekt die Schriftart für die Spaltenköpfe.

Das [Theme](#)-Objekt muss zuvor mit dem Befehl [WinThemeOpen\(\)](#) geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- [WinTypeDataList](#)
- [WinTypeDataListPopup](#)
- [WinTypeRecList](#)
- [WinTypeRecListPopup](#)
- [WinTypeStoList](#)
- [WinTypeStoListPopup](#)

Beispiele:

```
tTheme->wpThemeFontTextHdr(_WinTypeDataList) # tFont;tFont # tTheme->wpThemeFontTextHdr(_WinTypeD
```

Kontakt

Objekte von ThemeFontTextHdr

Folgende Objekte unterstützen die Eigenschaft ThemeFontTextHdr

Siehe [ThemeFontTextHdr](#),

[Liste der Objekte](#),

[Liste der](#)

[Eigenschaften](#)

- [Theme](#)
- [ThemeSet](#)

Kontakt

Icon (Symbol-Eigenschaften)

Eigenschaften

sortiert nach

Gruppen,

Siehe Alphabetische

Liste aller

Eigenschaften

HdrIcon

Symbol des Spaltenkopfes

HdrImageOption

Darstellungsoption für das Bild des Spaltenkopfes

HdrTileSize

Größe der Bilder

HdrImageTile

Verwendetes Bild des Spaltenkopfes

HdrImageTileUser

Nummer der Kachel einer Grafik

ImageOption

Darstellungsoption für das Bild

ImageTile

Verwendetes Bild

ImageTileUser

Nummer der Kachel einer Grafik

Icon

Symbol der Anwendung

NumIcon

Nummer der Icon-Grafik

ThemeMenuItemSize

Bildgröße für Kachelgrafiken in Menüs

ThemeMenuCntxtTileSize

Bildgröße für Kachelgrafiken in Kontextmenüs

TileButtonSpacing

Höhe der Toolbar-Schaltflächen mit benutzerdefinierten

Kacheln

Name der Grafik (Spaltenkopf)

TileNameMenu

Name der Menü-Grafik

TileNameNormal

Name der Button-Grafik

TileNamePressed

Name der Button-Grafik (gedrückte Darstellung)

TileNameSelected

Name der Button-Grafik (ausgewählte Darstellung)

TileNameUser

Name der Grafik

TileSize

Größe der ImageTile und NodeStyle

TypeIcon

Typ des Symbols

Kontakt

HdrColFg



Vordergrundfarbe des Spaltenkopfes

Typ **int**

[Liste](#), [Objekte](#), [WinPropGet\(\)](#),

Siehe [WinPropSet\(\)](#), [HdrShadeCol1](#),
[HdrShadeCol2](#), [HdrShadeMode](#)

Diese Eigenschaft bestimmt die Textfarbe des Spaltenkopfes sowie die Farbe des Trenners zwischen den einzelnen Spalten. Standardmäßig steht dieser Wert aus Kompatibilitätsgründen auf [WinColParent](#).

HdrIcon

Symbol des Spaltenkopfes

Typ alpha

Liste, Objekte,

Siehe WinPropGet(),
WinPropSet()

In dieser Eigenschaft kann der Name des Symbols für das Objekt angegeben werden.

Das Symbol muss im ICO-Format vorliegen.

 Bei den Objekten Column und GroupColumn muss das Symbol in der Datenbank vorliegen (siehe Ressource importieren). Bei diesen Objekten kann die Größe des verwendeten Bildes über die Eigenschaft HdrTileSize definiert werden.

 Damit das Bild angezeigt wird, darf die Eigenschaft HdrImageOption der Liste nicht auf WinImgIgnore stehen.

Beispiele:

```
$clmArticle->wpHdrIcon # 'Article.ico' // Datei aus der Datenbank
```

Kontakt

HdrImageOption



Darstellungsoption für das Bild des Spaltenkopfes

Typ int

Liste, Objekte, WinPropGet(), WinPropSet(),

Siehe HdrTileSize, HdrIcon, HdrImageTile,

HdrImageTileUser

Diese Eigenschaft gibt an wie das Bild, das in HdrImageTile, HdrImageTileUser oder HdrIcon angegeben wurde, darzustellen ist.

Folgende Darstellungen gibt es:

WinImgIgnore

- Das Bild wird nicht dargestellt

WinImgTextBelow

- Der Text erscheint unterhalb des Bildes

WinImgTextIgnore

- Der Text wird nicht dargestellt

WinImgTextLeft

- Der Text erscheint links vom Bild

WinImgTextRight

- Der Text erscheint rechts vom Bild

WinImgLeftTextRight - Das Bild erscheint links, der Text wird nach der Eigenschaft JustifyView

angezeigt.

HdrImageTile

Verwendetes Bild des Spaltenkopfes

Typ int
Siehe Liste, Objekte, WinPropGet(),

In dieser Eigenschaft wird das Bild angegeben, das vom Objekt zu verwenden ist. Ist bei dem Objekt zusätzlich die Eigenschaft HdrIcon angegeben, wird die Eigenschaft HdrIcon bevorzugt.



Damit das Bild angezeigt wird, darf die Eigenschaft HdrImageOption der Liste nicht auf WinImgIgnore stehen.

Die Bilder sind in unterschiedlichen Varianten und Abmessungen verfügbar. Die verwendete Variante wird über die Eigenschaft TileTheme und die Abmessung mit HdrTileSize definiert.



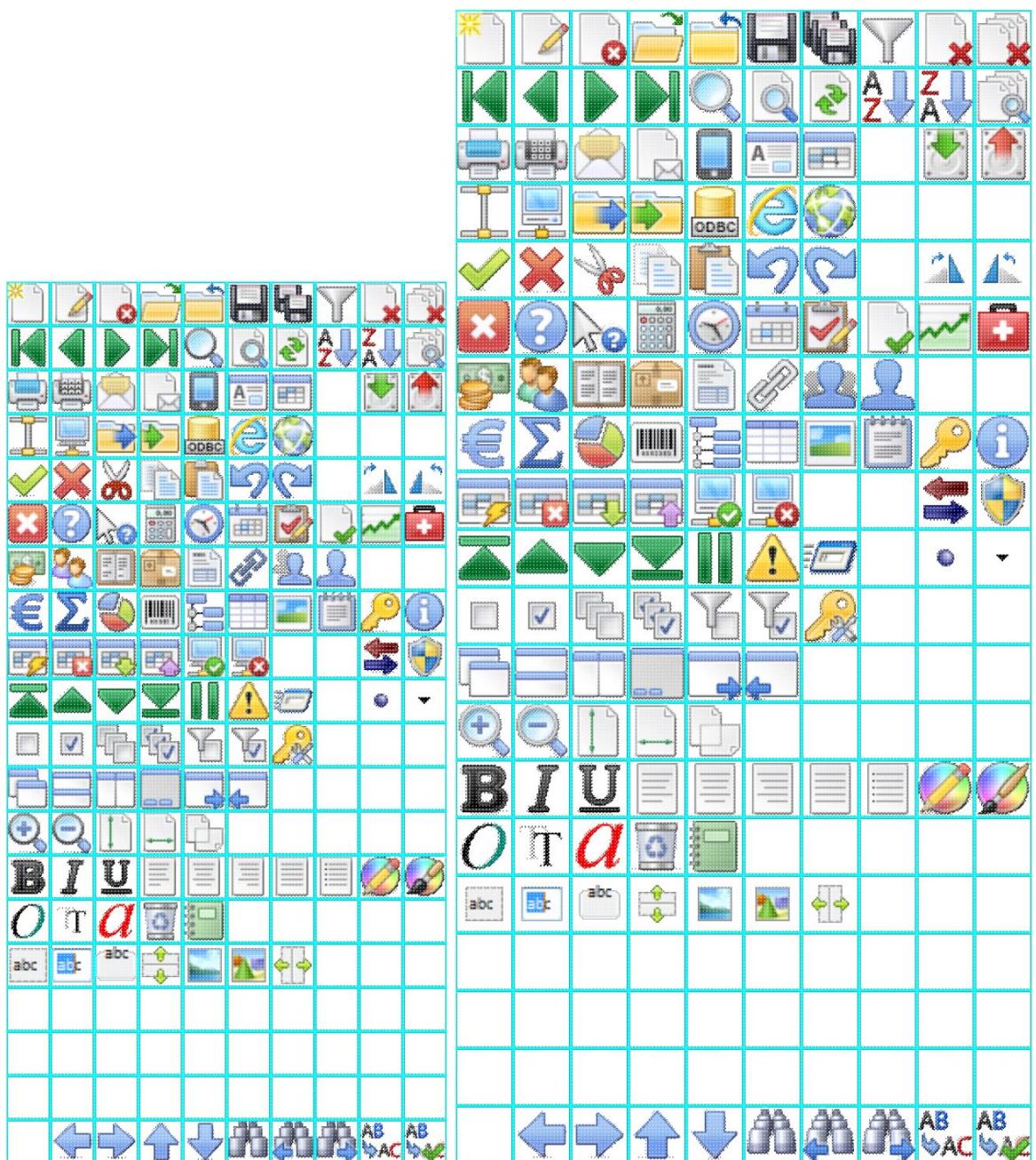
Bilder der Variante WinTileThemeFlatMono entsprechen der Variante

WinTileThemeFlatColor mit dem Unterschied, dass nur monochrome Farben verwendet werden. Daher ist in der folgenden Liste nur ein Beispiel für die Größe 48x48 Pixel enthalten. Die Bilder werden zur Laufzeit an die Schriftfarbe der jeweiligen Objektes im ausgewählten Theme angepasst.

Die Eigenschaft HdrImageTile kann folgende Werte annehmen:



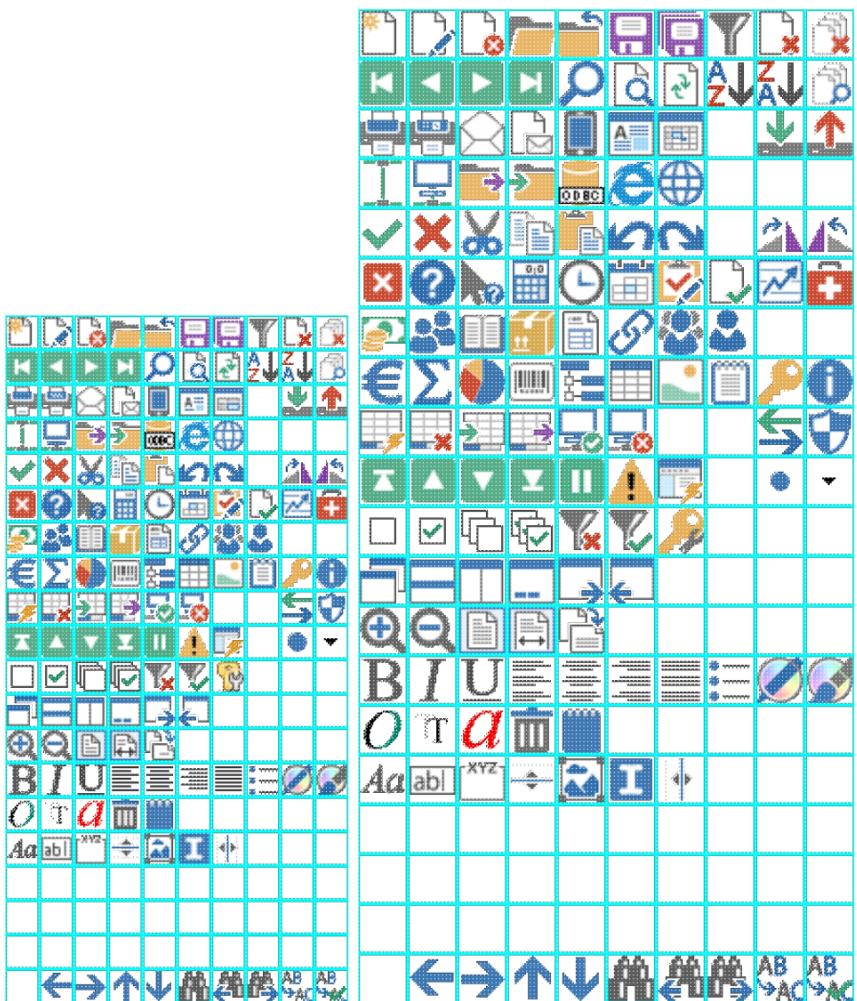
Kontakt



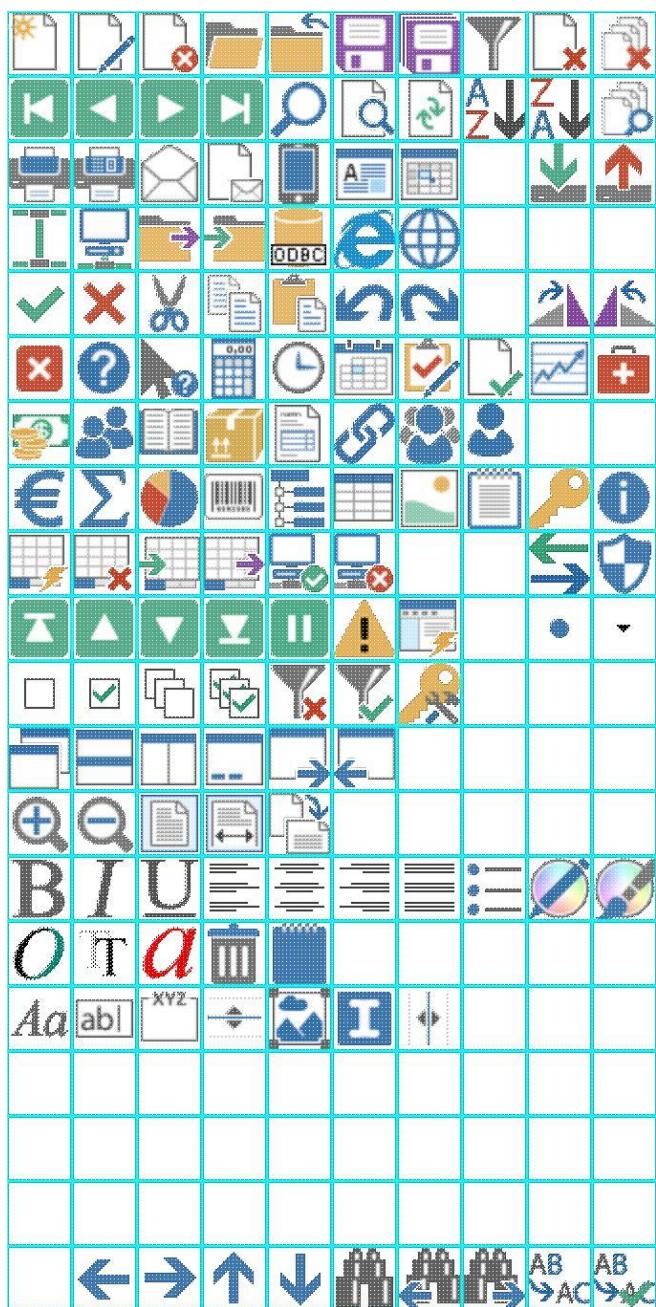
Kontakt



Kontakt



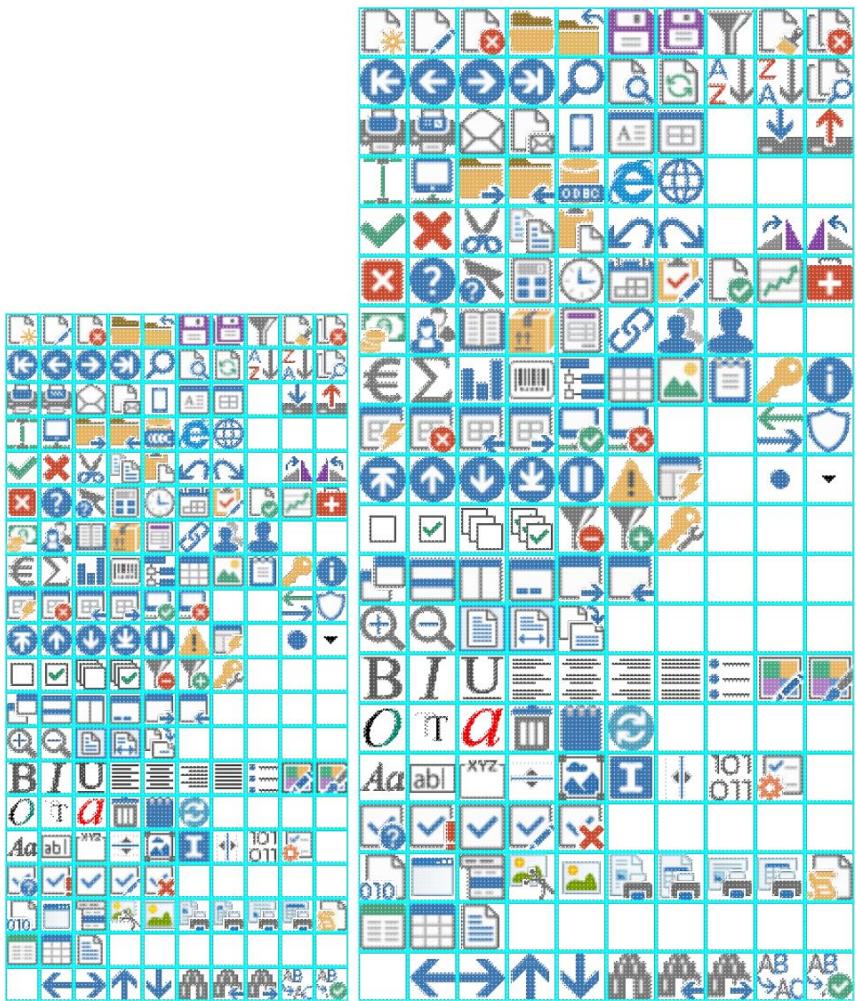
Kontakt



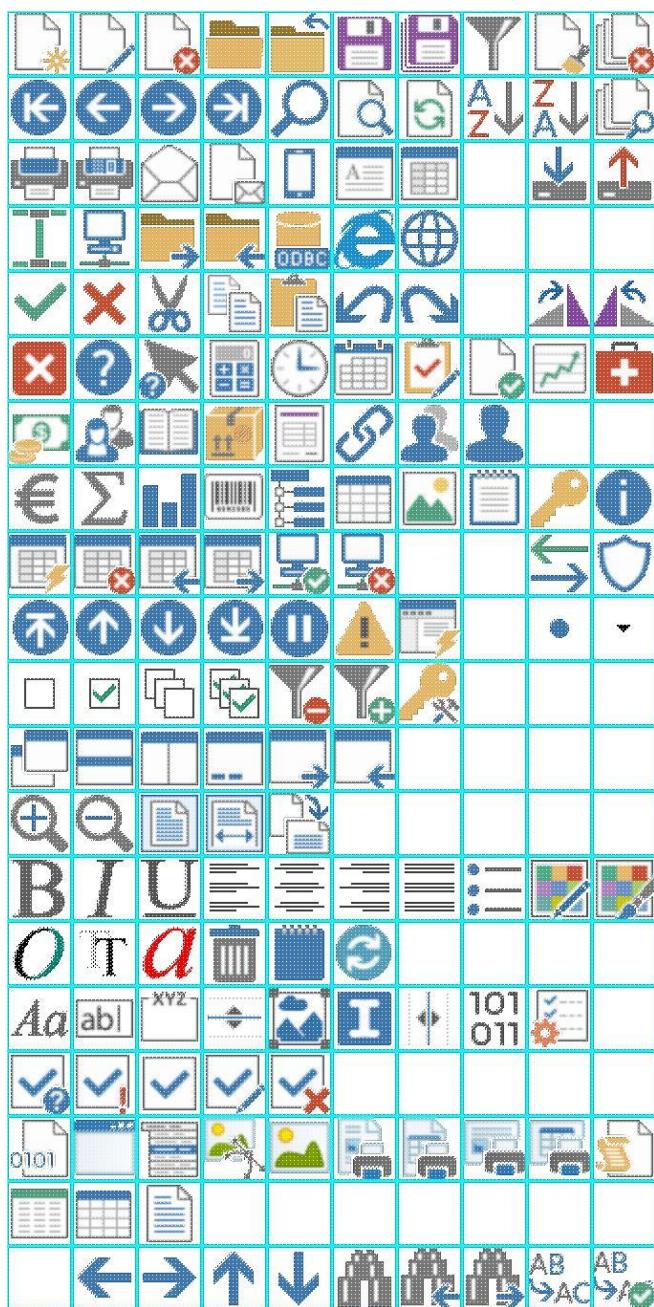
Kontakt



Kontakt



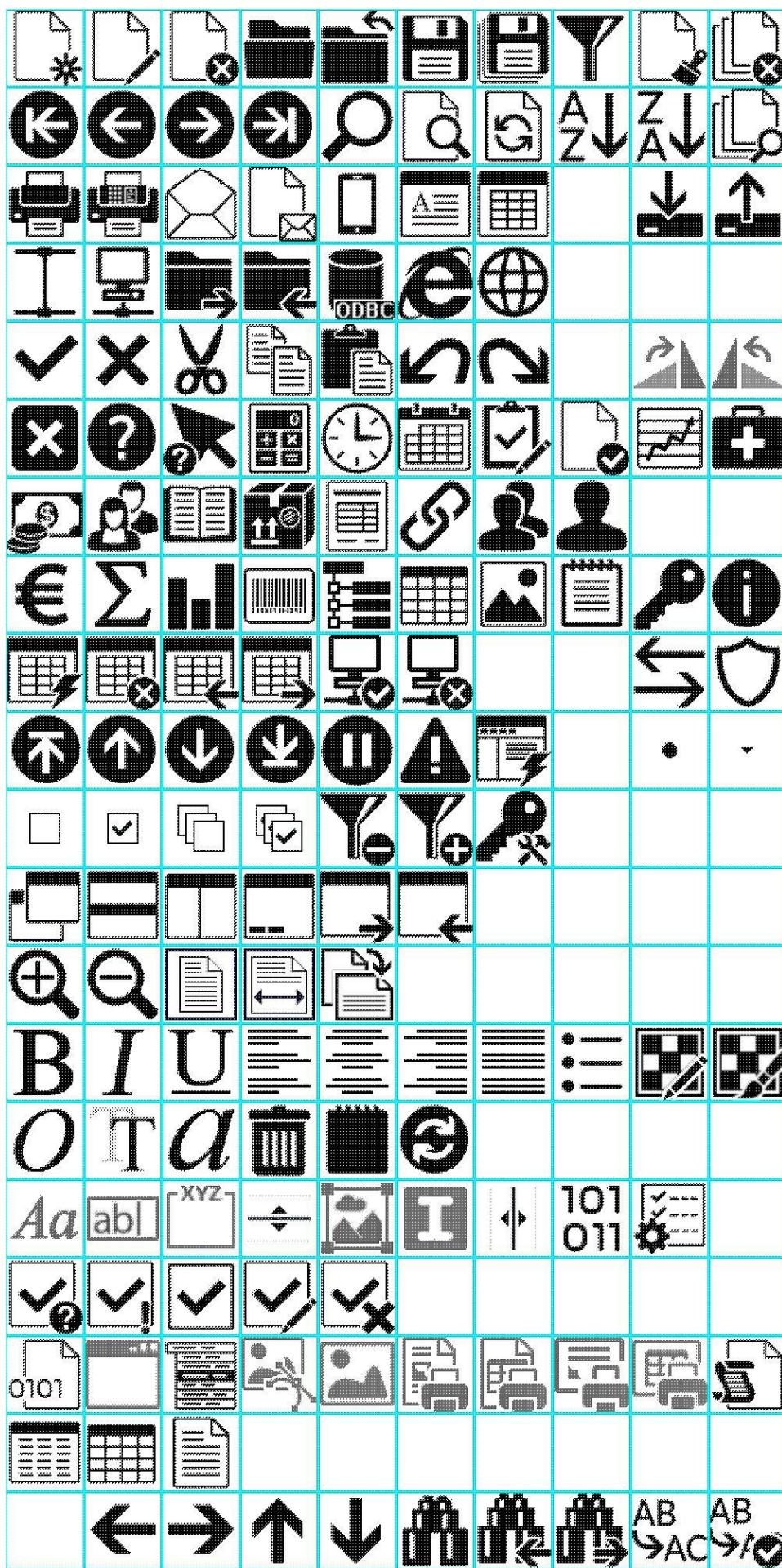
Kontakt



Kontakt



Kontakt



Kontakt

Konstante (Wert)	Classic	Enhanced	Modern	Flat
		16x16 24x24 32x32 48x48	16x16 24x24 32x32 48x48	1
<code>_WinImgNone (0)</code>				
<code>_WinImgAddress (63)</code>				
<code>_WinImgAlignCenter (135)</code>				
<code>_WinImgAlignJustify (137)</code>				
<code>_WinImgAlignLeft (134)</code>				
<code>_WinImgAlignRight (136)</code>				
<code>_WinImgArticle (64)</code>				
<code>_WinImgArrowLeft (192)</code>				
<code>_WinImgArrowRight (193)</code>				
<code>_WinImgArrowUp (194)</code>				
<code>_WinImgArrowDown (195)</code>				
<code>_WinImgBarcode (74)</code>				
<code>_WinImgBin (144)</code>				
<code>_WinImgBold (131)</code>				
<code>_WinImgBullets (138)</code>				
<code>_WinImgCalc (54)</code>				
<code>_WinImgCalendar (56)</code>				
<code>_WinImgCancel (42)</code>				
<code>_WinImgClear (9)</code>				
<code>_WinImgClock (55)</code>				
<code>_WinImgClose (5)</code>				
<code>_WinImgColorBkg (140)</code>				
<code>_WinImgColorFg (139)</code>				
<code>_WinImgComboBox (100)</code>				
<code>_WinImgConfig (58)</code>				
<code>_WinImgConnect (85)</code>				
<code>_WinImgCopy (44)</code>				
<code>_WinImgCut (43)</code>				
<code>_WinImgDataComm (31)</code>				
<code>_WinImgDblFind (20)</code>				
<code>_WinImgDblRemove (10)</code>				
<code>_WinImgDelete (3)</code>				
<code>_WinImgDiagnosis (60)</code>				
<code>_WinImgDisconnect (86)</code>				
<code>_WinImgEdit (2)</code>				
<code>_WinImgEditor (78)</code>				
<code>_WinImgEmail (24)</code>				
<code>_WinImgEuro (71)</code>				

_WinImgExcel (27)
_WinImgExcelClose (82)

_WinImgExcelExport (84)

_WinImgExcelImport (83)

_WinImgExcelOpen (81)
_WinImgExchange (89)
_WinImgExit (51)
_WinImgExport (33)
_WinImgFax (22)
_WinImgFilter (8)
_WinImgFind (15)
_WinImgFirst (11)
_WinImgHelp (52)
_WinImgHelpTip (53)
_WinImgIExplorer (36)
_WinImgImport (34)
_WinImgInfoSymbol (80)
_WinImgInstall (29)
_WinImgInternet (37)
_WinImgItalic (132)

_WinImgKeyRepair (107)

_WinImgLast (14)
_WinImgLink (66)
_WinImgMail (23)
_WinImgMarkAllClear
(103)
_WinImgMarkAllSet (104)
_WinImgMarkFilterClear
(105)
_WinImgMarkFilterSet
(106)
_WinImgMarkOneClear
(101)
_WinImgMarkOneSet (102)

_WinImgMdiArrange (114)

_WinImgMdiCascade (111)

_WinImgMdiNext (115)

_WinImgMdiPrev (116)

_WinImgMdiTileHorz (112)

_WinImgMdiTileVert (113)

_WinImgMoney (61)

_WinImgNetwork (32)

_WinImgNew (1)

_WinImgNext (13)

_WinImgObjDivider (154)

_WinImgObjDividerUp

(157)

_WinImgObjEdit (152)

_WinImgObjGroup (153)

_WinImgObjIcon (156)

_WinImgObjLabel (151)

_WinImgObjPicture (155)

_WinImgODBC (35)

_WinImgOk (41)

_WinImgOpen (4)

_WinImgOptimize (59)

_WinImgOrder (65)

_WinImgPageFirst (91)

_WinImgPageFlip (125)

_WinImgPageLast (94)

_WinImgPageNext (93)

_WinImgPagePrev (92)

_WinImgPartner (62)

_WinImgPassword (79)

_WinImgPaste (45)

_WinImgPause (95)

_WinImgPhone (25)

_WinImgPicture (77)

_WinImgPrev (12)

_WinImgPreview (16)

_WinImgPrint (21)
_WinImgRadioCheck (99)
_WinImgRedo (47)
_WinImgRefresh (17)
_WinImgReplace (199)
_WinImgReplaceAll (200)
_WinImgReport (145)
_WinImgRotateL (50)
_WinImgRotateR (49)
_WinImgSave (6)
_WinImgSaveAll (7)
_WinImgSearch (196)
_WinImgSearchNext (197)

_WinImgSearchPrev (198)

_WinImgSettings (57)
_WinImgShield (90)
_WinImgSortDown (19)

_WinImgSortUp (18)

_WinImgStart (97)
_WinImgStatistics (73)
_WinImgTable (76)
_WinImgTotal (72)
_WinImgTree (75)
_WinImgUnderline (133)
_WinImgUndo (46)
_WinImgUninstall (30)
_WinImgUser (68)
_WinImgUserGroup (67)
_WinImgWarning (96)
_WinImgWord (26)

_WinImgZoomFitPage (123)

**_WinImgZoomFitPageWidth
(124)**
_WinImgZoomMinus (122)
_WinImgZoomPlus (121)

HdrImageTileUser

Nummer der Kachel einer Grafik

Typ int Liste, Objekte, WinPropGet(),
Siehe

Hat diese Eigenschaft einen Wert größer 0, wird sie an Stelle der Eigenschaft HdrImageTile verwendet, um eine Kachel aus einer importierten Listen-Grafik zu referenzieren.

Die zu verwendende Grafik wird aus der Eigenschaft TileNameHeader (Eigenschaft der Listen-Objekte) selektiert.



Damit das Bild angezeigt wird, darf die Eigenschaft HdrImageOption der Liste nicht auf WinImgIgnore stehen.

Die Eigenschaft kann im Eigenschaftsfenster des jeweiligen Objekt-Editors oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

HdrShadeCol1

Füllfarbe des Spaltenkopfes

Typ int

Liste, Objekte,
WinPropGet(),

Siehe WinPropSet(),

HdrColFg,
HdrShadeCol2,
HdrShadeMode

Diese Eigenschaft bestimmt die Füllfarbe des Spaltenkopfes. Standardmäßig hat die Eigenschaft aus Kompatibilitätsgründen den Wert WinColParent. In Verbindung mit den Eigenschaften HdrShadeCol2 und HdrShadeMode können auch Farbverläufe dargestellt werden.

Kontakt

HdrShadeCol2



Farbe des Spaltenkopfes für Farbverlauf

Typ int

Liste, Objekte, WinPropGet(),

Siehe WinPropSet(), HdrColFg,

HdrShadeCol1, HdrShadeMode

Die Eigenschaft ist die zweite Farbe für die Darstellung eines Farbverlaufs in Zusammenhang mit den Eigenschaften HdrShadeCol1 und HdrShadeMode. Hat die Eigenschaft den Wert WinColParent, wird die Farbe WinColBtnFace verwendet.

Kontakt

HdrShadeMode



Modus des Farbverlaufs des Spaltenkopfes

Typ int

Liste, Objekte, WinPropGet(),

Siehe WinPropSet(), HdrColFg,

HdrShadeCol1, HdrShadeCol2

Diese Eigenschaft bestimmt den Modus des Farbverlaufs des Spaltenkopfes. Sie kann folgende Werte annehmen:

WinShadeSolid Der Spaltenkopf wird mit der Farbe HdrShadeCol1 gefüllt.

WinShadeTopBottom Der Farbverlauf wird von oben (HdrShadeCol1) nach unten (HdrShadeCol2) gezeichnet.

WinShadeBottomTop Der Farbverlauf wird von unten nach oben gezeichnet.

WinShadeLeftRight Der Farbverlauf wird von links nach rechts gezeichnet.

WinShadeRightLeft Der Farbverlauf wird von rechts nach links gezeichnet.

Kontakt

HdrTileSize



Größe der Bilder von HdrImageTile und HdrIcon Typ int

Siehe Liste, Objekte, WinPropGet(), WinPropSet()

Über diese Eigenschaft wird die Größe der Bilder von HdrImageTile und HdrIcon gesteuert.

Es können Werte zwischen 1 und 256 Pixel angegeben werden. Folgende Werte sind empfohlen:

16

Die Bilder werden in der Auflösung 16 x 16 Pixel dargestellt.

24

Die Bilder werden in der Auflösung 24 x 24 Pixel dargestellt.

32

Die Bilder werden in der Auflösung 32 x 32 Pixel dargestellt.

48

Die Bilder werden in der Auflösung 48 x 48 Pixel dargestellt.

WinTileSizeDefault Die Bilder werden in der Standardgröße (16 x 16 Pixel) dargestellt.

WinTileSizeApp Die Bilder werden in der definierten Größe des App-Objektes für den jeweiligen Objekttyp dargestellt (App->wpTileSize).

Ist das Bild nicht in der angegebene Größe vorhanden, dann wird das Bild mit der kleinsten abweichenden Größe verwendet. Bei mehreren Bildern mit derselben Größe (nur HdrIcon), wird jenes mit der kleinsten Abweichung bezügl. der Farbtiefe des Bildschirms verwendet.

Optional kann beim App-Objekt der betreffende Objekttyp als Parameter der Eigenschaft TileSize übergeben werden um alle Objekte dieses Typs zu beeinflussen, bei denen die Eigenschaft HdrTileSize auf WinTileSizeApp steht. Wird kein Parameter angegeben, bezieht sich die Größe auf alle Objekte. Folgende Objekttypen können angegeben werden:

- WinTypeListColumn
- WinTypeGroupColumn



Ist die Eigenschaft TileTheme des App-Objektes auf WinTileThemeClassic gesetzt, wird die Eigenschaft TileSize nicht ausgewertet.

Beispiele:

```
// Applikations-Update deaktivieren_App->wpAutoUpdate # false;// Größe der Standard Image-Tiles a
```

Kontakt

Icon

Symbol der Anwendung

Typ alpha

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet(), Blog

In dieser Eigenschaft kann der Name des Symbols für das Objekt angegeben werden.

Das Symbol muss im ICO-Format vorliegen.

Das Symbol erscheint bei Fensterobjekten oben links in der Titelleiste und in der Taskleiste.



Bei den Objekten Button, ColorButton, DatePicker, DecimalEdit, FloatEdit, GroupTile-Button, MenuButton, Statusbar-Button, Toolbar-Button, HyperLink, MenuItem und NotebookPage muss das Symbol in der Datenbank vorliegen (siehe Ressource importieren). Bei diesen Objekten (außer MenuItem und GroupTile-Button) kann die Größe des verwendeten Bildes über die Eigenschaft TileSize definiert werden.

Die Angabe der externen Datei kann mit absolutem oder relativem Pfad erfolgen. Bei der Angabe eines absoluten Pfades wird ein '*', bei der eines relativen Pfades (relativ zum Startverzeichnis des CONZEPT 16-Clients) wird ein '!' vorangestellt.

Bei Angabe des Namens ohne Präfix ('!' und '*') wird eine Icon-Datei in der Datenbank referenziert.

Beispiele:

```
'!Icons\Artikel.ico'           // externe Datei relativ angegeben'*C:\c16\Icons\Artikel.ico' // exte
```

Kontakt

ImageOption



Darstellungsoption für das Bild

Typ int

Liste, Objekte,

Siehe WinPropGet(),
WinPropSet()

Diese Eigenschaft gibt an wie das Bild, das in ImageTile angegeben wurde, darzustellen ist.

Folgende Darstellungen gibt es:

WinImgIgnore

- Das Bild wird nicht dargestellt

WinImgTextBelow

- Der Text erscheint unterhalb des Bildes

WinImgTextIgnore

- Der Text wird nicht dargestellt

WinImgTextLeft

- Der Text erscheint links vom Bild

WinImgTextRight

- Der Text erscheint rechts vom Bild

WinImgLeftTextRight - Das Bild erscheint links, der Text wird nach der Eigenschaft JustifyView angezeigt.



Beim Notebook-Objekt kann die Ausprägung WinImgLeftTextRight nicht verwendet werden.



Beim Toolbar-Button-Objekt wirkt sich die Ausprägung WinImgLeftTextRight erst aus, wenn die Eigenschaft Size (nicht 0) der übergeordneten Toolbar gesetzt wird.

ImageTile

Verwendetes Bild

Typ int

Liste, Objekte,

Siehe WinPropGet(),
WinPropSet()

In dieser Eigenschaft wird das Bild angegeben, das vom Objekt zu verwenden ist. Ist bei dem Objekt zusätzlich die Eigenschaft Icon angegeben, wird die Eigenschaft Icon bevorzugt.



Damit das Bild angezeigt wird, darf die Eigenschaft ImageOption nicht auf WinImgIgnore stehen.

Die Bilder sind in unterschiedlichen Varianten und Abmessungen verfügbar. Die verwendete Variante wird über die Eigenschaft TileTheme und die Abmessung mit TileSize definiert.

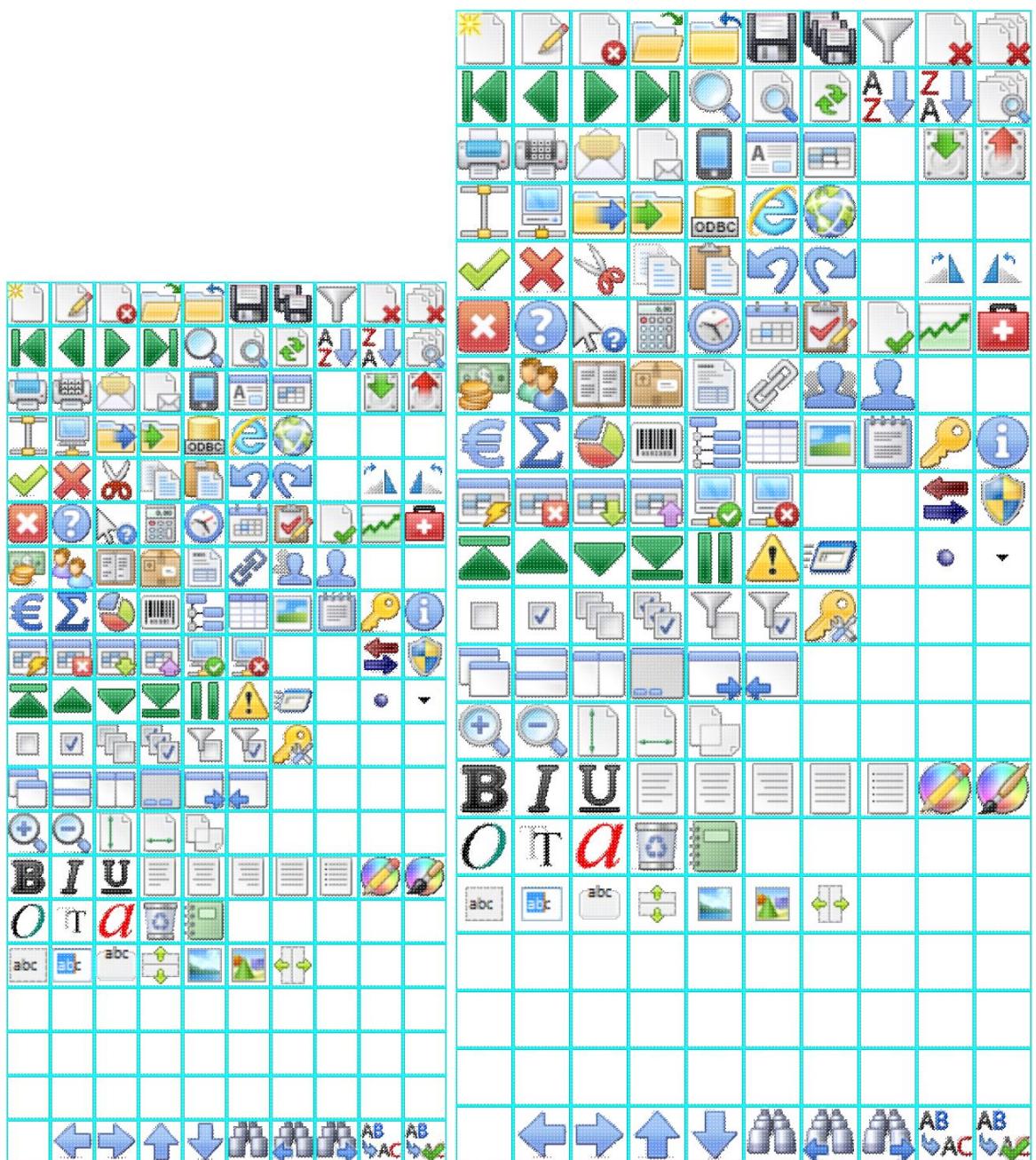


Bilder der Variante WinTileThemeFlatMono entsprechen der Variante WinTileThemeFlatColor mit dem Unterschied, dass nur monochrome Farben verwendet werden. Daher ist in der folgenden Liste nur ein Beispiel für die Größe 48x48 Pixel enthalten. Die Bilder werden zur Laufzeit an die Schriftfarbe der jeweiligen Objektes im ausgewählten Theme angepasst.

Die Eigenschaft ImageTile kann folgende Werte annehmen:



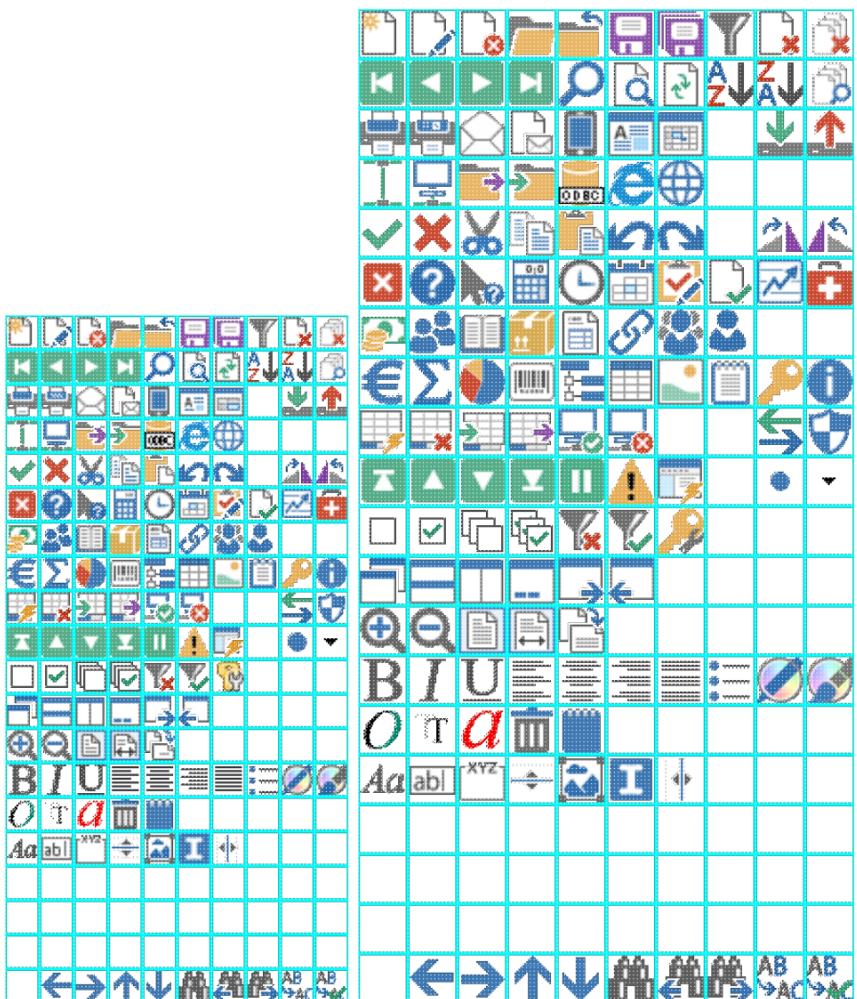
Kontakt



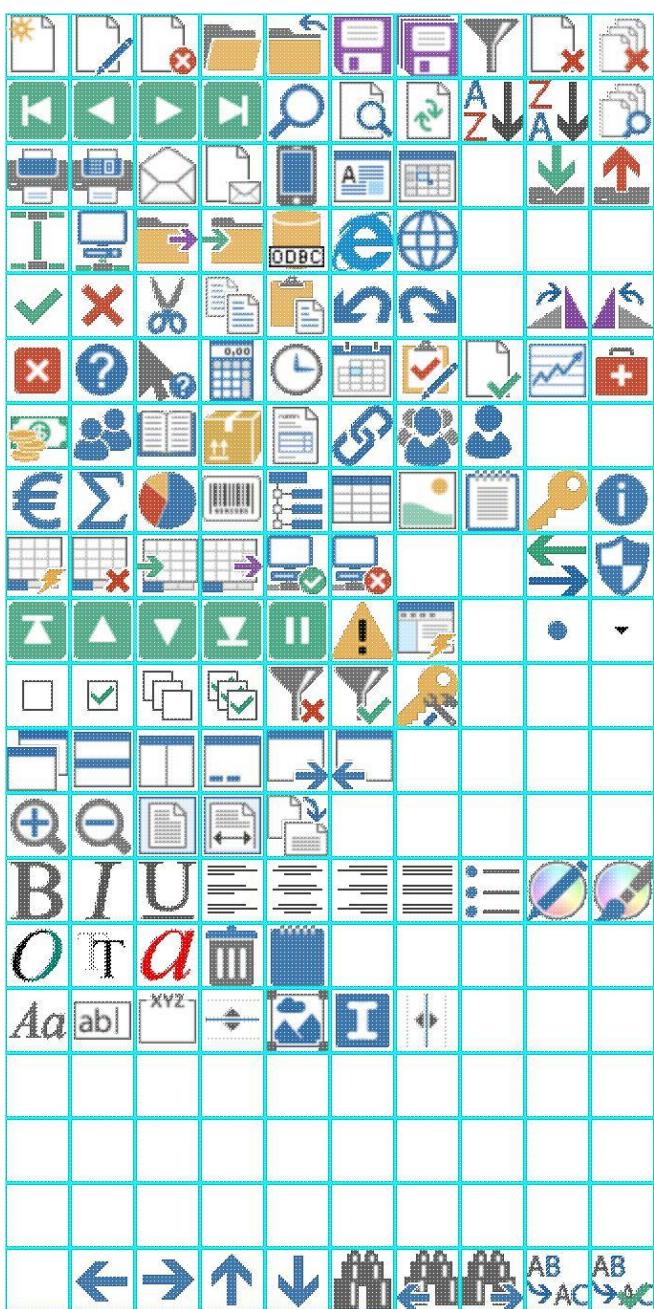
Kontakt



Kontakt



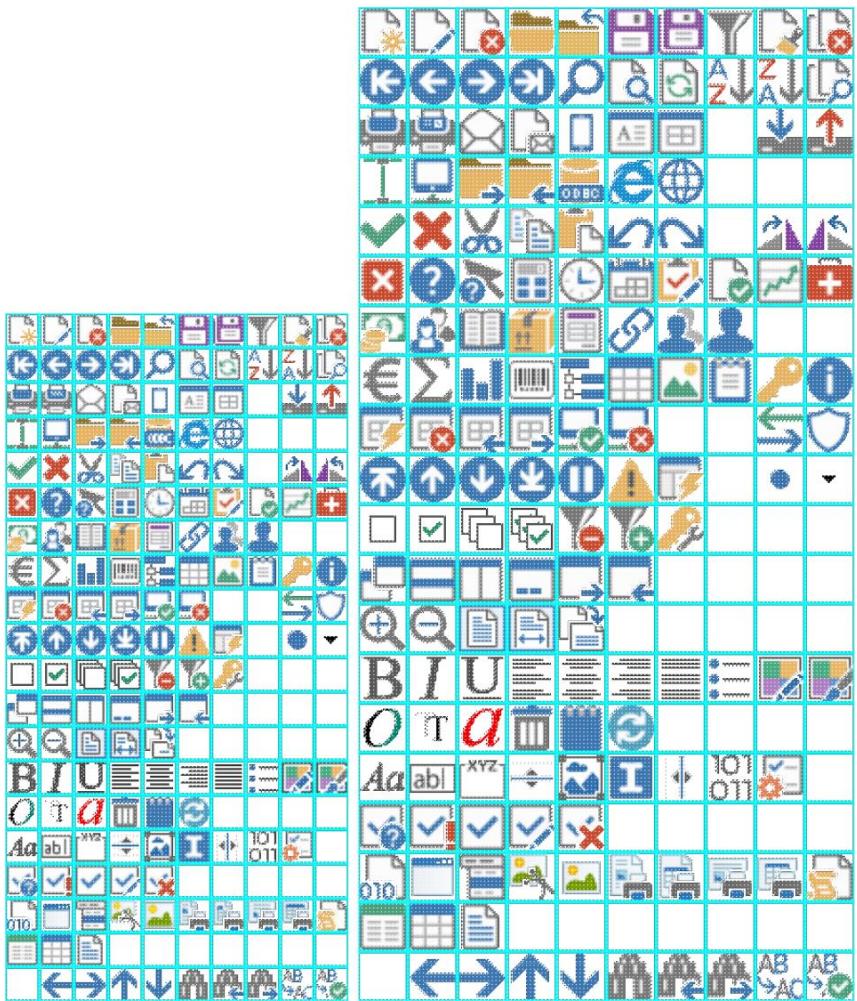
Kontakt



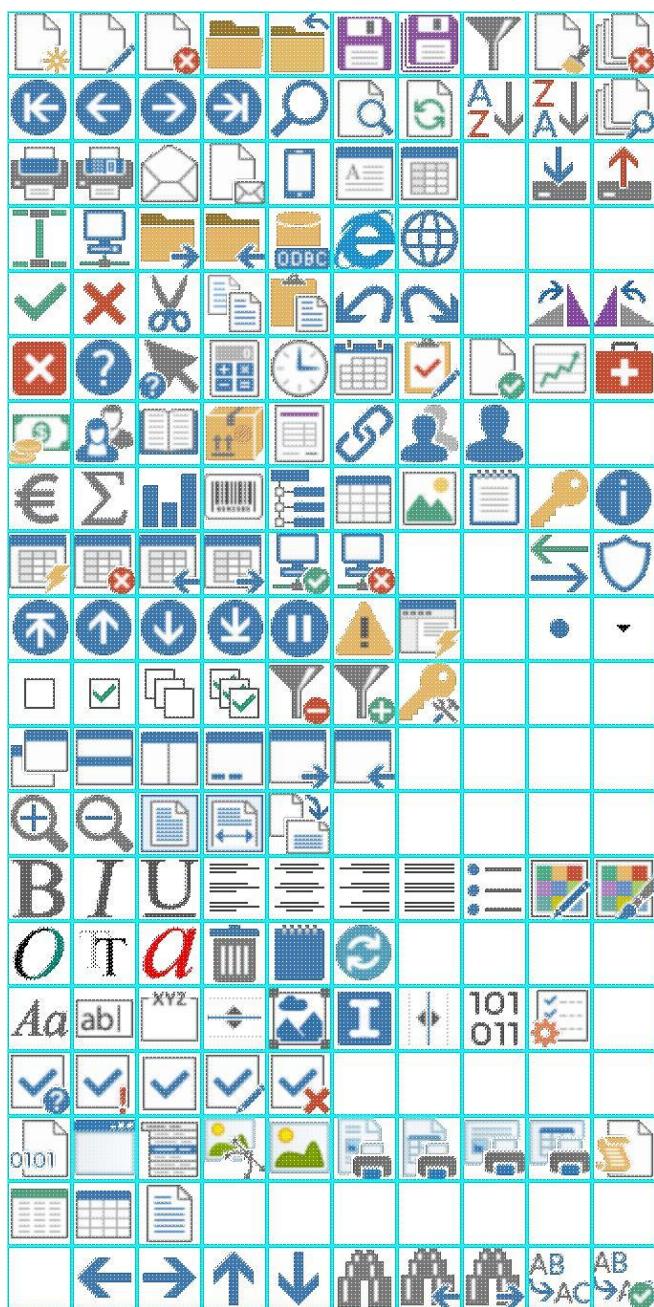
Kontakt



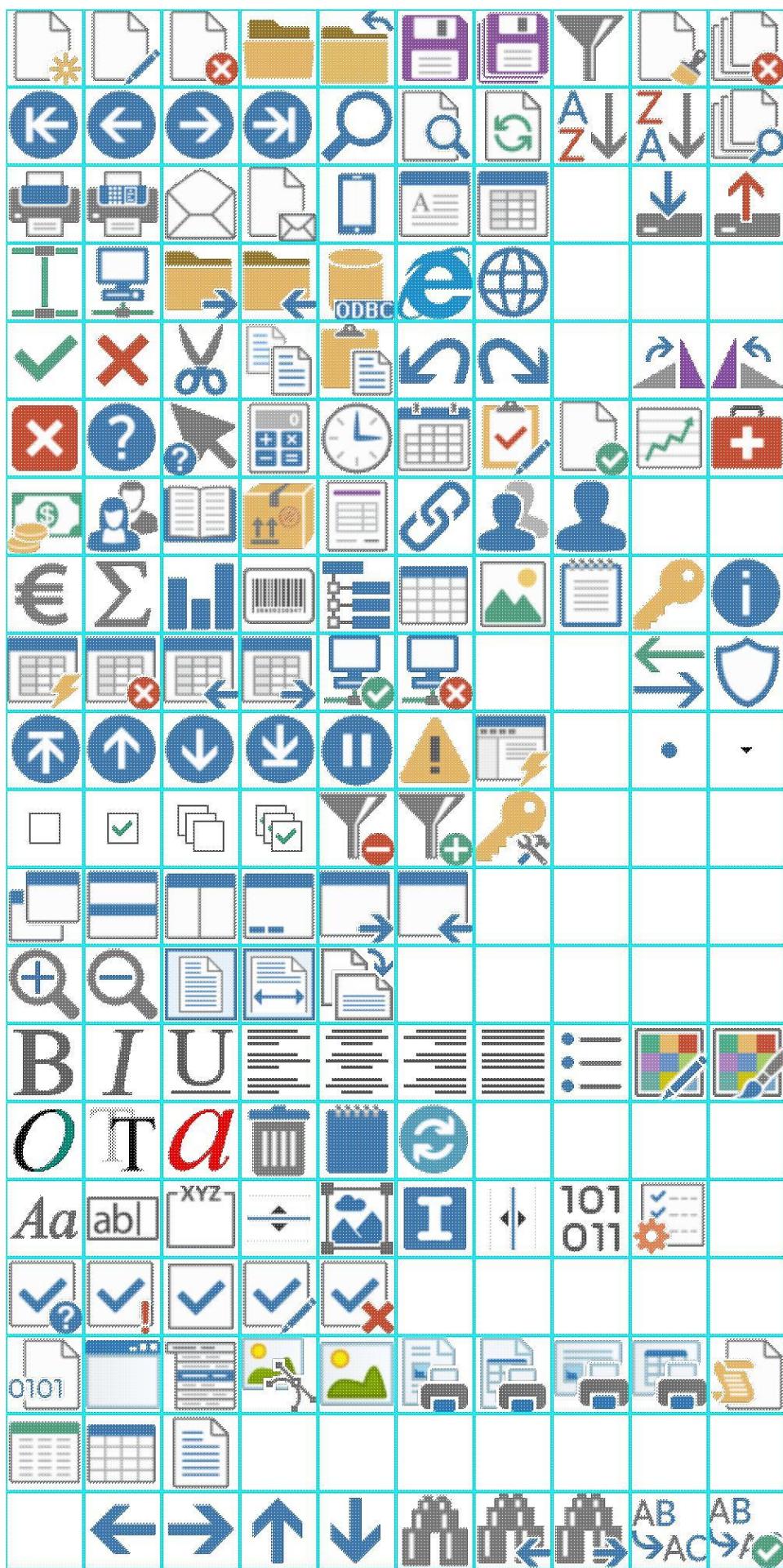
Kontakt



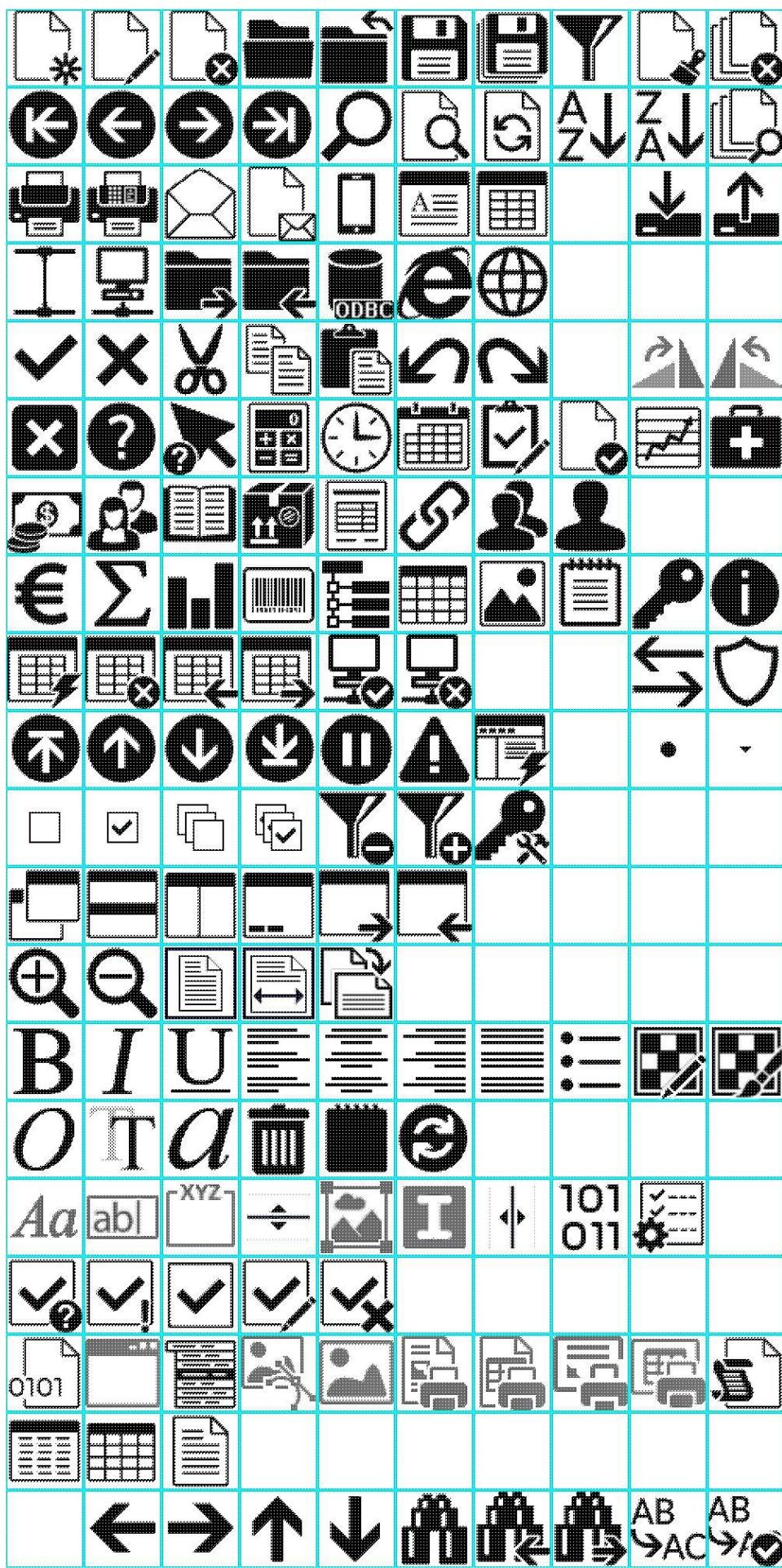
Kontakt



Kontakt



Kontakt



Kontakt

Konstante (Wert)

Classic Enhanced

Modern

_WinImgNone (0)
_WinImgAddress (63)
_WinImgAlignCenter (135)
_WinImgAlignJustify (137)
_WinImgAlignLeft (134)
_WinImgAlignRight (136)
_WinImgArticle (64)
_WinImgArrowLeft (192)
_WinImgArrowRight (193)
_WinImgArrowUp (194)
_WinImgArrowDown (195)
_WinImgBarcode (74)
_WinImgBin (144)
_WinImgBold (131)
_WinImgBullets (138)
_WinImgCalc (54)
_WinImgCalendar (56)
_WinImgCancel (42)
_WinImgClear (9)
_WinImgClock (55)
_WinImgClose (5)
_WinImgColorBkg (140)
_WinImgColorFg (139)
_WinImgComboBox (100)
_WinImgConfig (58)

_WinImgConnect (85)

_WinImgCopy (44)
_WinImgCut (43)
_WinImgDataComm (31)
_WinImgDblFind (20)
_WinImgDblRemove (10)
_WinImgDelete (3)
_WinImgDiagnosis (60)
_WinImgDisconnect (86)
_WinImgEdit (2)
_WinImgEditor (78)
_WinImgEmail (24)
_WinImgEuro (71)

16x16 24x24 32x32 48x48 16x16 24x24 32x32 48x48

_WinImgExcel (27)

_WinImgExcelClose (82)

_WinImgExcelExport (84)

_WinImgExcelImport (83)

_WinImgExcelOpen (81)

_WinImgExchange (89)

_WinImgExit (51)

_WinImgExport (33)

_WinImgFax (22)

_WinImgFilter (8)

_WinImgFind (15)

_WinImgFirst (11)

_WinImgHelp (52)

_WinImgHelpTip (53)

_WinImgIExplorer (36)

_WinImgImport (34)

_WinImgInfoSymbol (80)

_WinImgInstall (29)

_WinImgInternet (37)

_WinImgItalic (132)

_WinImgKeyRepair (107)

_WinImgLast (14)

_WinImgLink (66)

_WinImgMail (23)

_WinImgMarkAllClear (103)

_WinImgMarkAllSet (104)

_WinImgMarkFilterClear (105)

_WinImgMarkFilterSet (106)

_WinImgMarkOneClear (101)

_WinImgMarkOneSet (102)

_WinImgMdiArrange (114)

_WinImgMdiCascade (111)

_WinImgMdiNext (115)

_WinImgMdiPrev (116)

_WinImgMdiTileHorz (112)

_WinImgMdiTileVert (113)

_WinImgMoney (61)

_WinImgNetwork (32)

_WinImgNew (1)

_WinImgNext (13)

_WinImgObjDivider (154)

_WinImgObjDividerUp (157)

_WinImgObjEdit (152)

_WinImgObjGroup (153)

_WinImgObjIcon (156)

_WinImgObjLabel (151)

_WinImgObjPicture (155)

_WinImgODBC (35)

_WinImgOk (41)

_WinImgOpen (4)

_WinImgOptimize (59)

_WinImgOrder (65)

_WinImgPageFirst (91)

_WinImgPageFlip (125)

_WinImgPageLast (94)

_WinImgPageNext (93)

_WinImgPagePrev (92)

_WinImgPartner (62)

_WinImgPassword (79)

_WinImgPaste (45)

_WinImgPause (95)

_WinImgPhone (25)

_WinImgPicture (77)

_WinImgPrev (12)

_WinImgPreview (16)

_WinImgPrint (21)

_WinImgRadioCheck (99)

_WinImgRedo (47)

_WinImgRefresh (17)

_WinImgReplace (199)

_WinImgReplaceAll (200)

_WinImgReport (145)

_WinImgRotateL (50)

_WinImgRotateR (49)

_WinImgSave (6)

_WinImgSaveAll (7)

_WinImgSearch (196)

_WinImgSearchNext (197)

_WinImgSearchPrev (198)

_WinImgSettings (57)

_WinImgShield (90)

_WinImgSortDown (19)

_WinImgSortUp (18)

_WinImgStart (97)

_WinImgStatistics (73)

_WinImgTable (76)

_WinImgTotal (72)

_WinImgTree (75)

_WinImgUnderline (133)

_WinImgUndo (46)

_WinImgUninstall (30)

_WinImgUser (68)

_WinImgUserGroup (67)

_WinImgValDeleted (165)

_WinImgValElmBinObject (171)

_WinImgValElmFrame (172)

_WinImgValElmMenu (173)

**_WinImgValElmMetaPicture
(174)**

_WinImgValElmPicture (175)

_WinImgValElmPrintDoc (176)

_WinImgValElmPrintDocRecord

(177)

_WinImgValElmPrintForm

(178)

_WinImgValElmPrintFormList

(179)

_WinImgValElmProcedure

(180)

_WinImgValElmSelection (181)

_WinImgValElmTable (182)

_WinImgValElmText (183)

_WinImgValModified (164)

_WinImgValNotVerified (162)

_WinImgValUndefined (161)

_WinImgValVerified (163)

_WinImgWarning (96)

_WinImgWord (26)

_WinImgZoomFitPage (123)

_WinImgZoomFitPageWidth

(124)

_WinImgZoomMinus (122)

_WinImgZoomPlus (121)

Kontakt

ImageTileUser



Nummer der Kachel einer Grafik

Typ **int**

Liste, Objekte, WinPropGet(),

Siehe WinPropSet(),

GrayingModeUser

Je nach Objekt ist folgendes zu beachten:

- MenuItem

Hat diese Eigenschaft einen Wert größer 0, wird eine Kachel in der Menü-Grafik aus der Eigenschaft TileNameMenu des übergeordneten Frame-Objekts referenziert.

Die Eigenschaft ImageTile wird in diesem Fall ignoriert und nur verwendet, wenn ImageTileUser den Wert 0 besitzt.

- Button, DateEdit, DecimalEdit, FloatEdit

Hat diese Eigenschaft einen Wert größer 0, wird sie an Stelle der Eigenschaft ImageTile verwendet, um eine Kachel aus einer importierten Button-Grafik zu referenzieren.

Die zu verwendende Button-Grafik wird aus den Eigenschaften TileNameNormal, TileNameSelected und TileNamePressed (Eigenschaften des Frames) für folgende Button-Zustände selektiert:

TileNameNormal Button ist weder gedrückt noch fokussiert

TileNameSelected Button ist fokussiert

TileNamePressed Button ist gedrückt

Sind die Eigenschaft bei dem übergeordneten Fenster-Objekt (MdiFrame) nicht gesetzt, werden die Eigenschaften des nächsten übergeordneten Fenster-Objekts (AppFrame) verwendet.

Bei DateEdit-, DecimalEdit- und FloatEdit-Objekten bestimmt die Eigenschaft das Aussehen des InputCtrl.

- Toolbar-Button, Statusbar-Button

Analog zum Button-Objekt, jedoch kann pro Toolbar / Statusbar eine Button-Grafik angegeben werden.

- TreeNode

Hat die Eigenschaft einen Wert größer 0, wird die erste Kachel einer Gruppe aus einer importierten TreeView-Grafik referenziert. Die TreeView-Grafik wird beim TreeView-Objekt in der Eigenschaft TileNameUser angegeben.

Bei den aufgelisteten Objekten, mit Ausnahme von MenuItem, wird in Abhängigkeit vom Wert der Eigenschaft TileSize folgende Kachelgrafik referenziert:

- TrueColor-Modus (24, 32 BPP)
 - ◆ BtnTile.tcm.<TileSize>

Kontakt

- ◆ BtnTile.tcm
- ◆ BtnTile.hcm.<TileSize>
- ◆ BtnTile.hcm
- ◆ BtnTile.<TileSize>
- ◆ BtnTile
- HiColor-Modus (16 BPP)
 - ◆ BtnTile.hcm.<TileSize>
 - ◆ BtnTile.hcm
 - ◆ BtnTile<TileSize>
 - ◆ BtnTile
- Sonstige Modi
 - ◆ BtnTile<TileSize>
 - ◆ BtnTile



Die Grafiken mit der Erweiterung .<TileSize> werden nur verwendet, wenn die Eigenschaft TileTheme auf WinTileThemeEnhanced gesetzt ist.



Benutzerdefinierte Kacheln (Tiles) in der gesperrten Darstellung werden von CONZEPT 16 automatisch in Graustufen umkodiert. Die Eigenschaft GrayingModeUser kann gesetzt werden, um eine Graufärbung zu erreichen, die sich besser vom Normal-Zustand abhebt.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

NumIcon

Nummer der Icon-Grafik

Typ int

Siehe Liste, Objekte

Dieser Eigenschaft kann die Nummer einer importierten oder externen Icon-Grafik übergeben werden. Standardmäßig hat die Eigenschaft den Wert 1. Bei Wert 1 wird das erste Icon der Icon-Grafik referenziert.

In Fensterobjekten kann die Eigenschaft auf den Wert 0 gesetzt werden. In diesem Fall wird für den Fenstertitel und für die Taskleiste jeweils ein passendes Icon der Datei ausgewählt.

Wird die Eigenschaft bei Icon-Objekten auf -1 gesetzt, wird die Eigenschaft TileSize des Objektes verwendet um ein Icon in der passenden Größe zu ermitteln.

Kontakt

ThemeMenuCntxtTileSize



Größe der Bilder von ImageTile und Icon

Typ int

Siehe Liste, Objekte, WinPropGet(),
WinPropSet(), ThemeMenuTileSize

Über diese Eigenschaft wird die Größe der Bilder von ImageTile und Icon für Kontexmenüeinträge im ToolbarMenu-Objekt gesteuert.



Die Eigenschaft wird nur ausgewertet, wenn das Objekt im Modern Theme Style dargestellt wird (StyleTheme = WinStyleThemeModern beim Objekt oder einem Elternobjekt).

Bei Objekten außer dem App-Objekt können Werte zwischen 1 und 256 Pixel angegeben werden. Folgende Werte sind empfohlen:

16

Die Bilder werden in der Auflösung 16 x 16 Pixel dargestellt.

24

Die Bilder werden in der Auflösung 24 x 24 Pixel dargestellt.

32

Die Bilder werden in der Auflösung 32 x 32 Pixel dargestellt.

48

Die Bilder werden in der Auflösung 48 x 48 Pixel dargestellt.

WinTileSizeDefault Die Bilder werden in der Standardgröße (16 x 16 Pixel) dargestellt.

WinTileSizeApp Die Bilder werden in der definierten Größe des App-Objektes für den jeweiligen Objekttyp dargestellt (App->wpTileSize).

Beim App-Objekt sind die Werte 16, 24, 32, 48 und WinTileSizeDefault möglich.

Ist das Bild nicht in der angegebene Größe vorhanden, dann wird das Bild mit der kleinsten abweichenden Größe verwendet. Bei mehreren Bildern mit derselben Größe (nur Icon), wird jenes mit der kleinsten Abweichung bezügl. der Farbtiefe des Bildschirms verwendet.

Optional kann beim App-Objekt der betreffende Objekttyp als Parameter übergeben werden um alle Objekte dieses Typs zu beeinflussen, bei denen die Eigenschaft ThemeMenuCntxtTileSize auf WinTileSizeApp steht. Wird kein Parameter angegeben, bezieht sich die Größe auf alle Objekte. Folgende Objekttypen können angegeben werden:

- WinTypeToolbarMenu

Beispiele:

```
// Applikations-Update deaktivieren _App->wpAutoUpdate # false; // Größe der Standard Image-Tiles a
```

Kontakt

ThemeMenuItemSize



Größe der Bilder von ImageTile und Icon

Typ int

Liste, Objekte, WinPropGet(),

Siehe WinPropSet(),

ThemeMenuItemSize

Über diese Eigenschaft wird die Größe der Bilder von ImageTile und Icon für Menüeinträge im ToolbarMenu-Objekt gesteuert.



Die Eigenschaft wird nur ausgewertet, wenn das Objekt im Modern Theme Style dargestellt wird (StyleTheme = WinStyleThemeModern beim Objekt oder einem Elternobjekt).

Bei Objekten außer dem App-Objekt können Werte zwischen 1 und 256 Pixel angegeben werden. Folgende Werte sind empfohlen:

16

Die Bilder werden in der Auflösung 16 x 16 Pixel dargestellt.

24

Die Bilder werden in der Auflösung 24 x 24 Pixel dargestellt.

32

Die Bilder werden in der Auflösung 32 x 32 Pixel dargestellt.

48

Die Bilder werden in der Auflösung 48 x 48 Pixel dargestellt.

WinMenuItemSizeDefault Die Bilder werden in der Standardgröße (16 x 16 Pixel) dargestellt.

WinMenuItemSizeApp

Die Bilder werden in der definierten Größe des App-Objektes für den jeweiligen Objekttyp dargestellt (App->wpMenuItemSize).

Beim App-Objekt sind die Werte 16, 24, 32, 48 und WinMenuItemSizeDefault möglich.

Ist das Bild nicht in der angegebene Größe vorhanden, dann wird das Bild mit der kleinsten abweichenden Größe verwendet. Bei mehreren Bildern mit derselben Größe (nur Icon), wird jenes mit der kleinsten Abweichung bezügl. der Farbtiefe des Bildschirms verwendet.

Optional kann beim App-Objekt der betreffende Objekttyp als Parameter übergeben werden um alle Objekte dieses Typs zu beeinflussen, bei denen die Eigenschaft ThemeMenuItemSize auf WinMenuItemSizeApp steht. Wird kein Parameter angegeben, bezieht sich die Größe auf alle Objekte. Folgende Objekttypen können angegeben werden:

- WinTypeToolbarMenu

Beispiele:

```
// Applikations-Update deaktivieren _App->wpAutoUpdate # false; // Größe der Standard Image-Tiles a
```

Kontakt

TileButtonSpacing Höhe der Toolbar-Schaltflächen mit benutzerdefinierten Kacheln Typ int

Siehe [Liste](#), [Objekte](#), [WinPropGet\(\)](#), [WinPropSet\(\)](#)

Diese Eigenschaft beeinflusst die Höhe der Toolbar-Schaltflächen mit benutzerdefinierten Kacheln.

Die Eigenschaft wirkt sich nur aus, wenn die Eigenschaft [TileTheme](#) des [App-Objektes](#) auf [_WinTileThemeEnhanced](#) gesetzt ist.

Es stehen folgende Konstanten zur Verfügung:

_WinTileButtonSpacingClassic

Benutzerdefinierte Kacheln haben in etwa dieselbe Höhe, wie Schaltflächen im Classic-Modus ([TileTheme](#) = [_WinTileThemeClassic](#)).

_WinTileButtonSpacingMedium

Benutzerdefinierte Kacheln haben eine zusätzliche Höhe von 4 Pixeln gegenüber dem Classic-Modus.

_WinTileButtonSpacingEnhanced Benutzerdefinierte Kacheln haben eine zusätzliche Höhe von 8 Pixeln gegenüber dem Classic-Modus.

Standardmäßig ist die Eigenschaft auf [_WinTileButtonSpacingEnhanced](#) gesetzt.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle [WinPropGet\(\)](#) und [WinPropSet\(\)](#) gelesen bzw. geändert werden.

TileNameHeader

Name der Grafik (Spaltenkopf)

Typ alpha(40)

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Dieser Eigenschaft kann der Name einer importierten Kachelgrafik zugewiesen werden.

Die Eigenschaft HdrImageTileUser beim Spalten-Objekt definiert die zu verwendende Nummer der Kachel.

Die Eigenschaft kann im Eigenschaftsfenster des jeweiligen Objekt-Editors oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

TileNameMenu

Name der Menü-Grafik

Typ alpha(40)

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Bei Angabe des Namens einer importierten Menü-Grafik wird diese bei der Ausgabe von **Menüeinträgen** im aktuell gesetzten Menü (MenuName) oder Kontext-Menü (MenuNameCntxt) verwendet.

Bei einem **Toolbar-Objekt** werden die Bilder des Popup-Menüs verwendet, wenn nicht alle Schaltflächen der Toolbar angezeigt werden können. Statt der Schaltflächen wird "<<" angezeigt. Beim Klicken auf die Schaltfläche "<<" erscheint ein Menü mit den nicht darstellbaren Toolbar-Buttons als Menüeinträgen.

In ImageTileUser wird die Nummer der zu verwendenden Kachel angegeben.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

TileNameNormal

Name der Button-Grafik

Typ alpha(40)

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Dieser Eigenschaft kann der Name einer importierten Button-Grafik zugewiesen werden.

Die Eigenschaften TileNameSelected und TileNamePressed müssen ebenfalls gesetzt werden.

Die Button-Grafik wird dann verwendet, wenn der Button weder fokussiert noch gedrückt ist.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

TileNamePressed Name der Button-Grafik (gedrückte
Darstellung) Typ alpha(40)

Siehe Liste, Objekte, WinPropGet(), WinPropSet()

Dieser Eigenschaft kann der Name einer importierten Button-Grafik zugewiesen werden.

Die Eigenschaften TileNameNormal und TileNameSelected müssen ebenfalls gesetzt werden.

Die Button-Grafik wird dann verwendet, wenn der Button gedrückt wird.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

TileNameSelected Name der Button-Grafik (ausgewählte Darstellung) Typ alpha(40)

Siehe Liste, Objekte, WinPropGet(), WinPropSet()

Dieser Eigenschaft kann der Name einer importierten Button-Grafik zugewiesen werden.

Die Eigenschaften TileNameNormal und TileNamePressed müssen ebenfalls gesetzt werden.

Die Button-Grafik wird dann verwendet, wenn der Button fokussiert aber nicht gedrückt ist.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

TileNameUser

Name der Grafik

Typ alpha(40)

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Der Eigenschaft kann der Name einer importierten Grafik zugewiesen werden.

Die Kacheln in der Grafik werden von den Knoten- und Spalten-Objekten zur Bildausgabe verwendet.

- TreeView

Die Grafik muss als TreeView-Grafik importiert werden. Die Zuordnung der Kachelnummer geschieht über die Eigenschaft ImageTileUser des Knoten-Objekts.

- RecList, DataList

Die Grafik muss als Button-Grafik importiert werden. Die Zuordnung der Kachelnummer geschieht über den Inhalt der Spalte. Wird beispielsweise der Inhalt der Spalte auf 5 gesetzt, wird das Bild mit der Kachelnummer fünf angezeigt. Über die Eigenschaft ClmTypeImage der Spalte wird festgelegt, ob ein importiertes oder ein CONZEPT 16-Bild angezeigt werden soll.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt



TileSize

Größe der Bilder von ImageTile, NodeStyle und Icon Typ int

Siehe
Liste, Objekte, WinPropGet(), WinPropSet(),

TileTheme

Über diese Eigenschaft wird die Größe der Bilder von ImageTile, NodeStyle und Icon gesteuert.

Bei Objekten außer dem App-Objekt können Werte zwischen 1 und 256 Pixel angegeben werden. Folgende Werte sind empfohlen:

- 16 Die Bilder werden in der Auflösung 16 x 16 Pixel dargestellt.
- 24 Die Bilder werden in der Auflösung 24 x 24 Pixel dargestellt.
- 32 Die Bilder werden in der Auflösung 32 x 32 Pixel dargestellt.
- 48 Die Bilder werden in der Auflösung 48 x 48 Pixel dargestellt.

WinTileSizeDefault Die Bilder werden in der Standardgröße (16 x 16 Pixel) dargestellt.

WinTileSizeApp Die Bilder werden in der definierten Größe des App-Objektes für den jeweiligen Objekttyp dargestellt (App->wpTileSize).

Beim App-Objekt sind die Werte 16, 24, 32, 48 und WinTileSizeDefault möglich.

Ist das Bild nicht in der angegebene Größe vorhanden, dann wird das Bild mit der kleinsten abweichenden Größe verwendet. Bei mehreren Bildern mit derselben Größe (nur Icon), wird jenes mit der kleinsten Abweichung bezügl. der Farbtiefe des Bildschirms verwendet.

Optional kann beim App-Objekt der betreffende Objekttyp als Parameter übergeben werden um alle Objekte dieses Typs zu beeinflussen, bei denen die Eigenschaft TileSize auf WinTileSizeApp steht. Wird kein Parameter angegeben, bezieht sich die Größe auf alle Objekte. Folgende Objekttypen können angegeben werden:

- WinTypeButton
- WinTypeColorButton
- WinTypeDataList
- WinTypeDataListPopup
- WinTypeDateEdit
- WinTypeDecimalEdit
- WinTypeFloatEdit
- WinTypeHyperLink
- WinTypeIcon
- WinTypeMenuButton
- WinTypeNotebook
- WinTypeRecList
- WinTypeRecListPopup
- WinTypeStatusbar
- WinTypeToolbar
- WinTypeTreeView

Kontakt

In Menüs wird immer die Größe 16 x 16 Pixel verwendet. Die Größe von GroupTile-Button-Objekten wird automatisch aus dem zur Verfügung stehenden Platz ermittelt.



Ist die Eigenschaft TileTheme des App-Objektes auf WinTileThemeClassic gesetzt, wird die Eigenschaft TileSize nicht ausgewertet.

Beispiele:

```
// Applikations-Update deaktivieren _App->wpAutoUpdate # false; // Größe der Standard Image-Tiles a
```

Kontakt

TypeIcon 

Typ des Symbols

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Über diese Eigenschaft wird dem Objekt ein Symbol zugeordnet. Bei einem Frame-Objekt erscheint das Symbol in der Titel- und Taskleiste. Beim Icon-Objekt werden die Symbole immer in 32 x 32 Pixel dargestellt.

Hat die Eigenschaft den Wert _WinIcoExternFile, wird das in der Eigenschaft Icon (bei Frame-Objekten) bzw. FileName (bei Icon-Objekten) angegebene externe Symbol angezeigt.

Weitere Werte sind:

Konstante (Wert)	ohne Theme	mit Theme	Beschreibung
	16x16 32x32 16x16 32x32		
<u>_WinIcoApplication</u> (1)			Anwendungssymbol
<u>_WinIcoInformation</u> (2)			Informationssymbol
<u>_WinIcoError</u> (3)			Fehlersymbol
<u>_WinIcoWarning</u> (4)			Warnsymbol
<u>_WinIcoQuestion</u> (5)			Fragesymbol
<u>_WinIcoShield</u> (6)			Sicherheitssymbol
			Sofern die Eigenschaft <u>StyleTheme</u> so eingestellt wird, das das <u>Icon</u> -Objekt im <u>Modern Theme Style</u> dargestellt wird, wird die Variante der Icons in der Spalte <i>mit Theme</i> verwendet. Wird <u>WinDialogBox()</u> im <u>Modern Theme Style</u> dargestellt, werden ebenfalls die modernen Icons verwendet.

Kontakt

Help (Hilfe-Eigenschaften)

<u>Eigenschaften</u>	
<u>sortiert nach</u>	
<u>Gruppen,</u>	
<u>Siehe Alphabetische</u>	
<u>Liste aller</u>	
<u>Eigenschaften</u>	
<u>Help</u>	Text der Bubble-Help / HTML-Hilfeseite
<u>HelpFile</u>	Dateiname der CHM-Hilfedatei
<u>HelpTip</u>	Text des Helptip
<u>HelpTipJustify</u>	Ausrichtung des HelpTip-Textes
<u>HelpTipSysFont</u>	Schriftart des Helptip
<u>HelpTipTimeDelay</u>	Zeit, bis der Helptip angezeigt wird
<u>HelpTipTimeShow</u>	Anzeigedauer des Helptips
<u>StatusItemPos</u>	Statusbar-Button für Textanzeige
<u>StatusItemText</u>	Ausgabetext in der Status

Help

Text der Bubble-Help / HTML-Hilfeseite

Typ [alpha\(8192\)](#)

Siehe [Liste](#), [Objekte](#), [WinPropGet\(\)](#),

Dieser Text erscheint als Bubble-Help (kleines Hilfenster), wenn der Benutzer auf dem Objekt steht und die -Taste drückt.

Bei einem Dialog, der die Fragezeichenschaltfläche enthält, kann der Benutzer die Bubble-Help durch Klick auf die Fragezeichenschaltfläche und Klick auf das Objekt starten.

Falls bei dem Objekt oder bei einem Elternobjekt in der Eigenschaft HelpFile eine CHM-Hilfedatei angegeben wurde, kann in dieser Eigenschaft auch eine HTML-Seite der CHM-Datei eingetragen werden:

Syntax: 'HTM:<html-Seite>'

- Positionieren innerhalb einer HTML-Seite

Für das Positionieren innerhalb einer HTML-Seite muss ein Anker innerhalb der HTML-Seite definiert sein.

Syntax: 'HTM:<dateiname>#<ankernname>'

Der Ankernname innerhalb der HTML-Seite muss wie folgt definiert sein: <a

name="<ankernname>">Hier folgt zum Beispiel der Titel Beispiel:

\$ButtonClose->wpHelp # 'HTM:open_dialog#button_ok'

Definition in der HTML-Seite:

Schließen-Schaltfläche

Beim Aufruf der HTML-Hilfe für die Schaltfläche "ButtonClose" gelangt der Benutzer direkt zur Beschreibung der Schaltfläche. Beinhaltet der Ankernname Leerzeichen, müssen diese durch die Zeichenfolge '%20' ersetzt werden.

- Auf Index der HTML-Hilfe positionieren

Des Weiteren besteht die Möglichkeit in Analogie zu 'HTM:' den Präfix 'KEY:' anzugeben. Dadurch wird der Index der HTML-Hilfe mit dem angegebenen Schlüssel (der in der HTML-Hilfe definiert sein muss) aufgerufen:

Beispiel:

\$EditArtikel->wpHelp # 'KEY:Artikel'

Kontakt

Hierbei ist "Artikel" der Schlüssel in der HTML-Hilfe.



Die HTML-Seiten, die in der CHM-Datei zusammengestellt werden, müssen die Erweiterung HTM aufweisen, damit sie angezeigt werden können.

Eine bestimmte Seite einer CHM-Datei kann ebenfalls über die Funktion SysExecute() aufgerufen werden. Dazu wird das HTML Help Executable (hh.exe) gestartet und als Argument die gewünschte Seite angegeben.

Beispiel:

```
// Aufruf der Hilfe-Seite der Funktion SysExecute() SysExecute('hh.exe', 'C:\C16\Client\c16.chm::/
```

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

HelpFile

Dateiname der CHM-Hilfedatei

Typ alpha(8192)

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In dieser Eigenschaft kann eine Hilfedatei im Format "Compiled HTML"(CHM)

angegeben werden. Beim Drücken der Taste  wird die Datei aus dem aktuellen Verzeichnis gestartet. Wird dem Namen der Datei ein "!" vorangestellt, wird die Hilfe aus dem Verzeichnis aufgerufen, aus dem auch der Client gestartet wurde.

Soll eine bestimmte Seite aufgerufen werden, kann diese Seite in der Eigenschaft Help angegeben werden.



Falls bei einem Menüeintrag oder einem Toolbar-Button des Frames oder AppFrames die Eigenschaft MenuAction auf WinMnuActHelp gesetzt ist, sind die Hinweise auf der Seite WinMnuActHelp zu beachten.

Beispiel:

Der CONZEPT 16-Client wurde aus dem Verzeichnis "C:\Programme\CONZEPT 16\Client" gestartet. Die Hilfe-Datei "Codelibrary.chm" befindet sich im selben Verzeichnis.

```
...FsiPathChange (_Sys->spPathTemp) ;tObj->wpHelpFile # 'Codelibrary.chm'; // Hilfe kann nicht aufg
```

Die Hilfe-Datei kann auch relativ zum Start-Verzeichnis angegeben werden:

```
tObj->wpHelpFile # '!..\Doc\Codelibrary.chm';
```

Kontakt

HelpTip



Text des HelpTip

Typ alpha(8192)

Liste, Objekte,

Siehe HelpTipTimeDelay,

HelpTipTimeShow

Die Eigenschaft kann bei Oberflächen- und Druck-Objekten gesetzt werden. Bei Druck-Objekten wirkt sich die Eigenschaft erst innerhalb der Druckvorschau aus. Der Text der in dieser Eigenschaft eingetragen ist erscheint, wenn der Mauszeiger auf dem Objekt stehen bleibt. Über die Eigenschaften HelpTipTimeDelay und HelpTipTimeShow kann die Zeitverzögerung bis zur Anzeige und die Anzeigedauer der Hilfe festgelegt werden.

Bei den Objekten RecNavigator und Slider ist folgendes zu beachten:

Durch das Trennzeichen '|' ist es möglich, für jede Schaltfläche des RecNavigator-Objektes bzw. für jeden Wert des Slider-Objektes einen HelpTip zu definieren. Die Reihenfolge beim RecNavigator ist fest definiert:

```
$RecNav->wpHelpTip # 'Prev | Next | SeekPrev | SeekNext | First | Last | Insert | Delete | Replac
```

Die Reihenfolge muss immer eingehalten werden, egal welche StyleConsole eingestellt ist. Durch die Eingabe von zwei Trennzeichen unmittelbar hintereinander wird angegeben, dass für die entsprechende Schaltfläche kein HelpTip vorhanden ist.

Beim Slider werden die HelpTips in der Reihenfolge der Werte angegeben. Sind weniger Texte vorhanden als Werte im Wertebereich, wiederholen sich die Texte entsprechend.

Die Eigenschaft kann auch bei Druck-Objekten gesetzt werden. Der HelpTip wird dann angezeigt, wenn der Druckjob innerhalb des PrtJobPreview-Objekts angezeigt und der Mauszeiger auf dem entsprechenden Objekt zur Ruhe kommt.

Kontakt

HelpTipJustify



Ausrichtung des HelpTip-Textes

Typ int

Siehe Liste, Objekte, HelpTip,

Siehe WinPropGet(), WinPropSet()

In dieser Eigenschaft wird angegeben, mit welcher Ausrichtung ein mehrzeiliger HelpTip ausgegeben werden soll. Folgende Ausrichtungen stehen zur Verfügung:

WinJustLeft - Text wird linksbündig dargestellt

WinJustCenter - Text wird zentriert dargestellt

WinJustRight - Text wird rechtsbündig dargestellt

WinJustParent - Text wird wie beim übergeordneten Objekt angegeben dargestellt Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

HelpTipSysFont



Schriftart des HelpTip

Typ logic

Liste, Objekte,

WinPropGet(),

Siehe WinPropSet(),

Betriebssystemdarstellung

(Blog)

Bei true wird anstelle der bei dem Objekt eingestellten Schriftart (siehe Font) die Schriftart verwendet, den Windows zum Zeichnen des Desktops benutzt.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

HelpTipTimeDelay



Zeit, bis der HelpTip angezeigt wird

Typ int

Liste, Objekte,

Siehe HelpTipTimeShow,

WinPropGet(), WinPropSet()

Die Eigenschaft dient der Einstellung der Verzögerung (in Millisekunden), bis der HelpTip des Objekts angezeigt wird.

Wird in der Eigenschaft -1 angegeben, erscheint der HelpTip nach 200 Millisekunden.

Es können Werte zwischen -1 und 60000 (1 Minute) angegeben werden.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

HelpTipTimeShow

Anzeigedauer des HelpTip

Typ int

Liste, Objekte,

Siehe HelpTipTimeDelay,

WinPropGet(),

WinPropSet()

Die Eigenschaft dient der Einstellung der Anzeigedauer (in Millisekunden) des HelpTip.

Zulässige Werte sind -1 bis 300000 (5 Minuten). Wird in der Eigenschaft -1 angegeben, beträgt die Anzeigedauer 3000 Millisekunden. Dies entspricht dem Vorgabewert. Bei der Angabe von 0 wird kein HelpTip angezeigt.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

StatusItemPos

Statusbar-Button für Textanzeige

Typ int

Liste, Objekte,

Siehe StatusItemText,

WinPropGet(), WinPropSet(),

Blog

Diese Eigenschaft gibt die Position des Statusbar-Buttons an (von rechts nach links), welcher den über StatusItemText definierten Text anzeigen soll.

Ein Wert von 0 schaltet die Statusbar in den Menü-Modus um. Der Text erstreckt sich dann über die gesamte Statusbar.

Wird der Eigenschaft StatusItemPos der Wert -1 zugewiesen, wird nach dem übergeordneten Frame-Objekt gesucht, das einen Wert ≥ 0 enthält. Dadurch ist es möglich, dass ein untergeordnetes MdiFrame die Statuszeile des Applikationsfenster zur Ausgabe benutzen kann.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

StatusItemText

Ausgabetext in Statuszeile

Typ [alpha\(8192\)](#)

[Liste, Objekte,](#)

Siehe [StatusItemPos,](#)

[WinPropGet\(\)](#),

[WinPropSet\(\)](#), [Blog](#)

Der in dieser Eigenschaft angegebene Text wird in der **Statuszeile** ausgegeben.

Bei [MenuItem](#)-Objekten wird der Text in der Statuszeile ausgegeben, wenn der Menüeintrag angewählt wird. Bei [Toolbar-Button](#)-Objekten wird der Text in der Statuszeile ausgegeben, wenn sich die Maus über dem Button befindet.

Bei allen anderen Objekten wird der Text ausgegeben, wenn das Objekt den Tastaturfokus erhält.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle [WinPropGet\(\)](#) und [WinPropSet\(\)](#) gelesen bzw. geändert werden.

Focus (Eingabe-Eigenschaften)

Eigenschaften

sortiert nach

Gruppen,

Siehe

Alphabetische

Liste aller

TabOrder Reihenfolge der Notizbuchseiten

TabPos Reihenfolge des Fokus

TabStop Erhalt des Tastaturfokus

Kontakt

TabOrder



Reihenfolge der Notizbuchseiten

Typ int
Siehe Liste, Objekte, WinPropGet(),
WinPropSet()

Mit dieser Eigenschaft kann die Fokusreihenfolge bei Verwendung der Tasten und + im Notebook-Objekt beeinflusst werden.

Die Eigenschaft kann auf folgende Werte gesetzt werden:

- WinTabOrderNormal (default)

Steht der Fokus in dem letzten Objekt der NotebookPage und die Taste wird gedrückt, bekommt das erste Objekt außerhalb des Notizbuches den Fokus.
Entsprechendes gilt für den Fall, dass der Fokus im ersten Objekt der

NotebookPage steht und die Tasten + gedrückt werden. Es wird dann auf das letzte Objekt ausserhalb des Notebook-Objektes fokussiert.

Ist die Eigenschaft TabStop beim Notebook-Objekt auf false gesetzt, verbleibt der Fokus in der NotebookPage.

- WinTabOrderPageChild

Steht der Fokus auf dem letzten Objekt der NotebookPage und die Taste wird gedrückt, bekommt das erste Objekt der nächsten Notizbuchseite den Fokus. Das Notizbuch verliert den Fokus, wenn das letzte Objekt auf der letzten Notizbuchseite den Fokus verliert.
Entsprechendes gilt für den Fall, dass der

Fokus im ersten Objekt der Notebook-Seite steht und die Tasten + gedrückt werden. Es wird dann auf das letzte Objekt der vorhergehenden Notizbuchseite fokussiert oder das Notebook-Objekt wird verlassen.

Ist die maximale Eingabelänge (LengthMax) des letzten Objektes erreicht, findet ebenfalls ein Wechsel der Seite statt, wenn die Eigenschaft LeaveAtEnd dieses Objektes auf true gesetzt ist.

Damit ein Wechsel auf die entsprechende Notizbuchseite erfolgen kann, muss die Seite sichtbar (Visible = true) sein, erreichbar (Disabled = false) sein und ein Objekt enthalten, das den Fokus bekommen kann.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

TabPos

Reihenfolge des Fokus

Typ int

Liste, Objekte,

Siehe WinPropGet(),
WinPropSet()

Die Eigenschaft setzt oder ermittelt den Wert für die Erstellungs- bzw. Fokussierungs-Reihenfolge eines Fensterobjektes.

Der Designer erlaubt bereits die Änderung der Reihenfolge zur Entwicklungszeit. Durch Setzen der Eigenschaft kann die Reihenfolge auch prozedural (zur Laufzeit) verändert oder abgefragt werden.

Dabei hat der Wert der Eigenschaft genau dieselbe Bedeutung wie bei der Festlegung im Designer:

0 Objekt ist das Erste in der Reihenfolge

1..n Objekt gelangt an die Stelle 1..n der Reihenfolge (bei n Objekten im aktuellen Frame bzw. Container)

> n Objekt gelangt ans Ende der Reihenfolge

 Der Wert ist (genau wie auch im Designer) immer auf die aktuelle Ebene bezogen, in der sich das Objekt befindet.

Die Eigenschaft kann nicht zum Entwurfszeitpunkt im Eigenschaftsfenster geändert werden. Über das Kontextmenü kann aber die Fokusreihenfolge angezeigt und durch Doppelklick auf die angezeigte Zahl auch verändert werden.

Zur Laufzeit kann über die Befehle WinPropGet() und WinPropSet() der Wert der Eigenschaft gelesen bzw. geändert werden.

TabStop

Erhalt des Tastaturfokus

Typ logic

Siehe Liste, Objekte

Wenn diese Eigenschaft auf true gesetzt ist, erhält das Eingabe-Objekt den Tastaturfokus (durch die Tasten , , , usw.).

Bei false wird das Objekt nicht angesprungen. Das Objekt kann nur durch das Klicken mit der linken Maustaste auf das Eingabe-Objekt den Fokus erhalten. Soll auch dies unterbunden werden, muss die Eigenschaft ReadOnly auf true gesetzt sein.



Ist die Eigenschaft nicht gesetzt, kann der Fokus auch nicht mit dem Befehl WinFocusSet() in das Eingabeobjekt gesetzt werden.

Bei einem Notebook-Objekt wird der Eingabefokus vom Notebook-Objekt direkt zum ersten untergeordneten Objekt geleitet, das den Eingabefokus erhalten kann.

Wird die Eigenschaft beim RtfEdit-Objekt gesetzt, ist zu beachten, dass das Objekt zwar mit  den Fokus bekommen kann, das Drücken der Taste im RtfEdit-Objekt jedoch zum Schreiben eines Tabulators führt. Die Steuerung von  und  +  wird in der Eigenschaft RtfEditFlags vorgenommen.

Durch Drücken von  oder  +  werden die untergeordneten Objekte zyklisch durchlaufen.

Kontakt

Key (Tasten-Eigenschaften)

Eigenschaften

sortiert nach

Gruppen,

Siehe Alphabetische

Liste aller

Eigenschaften

Default Die Schaltfläche wird als Default-Schaltfläche definiert

EditKeyEnd Auslösende Funktionstaste zum Beenden des Bearbeitungsmodus

EditKeyStart Auslösende Funktionstaste zum Starten des Bearbeitungsmodus

EditMouseStart Mausaktion zum Starten der Bearbeitung

EmulateKeys Simulation der Tastensteuerung der textbasierten Oberfläche von

CONZEPT 16

MenuKey Auslösende Funktionstaste

MouseKeySort Mausaktion zum Setzen der Sortierung

ReturnKeyClick Mit Return-Taste auslösen

Kontakt

Default



Die Schaltfläche wird als Default-Schaltfläche definiert

Typ [logic](#)

Siehe [Liste, Objekte](#)

Wenn diese Eigenschaft auf [true](#) gesetzt ist, gilt die Schaltfläche als Default-Schaltfläche.

Das bedeutet, dass die Schaltfläche einen schwarzen Rahmen bekommt und sie automatisch bei einem ausgewählt wird. Bei Verwendung von mehreren Schaltflächen in einem Dialog sollte eine Schaltfläche als Default-Schaltfläche definiert werden, um dem Benutzer eine Standardaktion vorzugeben.

Die Default-Schaltfläche kann nicht mit der -Taste ausgelöst werden, wenn in dem übergeordneten Dialog die Eigenschaft [EmulateKeys](#) auf [true](#) gesetzt wurde.



Diese Eigenschaft kann nicht gesetzt werden, wenn die Eigenschaft [StyleButton](#) auf [WinStyleButtonTBar](#) gesetzt ist. In diesem Fall wird der Button im Windows-Theme gezeichnet, in dem die Default-Darstellung nicht unterstützt wird.

Kontakt

EditKeyEnd Auslösende Funktionstaste zum Beenden des Bearbeitungsmodus

Typ int

Siehe [Liste, Objekte](#)

Die Eigenschaft **EditKeyEnd** ermöglicht die Zuordnung einer Tastenkombination. Innerhalb des [Eigenschaften-Fensters](#) kann, nachdem der Fokus auf das Eingabe-Objekt der Eigenschaft gesetzt wurde, eine Tastenkombination gedrückt werden, die automatisch in die Eigenschaft eingetragen wird.

Wird eine Tastenkombination definiert, wird bei Betätigung der Editiermodus im [RecView](#) beendet. Das RecView-Objekt muss dazu den Fokus haben.

Für das Setzen der Funktionstasten zur Laufzeit stehen folgende Konstanten zur Verfügung:

[WinKeyA..._WinKeyZ](#)

[WinKeyF1..._WinKeyF12](#)

[WinKeyNull](#)

[WinKeyHome](#)

[WinKeyEnd](#)

[WinKeyPageUp](#)

[WinKeyPageDown](#)

[WinKeyShift](#)

[WinKeyCtrl](#)

[WinKeyEsc](#)

[WinKeyReturn](#)

Die Eigenschaft kann im [Eigenschaften-Fenster](#) oder über die Befehle [WinPropGet\(\)](#) und [WinPropSet\(\)](#) gelesen bzw. geändert werden.



Wird in dem Edit-Objekt eine [DataListPopup](#) erstellt (siehe [EvtLstEditStartItem](#)), und bei der [DataListPopup](#) die Eigenschaft [LstFlags](#) auf [WinLstEditClose](#) gesetzt, wird der aktuell ausgewählte Eintrag der [DataListPopup](#) vor dem Beenden des Bearbeitungsmodus übernommen. Dies geschieht ebenfalls, wenn ein Eintrag mit der Maus ausgewählt wird.

Kontakt

EditKeyStart Auslösende Funktionstaste zum Starten des Bearbeitungsmodus Typ int

Siehe [Liste, Objekte](#)

Die Eigenschaft **EditKeyStart** ermöglicht die Zuordnung einer Tastenkombination. Innerhalb des [Eigenschaften-Fensters](#) kann, nachdem der Fokus auf das Eingabe-Objekt der Eigenschaft gesetzt wurde, eine Tastenkombination gedrückt werden, die automatisch in die Eigenschaft eingetragen wird.

Wird eine Tastenkombination definiert, wird bei Betätigung der Editiermodus des ersten Items im RecView gestartet. Das RecView-Objekt muss dazu den Fokus haben.

 Ist bei einem [Toolbar-Button](#) oder einem [Menüeintrag](#) in der Eigenschaft MenuKey die gleiche Tastenkombination gesetzt, wird der Toolbar-Button bzw. der Menüeintrag prioritätär behandelt.

Für das Setzen der Funktionstasten zur Laufzeit stehen folgende Konstanten zur Verfügung:

[WinKeyA... WinKeyZ](#)
[WinKeyF1... WinKeyF12](#)
[WinKeyNull](#)
[WinKeyHome](#)
[WinKeyEnd](#)
[WinKeyPageUp](#)
[WinKeyPageDown](#)
[WinKeyShift](#)
[WinKeyCtrl](#)
[WinKeyEsc](#)
[WinKeyReturn](#)

Die Eigenschaft kann im [Eigenschaften-Fenster](#) oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

EditMouseStart  Mausaktion zum Starten der Bearbeitung Typ int

Siehe [Liste](#), [Objekte](#), [WinPropGet\(\)](#),

In einem [RecView](#)-Objekt kann durch Anklicken eines Items oder SubItems der Bearbeitungsmodus gestartet werden.

In dieser Eigenschaft wird die Maus-Tasten-Kombination festgelegt, mit der die Bearbeitung gestartet werden kann. Folgende Konstanten können in dieser Eigenschaft gesetzt werden:

[WinMouseLeft](#)

[WinMouseMiddle](#)

[WinMouseRight](#)

[WinMouseShift](#)

[WinMouseCtrl](#)

[WinMouseDouble](#)

Die Konstanten für die Maustasten können nicht miteinander kombiniert werden. Eine Kombination mit der - und der -Taste und dem Doppelklick ist möglich.

`_WinMouseLeft | _WinMouseShift | _WinMouseDouble`

In diesem Fall muss mit der linken Maustaste, bei gedrückter -Taste ein Doppelklick ausgeführt werden.

Kontakt

EmulateKeys



Simulation der Tastensteuerung der textbasierten Oberfläche von CONZEPT 16 Typ logic

Siehe Liste, Objekte, WinPropGet(), WinPropSet()

Bei true können die von der textbasierten Oberfläche gewohnten Tasten zum Objektwechsel verwendet werden.

Folgende Taste haben dann Auswirkungen auf die Steuerung:



- Auf vorheriges Eingabeobjekt springen



- Auf nächstes Eingabeobjekt springen



- Auf nächstes Eingabeobjekt springen

Alle Dialoge, die nicht über eine eigene Eigenschaft EmulateKeys verfügen, bekommen diese Eigenschaft vererbt. Im Besonderen wirkt sich das auf die Bedienung

der Default-Schaltfläche (siehe Eigenschaft Default) aus. Da die -Taste zum Fokuswechsel verwendet wird, kann über diese Taste nicht mehr die Default-Schaltfläche ausgelöst werden.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

MenuKey

Auslösende Funktionstaste

Typ int

Siehe Liste, Objekte

Die Eigenschaft MenuKey ermöglicht die Zuordnung einer Tastenkombination. Innerhalb des Menü-Editors bzw. des Toolbar-Editors kann, nachdem der Fokus auf das Eingabe-Objekt der Eigenschaft gesetzt wurde, eine Tastenkombination gedrückt werden, die automatisch in die Eigenschaft eingetragen wird.

Wird eine Tastenkombination definiert, hat das Betätigen den gleichen Effekt wie das Auswählen des Menüeintrags über das Menü oder das Anklicken eines Toolbar-Buttons.

Menüeinträge, die ein Untermenü oder Separator besitzen, können keine Funktionstasten zugeordnet werden.

Für das Setzen der Funktionstasten zur Laufzeit stehen folgende Konstanten zur Verfügung:

- WinKeyA... WinKeyZ
- WinKeyF1... WinKeyF12
- WinKeyNull
- WinKeyHome
- WinKeyEnd
- WinKeyPageUp
- WinKeyPageDown
- WinKeyShift
- WinKeyCtrl
- WinKeyAlt
- WinKeyEsc
- WinKeyLeft
- WinKeyRight
- WinKeyUp
- WinKeyDown



Die Konstante WinKeyReturn kann für diese Eigenschaft nicht verwendet werden.

Eindeutigkeit der Funktionstaste innerhalb der Anwendung:

1. Eine durch MenuKey zugewiesene Funktionstaste muss innerhalb des Toolbar-Objekts eindeutig sein.
2. Sind mehrere Toolbar-Objekte in einem Frame vorhanden, muss jede Funktionstaste über alle Toolbar-Objekte des Frame-Objekts eindeutig sein.
3. Hat ein Frame-Objekt eine oder mehrere Werkzeugleisten und eine Menüleiste, kann eine Doppelbelegung einer Funktionstaste erfolgen (die Taste x kann sowohl in einer der Toolbars als auch im Menü definiert sein).
Dies ist sinnvoll, da Menüs globale Ressourcen sind und unabhängig von einem Frame-Objekt editiert werden können.
4. Die Regeln 1-3 gelten auch für AppFrame-Objekte.
5. Die Regeln 1 und 2 gelten für MdiFrame-Objekte.

Kontakt

Beim Betätigen einer Funktionstaste greift folgende Such- Reihenfolge:

1. Ist die gesuchte Taste im Menü definiert, ist die Suche beendet, ansonsten wird mit Schritt 2 fortgefahren.
2. Ist die gesuchte Taste in einer Toolbar des Frame-Objekts definiert, ist die Suche beendet. Ansonsten ist die Taste nicht definiert.
Handelt es sich um ein AppFrame-Objekt (mit geladenen MdiFrame-Objekten), wird mit Schritt 3 fortgefahren.
3. Ist die gesuchte Taste in einer Toolbar des MdiFrame-Objekts, ist die Suche beendet. Im anderen Fall ist die Funktionstaste nicht definiert.

Die Suchreihenfolge ist insbesondere bei Mehrfachbelegungen wichtig.

Die Eigenschaft kann im Menü-Editor / Toolbar-Editor oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

MouseKeySort

Mausaktion zum Setzen der Sortierung

Typ int
Siehe Liste, Objekte, WinPropGet(),

In einem RecList- oder DataList-Objekt kann, sofern die entsprechenden Eigenschaften gesetzt sind (siehe auch DbKeyNo bei einem Spalten-Objekt), durch Anklicken des Spaltenkopfs die Sortierung der Liste geändert werden.

In dieser Eigenschaft wird die Maus-Tasten-Kombination festgelegt, mit der die Sortierung geändert werden kann. Folgende Konstanten können in dieser Eigenschaft gesetzt werden:

- WinMouseLeft
- WinMouseMiddle
- WinMouseRight
- WinMouseShift
- WinMouseCtrl
- WinMouseDouble

Die Konstanten für die Maustasten können nicht miteinander kombiniert werden. Eine Kombination mit der - und der -Taste und dem Doppelklick ist möglich.

_WinMouseLeft | _WinMouseShift | _WinMouseDouble

In diesem Fall muss mit der linken Maustaste, bei gedrückter -Taste ein Doppelklick ausgeführt werden.

Kontakt

ReturnKeyClick



Mit Return-Taste auslösen

Typ logic

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Wird diese Eigenschaft gesetzt, kann unabhängig der Eigenschaften Default und EmulateKeys das Button-Objekt mit der -Taste aktiviert werden, sofern es den Fokus besitzt.

Kontakt

SearchKeyNext



Tastenkombination zur Suche des nächsten Treffers im TreeView Typ int

Liste, Objekte, SearchKeyStart, SearchKeyPrev, WinPropGet(), Siehe
WinPropSet()
In dieser Eigenschaft kann eine Tastenkombination definiert werden, mit der der nächste Treffer einer Suche im TreeView-Objekt ausgewählt wird. Standardmäßig ist die Eigenschaft mit der Taste vorbelegt.



Um die Suche im TreeView zu verwenden, muss die Eigenschaft SearchEnabled auf true gesetzt sein.

Kontakt

SearchKeyPrev



Tastenkombination zur Suche des vorherigen Treffers im TreeView Typ int

Liste, Objekte, SearchKeyStart, SearchKeyNext, WinPropGet(),
WinPropSet() Siehe
In dieser Eigenschaft kann eine Tastenkombination definiert werden, mit der der vorherige Treffer
einer Suche im TreeView-Objekt ausgewählt wird. Standardmäßig
ist die Eigenschaft mit der Tastenkombination + vorbelegt.



Um die Suche im TreeView zu verwenden, muss die Eigenschaft SearchEnabled auf true
gesetzt sein.

Kontakt

SearchKeyStart  Tastenkombination zur Suche im TreeView Typ int

Liste, Objekte, SearchKeyNext,
Siehe SearchKeyPrev, WinPropGet(),
WinPropSet()

In dieser Eigenschaft kann eine Tastenkombination definiert werden, mit der die Suche innerhalb eines TreeView-Objekts gestartet werden kann. Bei Betätigung erscheint am ausgewählten Knoten ein Eingabefeld. In dieses kann der Benutzer einen Suchbegriff eingeben. Dieser wird in der Eigenschaft SearchPattern gespeichert.

Standardmäßig ist die Eigenschaft mit der Tastenkombination  +  vorbelegt.



Um die Suche im TreeView zu verwenden, muss die Eigenschaft SearchEnabled auf true gesetzt sein.

Kontakt

Menu (Menü-Eigenschaften)

Liste der
Objekte, Liste
Siehe der
Eigenschaften

<u>MenuAction</u>	Menüaktion festlegen
<u>MenuBreak</u>	Zeilen- bzw. Spaltenenumbruch
<u>MenuCheck</u>	Checkmarkierung
<u>MenuDefault</u>	Der Menüeintrag wird als Default-Menüeintrag definiert
<u>MenuId</u>	Nummer des Menüeintrages
<u>MenuJustifyRight</u>	Menüeintrag rechtsbündig darstellen
<u>MenuName</u>	Name des Menüs
<u>MenuNameCntxt</u>	Name des Kontextmenüs
<u>MenuRadioCheck</u>	Menüeintrag als Auswahlknopf darstellen
<u>MenuSeparator</u>	Menüeintrag als Separator darstellen

Kontakt

MenuAction Menüaktion

festlegen Typ int

Siehe Liste, Objekte

Diese Eigenschaft steuert die Menüaktivität.

Folgende Einstellungen sind möglich:

WinMnuActClose

- Fenster wird geschlossen

WinMnuActEditCopy

- Kopiert den Inhalt des Objekts in die Zwischenablage.

WinMnuActEditCut

- Schiebt den Inhalt des Objekts in die Zwischenablage.

WinMnuActEditDelete

- Löscht den Inhalt des Objekts.

WinMnuActEditPaste

- Fügt den Inhalt der Zwischenablage in das Objekt ein.

WinMnuActEditSelectAll

- Selektiert den gesamten Inhalt des Objekts.

WinMnuActEditRedo

- Stellt die letzte rückgängig gemachte Operation wieder her.

WinMnuActEditUndo

- Macht die letzte Operation rückgängig.

WinMnuActHelp

- Aufruf der Hilfe

WinMnuActHelpContext

- Aufruf der kontextsensitiven Hilfe

WinMnuActMdiArrange

Mdi-Frame-Objekte

- Neu-Anordnung aller MdiFrame-Objekte

- Schließen des aktuellen MdiFrame-Objektes

- Maximieren aller MdiFrame-Objekte

- Minimieren aller MdiFrame-Objekte

- Das nächste MdiFrame-Objekt wird aktiviert

- Das vorhergehende MdiFrame-Objekt wird aktiviert

- Wiederherstellen aller MdiFrame-Objekte

- Alle MdiFrame-Objekte werden übereinander angeordnet

- Alle MdiFrame-Objekte werden nebeneinander angeordnet

WinMnuActMdiCascade

WinMnuActMdiClose

WinMnuActMdiMaximize

WinMnuActMdiMinimize

WinMnuActMdiNext

WinMnuActMdiPrev

WinMnuActMdiRestore

WinMnuActMdiTileHorz

WinMnuActMdiTileVert

WinMnuActMdiWindowList - Liste aller MdiFrame-Objekte

- Es wird keine Menüfunktion ausgeführt

WinMnuActNone

MenuBreak

Zeilen- bzw. Spaltenumbruch

Typ logic

Siehe Liste, Objekte

Wenn diese Eigenschaft auf true steht, wird vor dem Menüeintrag ein Umbruch durchgeführt.

Bei Menüleisten wird an dieser Stelle ein Zeilenumbruch durchgeführt. Bei

Menüfenstern findet an dieser Stelle ein Spaltenumbruch statt.

Kontakt

MenuCheck

Checkmarkierung

Typ logic

Siehe Liste, Objekte

Wenn diese Eigenschaft auf true steht, wird vor einem Menüeintrag ein Haken angezeigt.

Das Aussehen des Hakens kann mit der Eigenschaft MenuRadioCheck zu einem Punkt verändert werden.

Ein Toolbar-Button wird bei true gedrückt dargestellt.

Kontakt

MenuDefault Der Menüeintrag wird als Default-Menüeintrag definiert Typ logic

Siehe Liste, Objekte

Wenn diese Eigenschaft auf true gesetzt ist, gilt der Menüeintrag als Default-Eintrag.

Das bedeutet, dass der Menüeintrag fett dargestellt wird.

Kontakt

MenuItem



Nummer des Menüeintrages

Typ int

Siehe Liste, Objekte

Diese Nummer wird an das Ereignis übergeben, wenn der Menüeintrag ausgewählt wurde.

Kontakt

MenuJustifyRight

Menüeintrag rechtsbündig darstellen

Typ [logic](#)

Siehe [Liste, Objekte](#)

Wenn diese Eigenschaft auf [true](#) steht, werden alle folgenden Menüeinträge rechtsbündig dargestellt.



Eigenschaft wirkt sich nur bei Menüleisten aus.

Kontakt

MenuName

Name der Menüs

Typ alpha(40)

Siehe Liste, Objekte

In dieser Eigenschaft kann der Name eines Menüs eingetragen werden. Das angegebene Menü erscheint im Editierfenster als Menüleiste. Der Name des Menüs kann im Designer eingetragen oder prozedural gesetzt werden. Durch einen Doppelklick auf den Namen der Eigenschaft wird ein Auswahlfenster geöffnet.



Wird ein Dialog im Modern Theme Style (Eigenschaft StyleTheme = WinStyleThemeModern) verwendet, muss er zusätzlich über ein ToolbarMenu-Objekt verfügen, da sonst das Menü nicht angezeigt wird.

MenuNameCntxt

Name des Kontextmenüs

Typ [alpha\(40\)](#)

Siehe [Liste, Objekte](#)

Diese Eigenschaft ordnet dem zugehörigen Objekt ein Menü zu, das angezeigt wird, wenn mit der rechten Maustaste auf das Objekt geklickt wurde. Der Name des Kontextmenüs kann im Designer eingetragen oder prozedural gesetzt werden. Durch einen Doppelklick auf den Namen der Eigenschaft wird ein Auswahlfenster geöffnet.

Kontextmenüs unterstützen nicht die Eigenschaft [MenuKey](#).

Wie Menüleisten erzeugen Kontextmenüs die Ereignisse [EvtMenuInitPopup](#) und [EvtMenuCommand](#). Wird innerhalb dieser Ereignisse ein [MenuItem](#) per \$<Name> referenziert, liefert \$<Name> immer den Deskriptor des Kontextmenü-Items. Dies ist wichtig, wenn eine Menüleiste gleichzeitig als Kontextmenü verwendet wird.

Besitzt ein Objekt kein eigenes Kontextmenü, wird das Kontextmenü des übergeordneten Objekts angezeigt. Zum Beispiel wird innerhalb eines [GroupBox](#)-Objekts das Kontextmenü des übergeordneten [Frame](#)-Objekts angezeigt.

Eingabeobjekte besitzen betriebssystembedingt bereits ein eigenes Kontextmenü. Wird einem Eingabeobjekt ein Kontextmenü zugewiesen, ersetzt dieses das System-Kontextmenü. Die im System-Kontextmenü vorhandenen Befehle können über die Eigenschaft [MenuAction](#) in das neue Kontextmenü eingefügt werden.

Kontakt

MenuRadioCheck Menüeintrag als
Auswahlknopf darstellen Typ logic

Siehe Liste, Objekte

Wenn diese Eigenschaft auf true steht, wird vor dem Menüeintrag ein Punkt gezeichnet, wenn die Eigenschaft MenuCheck gesetzt ist.



Diese Eigenschaft wirkt sich nur bei Menüfenstern aus.

Kontakt

MenuSeparator Menüeintrag oder Button als Separator
darstellen Typ logic

Siehe Liste, Objekte

Wenn diese Eigenschaft auf true steht, wird das Objekt als Separator dargestellt.

Die restlichen Eigenschaften bis auf Name haben dann keine Auswirkungen mehr.

Kontakt

Embedded (Eigenschaften eingebundener Objekte)

Eigenschaften

sortiert nach

Gruppen,

Siehe

Alphabetische

Liste aller

AlignInputCtrl Ausrichtung der Eingabehilfe CalculatorSize

Größe der Taschenrechner-Eingabehilfe

InputCtrl Eingabehilfe aktivieren

AlignInputCtrl

Ausrichtung der Eingabehilfe

Typ int

Liste, Objekte,

Siehe WinPropGet(),
WinPropSet()

Mit dieser Eigenschaft kann die Eingabehilfe ausgerichtet werden.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Diese Eigenschaft hat nur Auswirkung, wenn die Eigenschaft InputCtrl auf true gesetzt ist.

Es gibt folgende Ausrichtungen:

WinAlignInpLeft - Links im Objekt

WinAlignInpRight - Rechts im Objekt

WinAlignInpTop - Oben im Objekt

WinAlignInpBottom - Unten im Objekt

Kontakt

CalculatorSize

Größe des Taschenrechner-Eingabehilfe

Typ int
Siehe Liste, Objekte, FloatEdit,

Über die Eigenschaft kann bei FloatEdit- bzw. DecimalEdit-Objekten die Größe des Taschenrechners (InputCtrl) beeinflusst werden.

Folgende Werte sind möglich:

_WinCalculatorSizeDefault Standardgröße des Taschenrechners. Dies ist der voreingestellte Wert.

_WinCalculatorSizeTouch Vergrößerter Taschenrechner für Touch-Bedienung.

InputCtrl

Eingabehilfe aktivieren

Typ logic

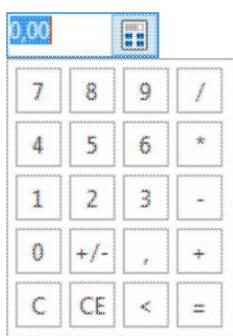
Siehe Liste, Objekte

Wenn diese Eigenschaft auf true gesetzt wird, ist im Eingabeobjekt eine Eingabehilfe vorhanden. Sie erscheint innerhalb des Eingabeobjektes und ist je nach Objekt unterschiedlich:

- FloatEdit, DecimalEdit

Anzeige eines Rechners. Durch einen Mausklick auf das Rechnersymbol wird der Rechner angezeigt. Über Tastatur erfolgt die Anzeige mit der

Tastenkombination  + .



Der Rechner kann mit der Maus oder der Tastatur bedient werden. Es stehen folgende Tasten zur Verfügung:



Ziffer



Division



Multiplikation



Addition



Subtraktion



Anzeige löschen (CE)



Zurücksetzen (C)



Vorzeichenwechsel (+/-)



Dezimaltrennzeichen. Das Zeichen ist abhängig vom Locale.

- DateEdit

Anzeige eines Kalenders über das Calendar-Objekt. Das Erscheinungsbild des Kalenders kann über einen Editor oder prozedural direkt beim Eingabeobjekt geändert werden, bevor das Popup geöffnet wird.

Durch einen Mausklick auf das Kalendersymbol wird der Kalender angezeigt.

Über Tastatur erfolgt die Anzeige mit der Tastenkombination  + .

Kontakt



In dem Kalender-Objekt kann mit Hilfe der Maus ein Datum ausgewählt werden. Die Auswahl des Monats kann über die entsprechenden Schaltflächen im Kopf des Objekts erfolgen. Durch Anklicken des Monats erscheint eine Liste von Monaten, mit der direkt der Monat angesprungen werden kann. Unter Windows Vista wird beim Anklicken des Monats in die Monatsübersicht und anschließend in die Jahresübersicht gewechselt.

Die Eingabehilfe kann ebenfalls mit der Tastatur gesteuert werden, dazu muss aber ein Datum in dem entsprechenden Eingabe-Objekt vorhanden sein. Folgende Tastenkombinationen stehen zur Verfügung:

- [strg] + [+] Nächster Tag
- [strg] + [↑] Voriger Tag
- [bildv] Nächster Monat
- [bilds] Voriger Monat
- [strg] + [bildv] Nächstes Jahr
- [strg] + [bilds] Voriges Jahr
- [strg] + [pos1] Erster des Monats
- [strg] + [ende] Letzter des Monats

Beim Wechsel des Monats oder des Jahres wird auf den gleichen Tag (zum Beispiel den 30. des Monats) gewechselt. Existiert das Datum nicht, findet auch kein Wechsel statt. Ein Wechsel vom 31. August in den nächsten Monat mit

[bildv] ist somit nicht möglich.

- IntEdit, BigIntEdit, TimeEdit

Darstellung des Objektes als "spin control". Es wird ein Auf- und ein Ab-Pfeil sichtbar. Über die Pfeile kann auf- oder absteigend geblättert werden.

Das Blättern über Tastatur kann über die Pfeiltasten hoch und runter und über die Tastenkombination [strg] + [↑] oder [strg] + [↓] vorgenommen werden.

- ColorEdit

Anzeige eines Dialoges zur Farbauswahl. Durch einen Mausklick auf die Schaltfläche wird der Dialog geöffnet.

Kontakt

Über Tastatur erfolgt die Anzeige mit der Tastenkombination + oder + .

Die Ausrichtung der Eingabehilfe kann mit der Eigenschaft AlignInputCtrl verändert werden.

Kontakt

Popup (Eigenschaften von Popup-Objekten)

Eigenschaften

sortiert nach

Gruppen,

Siehe

Alphabetische

Liste aller

ComboStyle Art der Übernahme eines Satzes aus der ComboBox

MaxLines Anzahl der Zeilen

PopupOpen ComboBox öffnen / schließen

PopupType Darstellung einer Pfeilschaltfläche

TabSelect Eintrag aus Populiste mit  bzw.  +  in Eingabefeld übertragen

Kontakt

ComboStyle



Art der Übernahme eines Satzes aus der Combobox Typ int

Siehe Liste, Objekte, WinPropGet(), WinPropSet()

Diese Eigenschaft bestimmt, ob ein Satz aus der Combobox per Doppel- oder Einfachklick übernommen wird.

Folgende Modi sind vorhanden:

WinComboDoubleClick Übernahme per Doppelklick (default)

WinComboSingleClick Übernahme per Einfachklick

WinComboIgnoreClick Liste nicht mit Mausklick schließen

WinComboTrackNoSelect Eintrag nicht mit Mauszeiger selektieren

WinComboHeaderStatic Anzeige der Spaltenüberschriften

Die Ausprägungen WinComboTrackNoSelect und WinComboHeaderStatic können mit den anderen Ausprägungen kombiniert werden.

Kontakt

MaxLines Anzahl

der Zeilen Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Diese Eigenschaft bestimmt, mit welcher Anzahl Zeilen das Objekt DataListPopup dargestellt wird. Eine DataListPopup wird standardmäßig mit maximal 10 Zeilen dargestellt. Sollen beispielsweise in einer DataListPopup die Monate eines Jahres aufgelistet werden, sind ohne zu Scrollen die Monate Januar - Oktober sichtbar. Wird die Eigenschaft MaxLines auf 12 gesetzt, wird die DataListPopup mit 12 Zeilen dargestellt und somit die Monate Januar - Dezember sichtbar. Der zulässige Wert liegt zwischen 1 und 100.

PopupOpen

ComboBox öffnen / schließen

Typ logic

Siehe Liste, Objekte

Über diese Eigenschaft kann das Popup des Eingabeobjekts prozedural geöffnet oder geschlossen werden. Wird dieser Eigenschaft der Wert true zugewiesen, wird die ComboBox geöffnet, bei der Angabe von false geschlossen.



Die Eigenschaft kann nicht im Designer gesetzt werden.

PopupType

Darstellung einer Pfeilschaltfläche

Typ [int](#)

Siehe [Liste](#), [Objekte](#)

Diese Eigenschaft bestimmt, ob eine Pfeilschaltfläche dargestellt wird und dieses Objekt als Combobox verwendet werden kann.

Folgende Modi sind vorhanden:

- [_WinPopupNone \(0\)](#)

Darstellung ohne Pfeilschaltfläche. Es steht keine Combobox zur Verfügung.

- [_WinPopupList \(1\)](#)

Darstellung einer Pfeilschaltfläche. Es steht ein [PopupList](#)-Objekt zur Verfügung. In dieses Objekt kann ein [RecListPopup](#)- oder [DataListPopup](#)-Objekt eingefügt werden.

- [_WinPopupListAuto \(3\)](#)

Diese Modus entspricht [_WinPopupList](#). Die Combobox wird aber bereits bei der ersten Tastatureingabe oder beim Einfügen aus der Zwischenablage geöffnet. Ist die Eigenschaft [ReadOnly](#) beim Objekt gesetzt, wird die Liste nicht automatisch geöffnet.

Kontakt

TabSelect



Aktuellen Eintrag des Popups in das Eingabefeld übertragen Typ logic

Siehe Liste, Objekte, WinPropGet(), WinPropSet()

Ist die Eigenschaft gesetzt, wird der aktuell im Popup ausgewählte Eintrag in das Eingabefeld übertragen, wenn das Feld mit oder verlassen wird.

Kontakt

Special (Objektspezifische Eigenschaften)

Eigenschaften

sortiert nach

Gruppen,

Siehe Alphabetische

Liste aller

Eigenschaften

AdVersion

Versionsnummer von Acrobat Distiller

AlignTab

Ausrichtung der Registerkartentitel

ArgInt

Array der hinterlegten Argument-Werte

ArgInt64

Array der hinterlegten Argument-Werte (bigint) _____

ArgIntCount

Anzahl der Argument-Werte

Attributes

Drucker-Attribute

AutoWrap

Kein horizontales Scrollen

BinSource

Druckerschacht

BinSourceName

Name des Druckerschachts

BorderType

Rahmentyp einer Column, SubColumn, eines Items oder SubItems

BorderWidth

Rahmenbreite in Pixel

BrushType

Schraffur

CellActiveHorz

Zellkoordinate für horizontale Zellmarkierung

CellActiveVert

Zellkoordinate für vertikale Zellmarkierung

CellCountHorz

Anzahl der horizontalen Zellen

CellCountVert

Anzahl der vertikalen Zellen

CellOfsHorz

Horizontaler Offset

CellOfsVert

Vertikaler Offset

CellSizeHorz

Breite der Zelle

CellSizeMode

Aufteilung des vorhandenen Platzes auf die Zellen

CellSizeVert

Höhe der Zelle

ClmAlign

Ausrichtung der Spalte

ClmFixed

Spalte fixieren

ClmOrder

Spaltenposition

ClmPopup

Anzeige im Edit-Objekt

ClmSortFlags

Optionen des Spalten-Objekts

ClmSortImage

Bild im Spaltenkopf

ClmStretch

Automatisches Anpassen der Spaltenbreite

ClmStretchWidth

Vorherige Spaltenbreite

ClmType

Typ der Spalte

ClmTypeImage

Anzeige als Bild

ClmWidth

Spaltenbreite

Collate

Collate-Modus

ColTheme

Darstellung der Image-Tiles

Comment

Kurzbeschreibung des Druckers

Kontakt

<u>ConnectionString</u>	ODBC-Verbindungszeichenkette
<u>ContentHeight</u>	Höhe des Inhalts einer Gruppe
<u>ContentHeightMax</u>	Maximale Höhe des Inhalts einer Zeile
<u>ContentLeft</u>	Linke Position des Inhalts einer Gruppe
<u>ContentSource</u>	Quelle des darzustellenden Inhaltes
<u>ContentTop</u>	Obere Position des Inhalts einer Gruppe
<u>ContentType</u>	Typ des Inhalts
<u>ContentWidth</u>	Breite des Inhalts einer Gruppe
<u>Copies</u>	Anzahl der Druck-Kopien
<u>CorrectionLevel</u>	Fehlertoleranzlevel bei QR-Barcodes
<u>Current</u>	Ausgewähltes Objekt
<u>CurrentInt</u>	Selektierter Knoten / selektierte Spalte
<u>Data</u>	Verweis auf Daten
<u>DataOwner</u>	Eigentümer der Daten
<u>DateWindow</u>	Bereichsangabe für Jahrhundert
<u>DbaName</u>	Name der Datenbank
<u>DbaUserID</u>	User-ID des Datenbankbenutzers
<u>DbaUserName</u>	User-Name des Datenbankbenutzers
<u>Decimals</u>	Anzahl der Nachkommastellen
<u>DefaultDate</u>	Standardwert des Eingabeobjektes
<u>Dictionaries</u>	Ausgegewählte Wörterbücher für Rechtschreibprüfung
<u>DragSource</u>	Deskriptor des Quell-Objekts
<u>DrawBkgAlways</u>	Hintergrundfarbe des Items immer anzeigen
<u>DropFiles</u>	Drag & Drop von Dateien unterbinden
<u>DropPlace</u>	Position der einzufügenden Daten
<u>Duplex</u>	Duplex-Modus
<u>EditMask</u>	Eingabemaske
<u>EmfFileName</u>	Dateiname des EMF-Dokuments
<u>Erg</u>	Ergebnis der letzten Datensatzoperation
<u>ExpireTime</u>	Zeit bis die Gruppe aus dem Cache entfernt wird
<u>FileCount</u>	Anzahl der selektierten Dateien
<u>FileName</u>	Name der Datei
<u>FileNameExt</u>	Standardmäßige Dateiendung
<u>FmtBigIntFlags</u>	Formatierungsattribute für bigint-Werte
<u>FmtBoolString</u>	Formatierungsattribut für logic-Werte
<u>FmtBoolStyle</u>	Formatierungsattribute für logic-Werte
<u>FmtDateString</u>	Formatierungsattribute für date-Werte
<u>FmtDateStyle</u>	Formatierungsattribute für date-Werte
<u>FmtDecimalFlags</u>	Formatierungsattribute für decimal-Werte
<u>FmtFloatFlags</u>	Formatierungsattribute für float-Werte
<u>FmtIntFlags</u>	Formatierungsattribute für int-Werte
<u>FmtOutput</u>	Ausgabeformatierung

Kontakt

<u>FmtPostComma</u>	Anzahl der Nachkommastellen
<u>FmtTimeFlags</u>	Formatierungsattribute für time-Werte
<u>FocusByMouse</u>	Setzen des Fokus beim Mausklick
<u>Format</u>	Format der Daten
<u>FormatDate</u>	Formatierung der Ausgabe
<u>FormatEnum</u>	Datenformate bei Drag & Drop-Operationen
<u>FormatTime</u>	Formatierung der Ausgabe
<u>FormatWrap</u>	Automatischen Umbruch speichern
<u>FormType</u>	Form eines CanvasGraphic-Objektes im Canvas-Objekt
<u>Frame</u>	Darstellungsart des Rahmens
<u>FrameEnd</u>	Nummer des letzten Bildes
<u>FrameStart</u>	Nummer des ersten Bildes
<u>GanttFlags</u>	Optionen
<u>GrayingModeUser</u>	Darstellung ausgegrauter Symbole
<u>GridHeight</u>	Vertikaler Abstand zwischen Rasterpunkten
<u>GridWidth</u>	Horizontaler Abstand zwischen Rasterpunkten
<u>Group</u>	Gruppe
<u>GroupDrawSelect</u>	Selektionsrahmen zeichnen
<u>GsEncryption</u>	Verschlüsselungstyp von PDF-Dokumenten
<u>GsOptions</u>	Optionen für GhostScript
<u>GsPasswordOwner</u>	Passwort des Eigentümers
<u>GsPasswordUser</u>	Passwort des Benutzers
<u>GsPdfColorProfile</u>	Farbprofil für PDF/A-Dokumente
<u>GsPermissions</u>	Rechtebeschränkung für PDF-Dokumente
<u>GsVersion</u>	Versionsnummer von GhostScript
<u>HeightIvl</u>	Höhe des Intervalls
<u>HeightMode</u>	Höhenmodus
<u>HyphMarginTwips</u>	Bereich ohne Silbentrennung
<u>ID</u>	ID des Objektes
<u>ImageSource</u>	Quelle des Bildes
<u>IndexColBkg</u>	Hintergrundfarbe für untergeordnete Objekte
<u>IndexColFg</u>	Vordergrundfarbe für untergeordnete Objekte
<u>IndexData</u>	Darstellungsindex des Eltern-Objektes
<u>IndexFont</u>	Schriftattribute für untergeordnete Objekte
<u>InputCheck</u>	Überprüfung der Eingabe
<u>InputMode</u>	Eingabemodus
<u>InstallCtxOffice</u>	Installation der Office-Erweiterung
<u>InstallPathCommon</u>	Suchpfade für gemeinsame Komponenten
<u>JobCount</u>	Druckjob-Anzahl
<u>JobPriority</u>	Druckjob-Priorität
<u>Justify</u>	Ausrichtung, wenn das Objekt fokussiert ist
<u>JustifyVert</u>	Angabe der vertikalen Textausrichtung

Kontakt

<u>JustifyView</u>	Ausrichtung, wenn das Objekt nicht fokussiert ist
<u>LangDisplay</u>	Sprache von Standard-Dialogen einstellen
<u>LeaveAtEnd</u>	Verlassen am Ende
<u>LengthMax</u>	Eingabelänge des Objektes
<u>LineCapEnd</u>	Pfeiltyp für den Endpunkt von Linien im <u>Canvas-Objekt</u>
<u>LineCapStart</u>	Pfeiltyp für den Startpunkt von Linien im <u>Canvas-Objekt</u>
<u>LineJustify</u>	Ausrichtung von Linien im <u>Canyas-Objekt</u>
<u>LineWidth</u>	Breite des <u>CanvasGraphic-Objektes</u> Linie im <u>Canvas-Objekt</u>
<u>LinkText</u>	Aktion, die beim Anklicken auszuführen ist
<u>Locale</u>	Länderspezifische Darstellung zuweisen
<u>Location</u>	Drucker-Standort
<u>LstFlags</u>	Optionen
<u>LstStyle</u>	Darstellung der Spaltentitel
<u>MarginLeft</u>	Linker Randabstand
<u>MaxBigInt</u>	Maximaler Eingabewert des Objektes
<u>MaxDate</u>	Maximaler Eingabewert des Objektes
<u>MaxDecimal</u>	Maximaler Eingabewert des Objektes
<u>MaxFloat</u>	Maximaler Eingabewert des Objektes
<u>MaxInt</u>	Maximaler Eingabewert des Objektes
<u>MaxTime</u>	Maximaler Eingabewert des Objektes
<u>MemObjHandle</u>	Deskriptor auf ein Memory-Objekt
<u>MinBigInt</u>	Minimaler Eingabewert des Objektes
<u>MinDate</u>	Minimaler Eingabewert des Objektes
<u>MinDecimal</u>	Minimaler Eingabewert des Objektes
<u>MinFloat</u>	Minimaler Eingabewert des Objektes
<u>MinInt</u>	Minimaler Eingabewert des Objektes
<u>MinTime</u>	Minimaler Eingabewert des Objektes
<u>MobileMode</u>	Unterstützung auf mobilen Geräten
<u>ModeColor</u>	Eingabebereich des ColorEdit
<u>ModeDraw</u>	Art der Darstellung
<u>ModeEffect</u>	Drehung des Bildes oder Dokuments
<u>ModeOptimize</u>	Optimierte Anzeige von TIFF-Formaten
<u>ModeZoom</u>	Art der Darstellung
<u>ModuleUnit</u>	Breite des schmalen Elements im Barcode
<u>NameComputer</u>	Rechnername
<u>NameDriver</u>	Name des Drucker-Treibers
<u>NamePort</u>	Name des Drucker-Ports
<u>NamePrinter</u>	Name des Druckers
<u>NameServer</u>	Name des Servers
<u>NameShare</u>	Share-Name des Druckers
<u>NameUser</u>	Name des Benutzers ermitteln
<u>NodeDynamic</u>	Dynamischer Knoten

Kontakt

<u>NodeExpanded</u>	Knoten offen / geschlossen
<u>NodeStyle</u>	Knoten-Symbole
<u>NotifyParent</u>	Funktion übergeben
<u>ObjLink</u>	Verweis auf ein Objekt
<u>OffsetLeft</u>	Linke Startposition
<u>OffsetTop</u>	Obere Startposition
<u>Opacity</u>	Deckkraft des Bildes
<u>OrderPass</u>	Reihenfolge der Notizbuchseiten
<u>PageCount</u>	Anzahl der Seiten
<u>PageNumClm</u>	Anzahl der Spalten für Seitenanzeige
<u>PageFrom</u>	Startseite der Druckausgabe
<u>PageTo</u>	Endeseite der Druckausgabe
<u>PageZoom</u>	Seitenzoom
<u>PathName</u>	Verzeichnis
<u>PdfAuthor</u>	Autor des PDF-Dokumentes
<u>PdfCompression</u>	Kompression des PDF-Dokuments
<u>PdfCreator</u>	Ersteller des PDF-Dokumentes
<u>PdfEncryptKeyLen</u>	Verschlüsselung des PDF-Dokuments
<u>PdfFileName</u>	Dateiname des PDF-Dokumentes
<u>PdfImageResolution</u>	Auflösung der Grafiken
<u>PdfJpegQuality</u>	Qualität der JPEG-Komprimierung
<u>PdfMode</u>	PDF/A-Dokument erstellen
<u>PdfPasswdOpen</u>	Password zum Öffnen des PDF-Dokuments
<u>PdfPasswdOwner</u>	Password des Eigentümers des PDF-Dokuments
<u>PdfRestriction</u>	Einschränkungen der Funktionen im PDF-Dokument
<u>PdfTitle</u>	Titel des PDF-Dokumentes
<u>PicBPP</u>	Farbauflösung des Bildes
<u>PicDpiDefaultX</u>	Horizontale DPI-Standardauflösung von Bildern
<u>PicDpiDefaultY</u>	Vertikale DPI-Standardauflösung von Bildern
<u>PicDpiExifX</u>	Horizontale DPI-Auflösung des Bildes
<u>PicDpiExifY</u>	Vertikale DPI-Auflösung des Bildes
<u>PicDpiHeadX</u>	Horizontale DPI-Auflösung des Bildes
<u>PicDpiHeadY</u>	Vertikale DPI-Auflösung des Bildes
<u>PicDpiX</u>	Horizontale DPI-Auflösung des Bildes
<u>PicDpiY</u>	Vertikale DPI-Auflösung des Bildes
<u>PicHeight</u>	Höhe des Bildes
<u>PicHeightLog</u>	Höhe des Bildes in logischen Einheiten
<u>PictureMode</u>	Art der Darstellung
<u>PictureName</u>	Hintergrundbild
<u>PicWidth</u>	Breite des Bildes
<u>PicWidthLog</u>	Breite des Bildes in logischen Einheiten
<u>PopupFlags</u>	Anzeige-Optionen der Font-Liste

Kontakt

<u>PopupLineHeight</u>	Zeilenhöhe der Drop-Down-Liste
<u>PopupWidth</u>	Breite der Drop-Down-Liste
<u>PreventOverlap</u>	Überschneidung von Intervallen
<u>PrinterDefault</u>	Standard-Drucker
<u>PrinterList</u>	Druckerliste
<u>PrintMode</u>	Druckmodus des Textes
<u>PrintOptions</u>	Druckoptionen
<u>PrintToFile</u>	In eine Datei drucken
<u>Priority</u>	Drucker-Priorität
<u>ProgressMax</u>	Maximaler Wert des Fortschrittsbalkens
<u>ProgressPos</u>	Position des Fortschrittsbalkens
<u>PrtDevice</u>	Drucker-Ausgabeeinheit
<u>PrtRtfFlags</u>	Optionen der Verarbeitung
<u>QualityX</u>	Horizontale Druckauflösung
<u>QualityY</u>	Vertikale Druckauflösung
<u>Radius</u>	Radius der abgerundeten Ecken eines Grafikobjektes im <u>Canvas</u> -Objekt
<u>Range</u>	markierter Bereich
<u>Repeat</u>	Abspielen des Films wiederholen
<u>Rotation</u>	Drehung des Bildes
<u>RtfEditFlags</u>	Optionen der Darstellung
<u>RtfMixMarker</u>	Markierungszeichen für Platzhalter
<u>Ruler</u>	Linealanzeige
<u>SBarStyle</u>	Position der Scrollbar
<u>ScalaLabelCount</u>	Definiert die Anzahl der Beschriftungen auf der Achsenkala
<u>ScalaLabels</u>	Definiert die Beschriftung der Achsenkala
<u>ScalaTextJustify</u>	Ausrichtung der Beschriftung für die Achsenkala
<u>ScalaTextOrigin</u>	Anordnung der Skalen
<u>SearchEnabled</u>	Suche im <u>TreeView</u> aktivieren
<u>SearchFlags</u>	Optionen für die Suche im <u>TreeNode</u>
<u>SearchPattern</u>	Suchtext zur Suche im <u>TreeView</u>
<u>SelData</u>	Deskriptor für <u>SelectionData</u> -Objekt
<u>SelectedVisible</u>	Selektierte Zeile im sichtbaren Bereich
<u>SelectPrt</u>	Seitenauswahl
<u>SelectPvw</u>	Seitenauswahl
<u>ShapeType</u>	Ausrichtung
<u>ShowArrow</u>	Darstellung eines Arrow-Buttons
<u>ShowBorder</u>	Anzeige eines Randes
<u>ShowFocus</u>	Zeichnen des Fokusrechtecks
<u>ShowGrip</u>	Darstellung des Ziehpunktes einer Toolbar
<u>ShowText</u>	Textausgabe in den Intervallen

Kontakt

<u>SignColor</u>	Farbe für Zeichnungsmodus
<u>SignMode</u>	Zeichnungsmodus aktivieren
<u>SignWidth</u>	Liniendicke für Zeichnungsmodus
<u>Silent</u>	Anzeige von Meldungen unterdrücken
<u>Size</u>	Größe / Breite des Objektes
<u>SizeScala</u>	Größe der Skala
<u>SkipPrint</u>	Bedingungen zum Drucken
<u>SoftLineCount</u>	Anzahl der Zeilen
<u>SpellIgnoreAllCapsWords</u>	Wörter, die nur aus Großbuchstaben bestehen, ignorieren
<u>SpellSplitWords</u>	Zusammengesetzte Wörter für Rechtschreibprüfung trennen
<u>SplitFlags</u>	Optionen des GroupSplit-Objekts
<u>SplitStyle</u>	Position des Trenn-Balkens (Split bar)
<u>Status</u>	Druckerstatus
<u>StoClmType</u>	Typ der Spalte
<u>StreamSource</u>	Art der Quelle
<u>StyleButton</u>	Darstellung der Schaltfläche
<u>StyleCloseBox</u>	Aktivität der Schließen-Schaltfläche
<u>StyleConsole</u>	Funktionsumfang des Objektes
<u>StyleCorner</u>	Eckenform der Markierung
<u>StyleDisplay</u>	Darstellung des Kalenders
<u>StyleFrame</u>	Fensterstil
<u>StyleIvl</u>	Darstellung der Intervalle
<u>StylePen</u>	Linientyp der Markierung
<u>StyleRow</u>	Darstellung der Registerreiter-Zeile
<u>StyleShade</u>	Darstellungsart
<u>StyleTab</u>	Darstellung der Registerreiter
<u>SubDivisions</u>	Skalenunterteilung der Achse
<u>SubItemStretch</u>	<u>ClmStretch</u> auch bei <u>SubItems</u> berücksichtigen
<u>TaskBarButton</u>	Anzeige des Fensters in der Taskleiste des Betriebssystems
<u>ThemeBaseName</u>	Name des verwendeten Basis-Themes
<u>ThemeBkg</u>	Darstellung des Hintergrundes der Werkzeugleiste
<u>Themed</u>	Darstellung des Hintergrundes der Titelzeile von <u>GroupSplit-</u> und <u>GroupTile-</u> Objekten
<u>TickAlignment</u>	Ausrichtung der Markierungen des Objekts
<u>TickInterval</u>	Ausrichtung der Markierungen des Objekts
<u>TickLineSize</u>	Anzahl der Positionen, um die der Schieberegler mit den Pfeiltasten verschoben wird
<u>TickPageSize</u>	Anzahl der Positionen, um die der Schieberegler mit den Tasten verschoben wird
<u>TifFileName</u>	Dateiname des TIF-Dokuments
<u>TifMode</u>	Farb-Modus bei Drucken als TIFF-Datei

Kontakt

<u>TileBorder</u>	Rahmen des GroupSplit-Objekts
<u>TileTheme</u>	Darstellung der ImageTile und NodeStyle
<u>TimeStart</u>	Startzeit der Drucker-Verfügbarkeit
<u>TimeUntil</u>	Endzeit der Drucker-Verfügbarkeit
<u>TitleJustify</u>	Ausrichtung der Beschriftung des Achsentitels
<u>TitleText</u>	Titel des Objektes
<u>Toolbars</u>	Anzeige von Toolbars
<u>TrayFrame</u>	TrayFrame-Applikation zuweisen
<u>TreeFlags</u>	Darstellung des TreeView-Objektes
<u>TypeBarcode</u>	Typ des zu druckenden Barcodes
<u>TypeButton</u>	Aktion der Schaltfläche
<u>TypeFile</u>	Filmtyp
<u>TypeIcon</u>	Typ des Symbols
<u>Unicode</u>	Objekt mit Unicode-Unterstützung
<u>Unit</u>	Maßeinheit für Lineal
<u>Url</u>	Aktuell angezeigte Adresse des WebNavigators
<u>Version</u>	Version des Druckertreibers
<u>Vertical</u>	Vertikale Darstellung des Balkens
<u>VerticalText</u>	Vertikale Darstellung
<u>ViewId</u>	Nummer des Views
<u>ViewMode</u>	Anzeigemodus
<u>ViewType</u>	Bedingungen zur Anzeige von Overlays
<u>WaitCursor</u>	Mauszeiger als Sanduhr darstellen
<u>WidthPen</u>	Strichstärke
<u>WordBreak</u>	Automatischer Umbruch bei zu langem Text
<u>XmlFileName</u>	Dateiname des XML-Dokumentes
<u>ZoomEffective</u>	Vergrößerungsfaktor
<u>ZoomFactor</u>	Benutzerdefinierter Vergrößerungsfaktor
<u>ZOrder</u>	Darstellungsreihenfolge

Kontakt

Special (Objektspezifische Eigenschaften) - A bis E

Eigenschaften sortiert

nach Gruppen,

Alphabetische Liste

Siehe aller Eigenschaften,

Liste aller

Special-Eigenschaften

AdVersion Versionsnummer von Acrobat Distiller

AlignTab Ausrichtung der Registerkartentitel

ArgInt Array der hinterlegten Argument-Werte

ArgInt64 Array der hinterlegten Argument-Werte (bigint)

ArgIntCount Anzahl der Argument-Werte

Attributes Drucker-Attribute

AutoWrap Kein horizontales Scrollen

BinSource Druckerschacht

BinSourceName Name des Druckerschachts

BorderType Rahmentyp einer Column, SubColumn, eines Items oder SubItems

BorderWidth Rahmenbreite in Pixel

BrushType Schraffur

CellActiveHorz Zellkoordinate für horizontale Zellmarkierung

CellActiveVert Zellkoordinate für vertikale Zellmarkierung

CellCountHorz Anzahl der horizontalen Zellen

CellCountVert Anzahl der vertikalen Zellen

CellOfsHorz Horizontaler Offset

CellOfsVert Vertikaler Offset

CellSizeHorz Breite der Zelle

CellSizeMode Aufteilung des vorhandenen Platzes auf die Zellen

CellSizeVert Höhe der Zelle

CImAlign Ausrichtung der Spalte

CImFixed Spalten fixieren

CImOrder Spaltenposition

CImPopup Anzeige im Edit-Objekt

CImSortFlags Optionen des Spalten-Objekts

CImSortImage Bild im Spaltenkopf

CImStretch Automatisches Anpassen der Spaltenbreite

CImStretchWidth Vorherige Spaltenbreite

CImType Typ der Spalte

CImTypeImage Anzeige als Bild

CImWidth Spaltenbreite

Collate Collate-Modus

ColTheme Darstellung der Image-Tiles

Comment Kurzbeschreibung des Druckers

ConnectionString ODBC-Verbindungszeichenkette

Kontakt

<u>ContentHeight</u>	Höhe des Inhalts einer Gruppe
<u>ContentHeightMax</u>	Maximale Höhe des Inhalts einer Zeile
<u>ContentLeft</u>	Linke Position des Inhalts einer Gruppe
<u>ContentSource</u>	Quelle des darzustellenden Inhaltes
<u>ContentTop</u>	Obere Position des Inhalts einer Gruppe
<u>ContentType</u>	Typ des Inhalts
<u>ContentWidth</u>	Breite des Inhalts einer Gruppe
<u>Copies</u>	Anzahl der Druck-Kopien
<u>CorrectionLevel</u>	Fehlertoleranzlevel bei QR-Barcodes
<u>Current</u>	Ausgewähltes Objekt
<u>CurrentInt</u>	Selektierter Knoten / selektierte Spalte
<u>Data</u>	Verweis auf Daten
<u>DataOwner</u>	Eigentümer der Daten
<u>DateWindow</u>	Bereichsangabe für Jahrhundert
<u>DbaName</u>	Name der Datenbank
<u>DbaUserID</u>	User-ID des Datenbankbenutzers
<u>DbaUserName</u>	User-Name des Datenbankbenutzers
<u>Decimals</u>	Anzahl der Nachkommastellen
<u>DefaultDate</u>	Standardwert des Eingabeobjektes
<u>Dictionaries</u>	Ausgegewählte Wörterbücher für Rechtschreibprüfung
<u>DragSource</u>	Deskriptor des Quell-Objekts
<u>DrawBkgAlways</u>	Hintergrundfarbe des Items <u>immer anzeigen</u>
<u>DropFiles</u>	Drag & Drop von Dateien unterbinden
<u>DropPlace</u>	Position der einzufügenden Daten
<u>Duplex</u>	Duplex-Modus
<u>EditMask</u>	Eingabemaske
<u>EmfFileName</u>	Dateiname des EMF-Dokuments
<u>Erg</u>	Ergebnis der letzten Datensatzoperation
<u>ExpireTime</u>	Zeit bis die Gruppe aus dem Cache entfernt wird

Kontakt

AdVersion



Versionsnummer von Acrobat Distiller

Typ alpha

Siehe Liste, Objekte, PrtPropGet()

Diese Eigenschaft enthält die Versionsnummer vom Acrobat Distiller. Ist der Distiller nicht installiert, enthält die Eigenschaft den Wert ". Die Eigenschaft kann nur gelesen werden.

Kontakt

AlignTab



Ausrichtung der Registerkartentitel

Typ int
Siehe Liste, Objekte, WinPropGet(),
WinPropSet()

Diese Eigenschaft bestimmt die Ausrichtung der Titel der Registerkarten.

Es gibt folgende Ausrichtungen:

WinAlignTabTop - Registerkartentitel oben

WinAlignTabBottom - Registerkartentitel unten

WinAlignTabLeft - Registerkartentitel links

WinAlignTabRight - Registerkartentitel rechts

Kontakt

ArgInt



Array der hinterlegten Argument-Werte

Typ int[]

Siehe Liste, Objekte, ArgInt64,

WinPropGet(), WinPropSet()

In dieser Eigenschaft werden abhängig vom Ziel der Drag & Drop-Operation zusätzliche Argumente übergeben. Ist in der Eigenschaft DropPlace der Wert WinDropPlaceNone gesetzt, werden keine Argumente übergeben und das Ereignis EvtDrop nicht aufgerufen. Folgende Argumente können übergeben werden:

Ziel-Objekt	<u>ArgIntCount</u>	ArgInt	Bedeutung
<u>Icon</u>	2	0	horizontale Position (Pixel)
		1	vertikale Position (Pixel)
<u>Picture</u>	2	0	horizontale Position (Pixel)
		1	vertikale Position (Pixel)
<u>Animation</u>	2	0	horizontale Position (Pixel)
		1	vertikale Position (Pixel)
<u>Einzelige</u>	1	0	Zeichenposition
<u>Eingabe-Objekte</u>			
<u>TextEdit</u>	3	0	Zeichenposition
		1	Zeilennummer
		2	Spaltennummer
<u>RecList</u> , <u>RecListPopup</u>	2	0	Record-ID
		1	Spaltendeskriptor
<u>DataList</u> , <u>DataListPopup</u>	2	0	Zeilennummer
		1	Spaltendeskriptor
<u>TreeView</u>	1	0	Deskriptor des Knotens
<u>GanttGraph</u>	3	0	horizontale Zellkoordinate
		1	vertikale Zellkoordinate
		2	Deskriptor des Intervalls an der Drop-Position
<u>DocView</u>	3	0	horizontale Position (Pixel)
		1	vertikale Position (Pixel)
		2	Seitennummer oder 0, falls an der Drop-Position keine Seite vorhanden ist
<u>RecView</u>	3	0	<u>Group</u> -Deskriptor, sofern vorhanden. <u>SelectorItem</u> und <u>SelectorSubItem</u> sind entsprechend gesetzt.
		1	<u>ViewId</u>
		2	<u>Record-ID</u>
<u>Button</u>	2	0	horizontale Position
		1	vertikale Position

Kontakt

-  Das Objekt **RtfEdit** verarbeitet keine dynamischen Drag & Drop-Operationen. Das Drag & Drop wird vollständig vom Objekt verarbeitet.
-  Die Eigenschaft liefert bei Werten außerhalb des Bereichs vom Datentyp **int** den Laufzeitfehler **ErrValueOverflow**, wenn sie ausgelesen wird. An Stellen, an denen Werte außerhalb des Bereichs erwartet werden (z. B. **Record-ID** beim **RecView**-Objekt) sollte statt dessen die Eigenschaft **ArgInt64** abgefragt werden.

Kontakt

ArgInt64



Array der hinterlegten Argument-Werte

Typ [bigint\[\]](#)

Siehe [Liste](#), [Objekte](#), [ArgInt](#),

[WinPropGet\(\)](#), [WinPropSet\(\)](#)

In dieser Eigenschaft werden abhängig vom Ziel der Drag & Drop-Operation zusätzliche Argumente übergeben. Ist in der Eigenschaft [DropPlace](#) der Wert [WinDropPlaceNone](#) gesetzt, werden keine Argumente übergeben und das Ereignis [EvtDrop](#) nicht aufgerufen. Folgende Argumente können übergeben werden:

Ziel-Objekt	ArgIntCount	ArgInt64	Bedeutung
Icon	2	0	horizontale Position (Pixel)
		1	vertikale Position (Pixel)
Picture	2	0	horizontale Position (Pixel)
		1	vertikale Position (Pixel)
Animation	2	0	horizontale Position (Pixel)
		1	vertikale Position (Pixel)
Einzelige	1	0	Zeichenposition
Eingabe-Objekte			
TextEdit	3	0	Zeichenposition
		1	Zeilennummer
		2	Spaltennummer
RecList , RecListPopup	2	0	Record-ID
		1	Spaltendeskriptor
DataList , DataListPopup	2	0	Zeilennummer
		1	Spaltendeskriptor
TreeView	1	0	Deskriptor des Knotens
GanttGraph	3	0	horizontale Zellkoordinate
		1	vertikale Zellkoordinate
		2	Deskriptor des Intervalls an der Drop-Position
DocView	3	0	horizontale Position (Pixel)
		1	vertikale Position (Pixel)
		2	Seitennummer oder 0, falls an der Drop-Position keine Seite vorhanden ist
RecView	3	0	Group-Deskriptor, sofern vorhanden. SelectorItem und SelectorSubItem sind entsprechend gesetzt.
		1	ViewId
		2	Record-ID
Button	2	0	horizontale Position
		1	vertikale Position

Kontakt



Das Objekt RtfEdit verarbeitet keine dynamischen Drag & Drop-Operationen. Das Drag & Drop wird vollständig vom Objekt verarbeitet.

ArgIntCount

Anzahl der Argument-Werte

Typ int

Liste, Objekte, ArgInt,

Siehe ArgInt64, WinPropGet(),

WinPropSet()

In dieser Eigenschaft ist die Anzahl der Argumente hinterlegt, die in der Eigenschaft ArgInt bzw. ArgInt64 übergeben werden. Die Anzahl der Argumente ist vom Ziel der Drag & Drop-Operation abhängig. Enthält die Eigenschaft DropPlace den Wert WinDropPlaceNone ist diese Eigenschaft leer (0) und ArgInt bzw. ArgInt64 enthält keine Argumente.

Kontakt

Attributes Drucker-

Attribute Typ int

Liste,
Siehe Objekte,

PrtPropGet()

Diese Eigenschaft ermittelt Attribute zu einem Printer-Objekt.

<u>PrtAttribDefault</u>	Standard-Drucker
<u>PrtAttribDirect</u>	Direct-Status
<u>PrtAttribEnableBidi</u>	Bidirektionale Kommunikation
<u>PrtAttribFax</u>	Faxdrucker
<u>PrtAttribHidden</u>	Hidden-Status
<u>PrtAttribLocal</u>	Lokaler Drucker
<u>PrtAttribNetwork</u>	Netzwerkdrucker
<u>PrtAttribPublished</u>	Published-Status
<u>PrtAttribQueued</u>	Queued-Status
<u>PrtAttribRawOnly</u>	Raw-Datenformat
<u>PrtAttribShared</u>	Verteilte Nutzung
PrtAttribWorkOffline	Druckerverbindung

Die Attribute können wie folgt abgefragt werden:

```
if (tPrinter->ppAttributes & _PrtAttribDefault > 0) { // Drucker ist als Standarddrucker definiert
```

AutoWrap

Kein horizontales Scrollen

Typ logic

Liste, Objekte,

Siehe WinPropGet(),
WinPropSet()

Ist diese Eigenschaft gesetzt, wird nicht horizontal gescrollt, wenn bei der Eingabe der rechte Rand des Objektes erreicht ist. Anstelle dessen wird der Text in der nächsten Zeile fortgesetzt. Dadurch kann verhindert werden, dass der Benutzer Eingaben tätigt die über den sichtbaren Bereich des Eingabeobjektes hinausreichen.

In Verbindung mit der Eigenschaft ScrollbarVisible existieren nachfolgende

Abhängigkeiten:

AutoWrap ScrollbarVisible Scrollbar Hor./Vert. Scrollen horizontal

<u>false</u>	<u>false</u>	Nein/Nein	Ja
<u>true</u>	<u>false</u>	Nein/Nein	Nein
<u>false</u>	<u>true</u>	Ja/Ja	Ja
<u>true</u>	<u>true</u>	Nein/Ja	Nein

 Wird die Eigenschaft AutoWrap oder ScrollbarVisible prozedural verändert und ist dem Objekt ein Textpuffer zugeordnet (Eigenschaft DbTextBuf), gehen eventuell getätigte Texteingaben des Anwenders verloren. Sollen diese erhalten bleiben, muss vor dem Setzen der genannten Eigenschaften der Befehl WinUpdate() mit dem Parameter WinUpdObj2Buf aufgerufen werden.

Kontakt

BinSource



Druckerschacht

Typ int

Liste, Objekte,

Siehe BinSourceName,

PrtPropGet(),

PrtPropSet()

Setzt oder liest die Nummer des Druckerschachtes, der zur Druckausgabe benutzt werden soll.
Die Nummer (ID) des Druckerschachtes und die Anzahl der Druckerschächte eines Druckers
kann mit dem Befehl PrtInfo() ermittelt werden.



Das Setzen der Eigenschaft ändert ebenfalls die Eigenschaft BinSourceName.

BinSourceName

Name des Druckerschachts

Typ alpha(8192)

Liste, Objekte,

Siehe BinSource,

PrtPropGet(),

PrtPropSet()

Die Eigenschaft legt den Papierschacht des Druckers (PrintDevice-Objekt) fest. Beim Objekt PpcObject kann der Papierschacht ermittelt werden, auf den der Ausdruck erfolgte. In diesem Fall kann die Eigenschaft nur gelesen werden.



Das Setzen der Eigenschaft ändert ebenfalls die Eigenschaft BinSource.

Der Name kann zuvor mit dem Befehl PrtInfoStr() ermittelt werden. Die Namen der Schächte sind im entsprechenden Windows-Druckertreiber hinterlegt. Je nach Version und Sprache des Druckertreibers können unterschiedliche Namen vorliegen. Gerade in einer mehrsprachigen Umgebung sollten die Namen über den Befehl PrtInfoStr() ermittelt werden.

Der Name des Papierschachtes kann mit Wildcards (*) und (?) angegeben werden.

Beispiele:

Fach 1 Setzt den Papierschacht "Fach 1" bei diesem Drucker.

Manuell Setzt das Format auf das erste Format, das die Zeichenkette "Manuell" beinhaltet.



BorderType

Rahmentyp einer Column, SubColumn, eines Items, SubItems oder eines RecView-Objektes

Typ int

Siehe Liste, Objekte

Diese Eigenschaft bestimmt den Rahmentyp einer Column, SubColumn, eines Items oder SubItems.

Sie kann folgende Werte annehmen:

[WinBorderTypeNone](#)

Es wird kein Rahmen um das Objekt gezeichnet.

[WinBorderTypeDot](#)

Es wird ein gepunkteter Rahmen um das Objekt gezeichnet.

[WinBorderTypeDotLeft](#)

Es wird ein gepunkteter Rahmen an der linken Seite des Objekts gezeichnet.

[WinBorderTypeDotTop](#)

Es wird ein gepunkteter Rahmen an der oberen Seite des Objekts gezeichnet.

[WinBorderTypeDotRight](#)

Es wird ein gepunkteter Rahmen an der rechten Seite des Objekts gezeichnet.

[WinBorderTypeDotBottom](#)

Es wird ein gepunkteter Rahmen an der unteren Seite des Objekts gezeichnet.

[WinBorderTypeDotRightBottom](#)

Es wird ein gepunkteter Rahmen an der rechten und unteren Seite des Objekts gezeichnet.
Es wird ein durchgängiger Rahmen um das Objekt gezeichnet.

[WinBorderTypeSolid](#)

Es wird ein durchgängiger Rahmen an der linken Seite des Objekts gezeichnet.

[WinBorderTypeSolidLeft](#)

Es wird ein durchgängiger Rahmen an der oberen Seite des Objekts gezeichnet.

[WinBorderTypeSolidTop](#)

Es wird ein durchgängiger Rahmen an der rechten Seite des Objekts gezeichnet.

[WinBorderTypeSolidRight](#)

Es wird ein durchgängiger Rahmen an der unteren Seite des Objekts gezeichnet.

[WinBorderTypeSolidBottom](#)

Es wird ein durchgängiger Rahmen an der rechten und unteren Seite des Objekts gezeichnet.

[WinBorderTypeSolidRightBottom](#) Es wird ein durchgängiger Rahmen an der rechten und unteren Seite des Objekts gezeichnet.

Wird die Eigenschaft beim RecView-Objekt gesetzt, erhalten alle Spalten und Gruppen diesen Rahmentyp. Durch Setzen der Eigenschaft auf

[WinBorderTypeSolidRightBottom](#) bzw. [WinBorderTypeDotRightBottom](#) lässt sich ein Raster (vergleichbar zur RecList) zeichnen.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle [WinPropGet\(\)](#) und [WinPropSet\(\)](#) gelesen bzw. geändert werden.

BorderWidth

Rahmenbreite in Pixel

Typ [int](#)

Siehe [Liste, Objekte](#)

Diese Eigenschaft bestimmt die Breite des Rahmens. Sie kann Werte zwischen 0 und 8 annehmen. Bei 0 wird kein Rahmen gezeichnet. Wird ein Wert außerhalb dieses Bereichs angegeben, wird der Laufzeitfehler [ErrValueInvalid](#) ausgelöst.

Wird die Eigenschaft beim [RecView](#)-Objekt gesetzt, erhalten alle [Spalten](#) und [Gruppen](#) diese Rahmenbreite.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle [WinPropGet\(\)](#) und [WinPropSet\(\)](#) gelesen bzw. geändert werden.

Kontakt

BrushType



Schraffur

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Über diese Eigenschaft wird festgelegt, welche Optik die Schraffur annehmen soll.

Es gibt folgende Einstellungen:

WinHatchNone / PrtHatchNone

Keine Schraffur

WinHatchBDiag / PrtHatchBDiag

diagonal von unten nach oben gehende
Schraffur

WinHatchFDiag / PrtHatchFDiag

diagonal von oben nach unten gehende
Schraffur

WinHatchHorz / PrtHatchHorz

horizontal verlaufende Schraffur

WinHatchVert / PrtHatchVert

vertikal verlaufende Schraffur

WinHatchCross / PrtHatchCross

Gitter-Schraffur

WinHatchDCross / PrtHatchDCross

Kreuz-Schraffur

Kontakt

CellActiveHorz Zellkoordinate für horizontale
Zellmarkierung Typ int

Siehe
Liste, Objekte, CellActiveVert, ColCellActive,
WinPropGet(), WinPropSet()

Legt die Zellkoordinate für die horizontale Zellmarkierung (aktuelle Spalte) fest. Bei einem Wert < 0 wird die horizontale Zellmarkierung nicht gezeichnet.

Die Farbe der Markierung wird in der Eigenschaft ColCellActive angegeben.

Kontakt

CellActiveVert Zellkoordinate für vertikale

Zellmarkierung Typ int

Liste, Objekte, CellActiveHorz,

Siehe ColCellActive, WinPropGet(),

WinPropSet()

Legt die Zellkoordinate für die vertikale Zellmarkierung (aktuelle Zeile) fest. Bei einem Wert < 0 wird die vertikale Zellmarkierung nicht gezeichnet.

Die Farbe der Markierung wird in der Eigenschaft ColCellActive angegeben.

CellCountHorz

Anzahl der horizontalen Zellen

Typ int
Siehe [Liste](#), [Objekte](#), [WinPropGet\(\)](#),

Diese Eigenschaft definiert die maximale Anzahl von Zellen auf der Abszisse. Der Wert kann jederzeit vergrößert werden. Eine Verkleinerung des Wertes ist jedoch nur möglich, wenn sich keine Intervalle außerhalb des eingestellten Wertes befinden.

Kontakt

CellCountVert



Anzahl der vertikalen Zellen

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Diese Eigenschaft definiert die maximale Anzahl von Zellen auf der Ordinate. Der Wert kann jederzeit vergrößert werden. Eine Verkleinerung des Wertes ist jedoch nur möglich, wenn sich keine Intervalle außerhalb des eingestellten Wertes befinden.

CellOfsHorz

Horizontaler Offset

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Positioniert den Ausgabebereich in horizontaler Richtung auf die in der Eigenschaft angegebene Zellkoordinate.

Wird die Eigenschaft beim GanttGraph-Objekt ausgelesen, wird nur die zuletzt gesetzte Position zurückgegeben. Die zur Zeit angezeigte Position kann über die Eigenschaft beim View-Objekt ermittelt werden.

CellOfsVert

Vertikaler Offset

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Positioniert den Ausgabebereich in vertikaler Richtung auf die in der Eigenschaft angegebene Zellkoordinate.

Wird die Eigenschaft beim GanttGraph-Objekt ausgelesen, wird nur die zuletzt gesetzte Position zurückgegeben. Die zur Zeit angezeigte Position kann über die Eigenschaft beim View-Objekt ermittelt werden.

Kontakt

CellSizeHorz

Breite der Zelle

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In dieser Eigenschaft wird die Breite der Zelle in Pixel angegeben.

Kontakt

CellSizeMode



Aufteilung des vorhandenen Platzes auf die Zellen Typ int

Siehe Liste, Objekte, PrtPropGet(), PrtPropSet()

In dieser Eigenschaft wird bestimmt, wie der für die Zellen zur Verfügung stehende Platz aufgeteilt wird.

Folgende Werte können gesetzt werden:

- PrtCellSizeModeNone

Ist dieser Wert gesetzt, definiert AreaWidthCells und AreaHeightCells Anzahl der Zellen in horizontaler bzw. vertikaler Richtung die der Ausgabebereich des PrtGanttGraph-Objektes umfassen soll. Die Größe des PrtGanttGraph-Objektes wird entsprechend angepasst.

Das Setzen der Eigenschaften CellSizeHorz bzw. CellSizeVert in diesem Modus führt zu einer Veränderung der Werte in AreaWidthCells bzw. AreaHeightCells.

- PrtCellSizeModeStretch

Ist dieser Wert gesetzt, wird die Zellgröße (CellSizeHorz / CellSizeVert) so angepasst, dass die in AreaWidthCells und AreaHeightCells angegebene Anzahl Zellen im PrtGanttGraph dargestellt werden. Das Verändern der Größe des PrtGanttGraph-Objektes in diesem Modus führt somit auch zu einer Anpassung der Zellgröße.

Die Eigenschaften CellSizeHorz und CellSizeVert sind nur lesbar.

Kontakt

CellSizeVert

Höhe der Zelle

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In dieser Eigenschaft wird die Höhe der Zelle in Pixel angegeben.

Kontakt

ClmAlign



Ausrichtung der Spalte

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Mit dieser Eigenschaft wird die Ausrichtung der Spalteninhalte und des Spaltentitels festgelegt.

Folgende Attribute können vergeben werden:

WinClmAlignLeft linksbündige Darstellung

WinClmAlignCenter zentrierte Darstellung

WinClmAlignRight rechtsbündige Darstellung

ClmFixed

Spalte fixieren

Typ int
Liste,

Siehe Objekte

Mit dieser Eigenschaft können einzelne Spalten fixiert werden. Fixierte Spalten werden innerhalb der Liste am linken Rand angezeigt und können durch einen horizontalen Scrollbar nicht verschoben werden. Die Eigenschaft kann auf folgende Werte gesetzt werden:

- _WinClmFixedNone (0)

Die Spalte ist nicht fixiert (default).

- _WinClmFixedLeft (1)

Die Spalte ist fixiert und wird auf der linken Seite angezeigt.

Ist eine Spalte fixiert wird sie zwar optisch an den linken Rand verschoben, hinsichtlich ihrer Position in der Liste der Spalten ändert sich jedoch nichts. Wird z. B. die dritte Spalte fixiert, wird diese als erste Spalte angezeigt. In der Liste steht sie jedoch nach wie vor an Position drei.

Eine fixierte Spalte kann nicht gleichzeitig die Eigenschaft ClmStretch gesetzt haben. Umgekehrt kann eine solche Spalte auch nicht fixiert sein. Das Setzen der Eigenschaft bewirkt das Ausschalten der jeweils anderen Eigenschaft.

Mit der Eigenschaft ColSeparator im zugehörigen Listen-Objekt kann die Farbe eingestellt werden, in der der Separator - welcher fixierte und nicht fixierte Spalten trennt - gezeichnet wird.

Kontakt

ClmOrder



Spaltenposition

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Mit dieser Eigenschaft wird die Position der Spalte festgelegt.

Die Eigenschaft kann im Editor oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

ClmPopup



Anzeige im Edit-Objekt

Typ logic

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Wird diese Eigenschaft auf true gesetzt, wird der Inhalt der Spalte der selektierten Zeile in das Edit-Objekt der Combobox übernommen.

ClmSortFlags

Optionen des Spalten-Objekts

Typ [int](#)

Siehe [Liste](#), [Objekte](#)

In dieser Eigenschaft können folgende Ausprägungen gesetzt werden:

- [WinClmSortFlagsAutoActive](#)

Die Einstellungen der Liste bezüglich der Sortierung werden ignoriert. Statt dessen werden die entsprechenden Eintragungen bei der Spalte verwendet, wenn der Spaltenkopf angeklickt wird. Die Ausprägung muss bei allen Spalten gesetzt werden, nach denen der Benutzer sortieren können soll.

- [WinClmSortFlagsAutoSelected](#)

Die Sortierung findet automatisch nach den Einstellungen in dieser Spalte statt. Es kann nur eine Spalte diese Ausprägung gesetzt haben. Zur Laufzeit kann der Benutzer diese Ausprägung setzen, indem er mit der Maus den Spaltenkopf anklickt. Die Ausprägung kann auch prozedural gesetzt werden.

Bei einem [DataList](#)-Objekt wird die Sortierreihenfolge, die beim ersten Anklicken der Spalte verwendet wird, durch die Eigenschaft [ClmSortImage](#) bestimmt. Bei einem [RecList](#)-Objekt bestimmen die Eigenschaften [DbKeyNo](#) und [DbKeyReverseSort](#) der Spalte die ursprüngliche Sortierung.

Kontakt

ClmSortImage



Bild im Spaltenkopf

Typ int

Liste, Objekte,

Siehe WinPropGet(),
WinPropSet()

Mit dieser Eigenschaft wird die Anzeige eines Bildes im Spaltenkopf festgelegt.

Folgende Einstellungen können vergeben werden:

WinClmSortImageNone kein Bild

WinClmSortImageUp Dreieck nach oben

WinClmSortImageDown Dreieck nach unten

WinClmSortImageKey Schlüssel

Erfolgt innerhalb eines DataList-Objekts eine automatische Sortierung (vgl. ClmSortFlags), wird mit dem Bild angegeben, in welche Richtung die Zeilen beim ersten Anklicken der Spalte sortiert werden soll.



Mit dem Setzen der ClmSortFlags (auch durch Anklicken des Spaltenkopfes) wird unabhängig von dieser Eigenschaft ein Schlüsselsymbol im Spaltenkopf gezeichnet. Die Eigenschaft wird dadurch nicht verändert.

Ist als Bild WinClmSortImageDown angegeben, wird beim ersten Klicken auf den Spaltenkopf dieses Bild angezeigt und die Zeilen in absteigender Reihenfolge sortiert. Bei der Angabe von WinClmSortImageUp erfolgt die Sortierung in aufsteigender Reihenfolge.

Durch erneutes Anklicken der Spalte, nach der bereits sortiert ist, wird die Reihenfolge umgedreht.

Kontakt

ClmStretch



Automatisches Anpassen der Breite des Objektes

Typ logic

Siehe Liste, Objekte

- RecList-/DataList-Spalte

Ist diese Eigenschaft gesetzt, wird die Spalte so verbreitert, dass die gesamte Breite des RecList- bzw. DataList-Objektes ausgenutzt wird. Wird diese Eigenschaft gesetzt, wird automatisch die Eigenschaft ClmFixed auf den Wert _WinClmFixedNone gesetzt.

- RecView-Spalte

Ist diese Eigenschaft gesetzt, wird die Spalte so verbreitert, dass die gesamte Breite des RecView-Objektes ausgenutzt wird. Die Eigenschaft hat nur Auswirkung auf SubColumns, wenn beim RecView-Objekt die Eigenschaft SubItemStretch gesetzt ist. Ist das RecView gesplittet und es werde zwei Views nebeneinander oder vier Views dargestellt, hat die Eigenschaft ebenfalls keine Auswirkung.

- Statusbar-Button

Ist diese Eigenschaft gesetzt, wird der Button der Breite der Statusbar angepasst. Diese Eigenschaft ist exklusiv, kann also jeweils nur bei einem einzigen Button pro Statusbar gesetzt werden.

Kontakt

ClmStretchWidth Vorherige

Spaltenbreite Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In dieser Eigenschaft wird die ursprüngliche Breite einer Spalte eingetragen. Wird die Eigenschaft ClmStretch auf false gesetzt, kann diese Eigenschaft ausgelesen und die ursprüngliche Spaltenbreite wieder hergestellt werden.

Kontakt

ClmType

Typ der Spalte

Typ int

Liste,

Siehe Objekte,

ClmTypeSort

Mit dieser Eigenschaft wird der Typ der Spalte festgelegt. Der Typ richtet sich nach dem Datentyp der in der Spalte aufgenommen werden soll.

Die Eigenschaft kann zur Laufzeit unter folgenden Bedingungen verändert werden:

- die DataList oder RecList ist leer oder
- die Spalte wurde noch keiner Liste mit WinAdd() hinzugefügt.

Folgende Typen sind möglich:

<u>TypeAlpha</u>	Daten vom Typ <u>alpha</u>
<u>TypeBool</u>	Daten vom Typ <u>logic</u>
<u>TypeByte</u>	Daten vom Typ <u>byte</u>
<u>TypeInt</u>	Daten vom Typ <u>int</u>
<u>TypeBigInt</u>	Daten vom Typ <u>bigint</u>
<u>TypeFloat</u>	Daten vom Typ <u>float</u>
<u>TypeDecimal</u>	Daten vom Typ <u>decimal</u>
<u>TypeDate</u>	Daten vom Typ <u>date</u>
<u>TypeTime</u>	Daten vom Typ <u>time</u>
<u>TypeWord</u>	Daten vom Typ <u>word</u>

Kontakt

ClmTypeImage

Anzeige als Bild

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Bei einem DataList-Objekt wird die Eigenschaft nur dann ausgewertet, wenn der Spaltentyp (ClmType) die Ausprägung TypeInt oder TypeAlpha besitzt. Bei TypeInt definiert der Zellinhalt die zu verwendende Kachelnummer, bei TypeAlpha den Namen des in der Datenbank vorliegenden Symbols.

Bei einem RecList-Objekt muss das der Spalte zugeordnete Datenbankfeld vom Typ "ganzzahlig lang", "ganzzahlig kurz" oder "alphanumerisch" sein. Wie bei der DataList wird auch hier das darzustellende Bild vom Zellinhalt definiert.

ClmTypeImage kann folgende Werte annehmen:

WinClmTypeImageNone keine Grafik-Ausgabe (default)

WinClmTypeImageC16 der Inhalt der Spalte wird als Nummer eines vordefinierten CONCEPT 16-Tiles interpretiert.

WinClmTypeImageUser der Inhalt der Spalte wird als Nummer eines benutzerdefinierten Tiles interpretiert.

Die Größe der angezeigten Bilder kann bei dem Listen- oder dem App-Objekt in der Eigenschaft TileSize definiert werden.

Wird diese Eigenschaft im Ereignis EvtLstDataInit gesetzt, wirkt sich das nur auf die aktuelle Zelle aus. Die Eigenschaft wird nicht für die gesamte Spalte geändert.

ClmTypeSort

Typ der Spalte zur Sortierung

Typ int

Siehe Liste, Objekte, ClmType

Mit dieser Eigenschaft wird der Typ der Spalte für die Sortierung festgelegt.

Die Eigenschaft kann zur Laufzeit unter folgenden Bedingungen verändert werden:

- die DataList ist leer oder
- die Spalte wurde noch keiner Liste mit WinAdd() hinzugefügt.

Folgende Typen sind möglich:

<u>TypeNone</u>	Sortierung der Liste nach den angezeigten Daten (Standardwert)
<u>TypeAlpha</u>	Sortierdaten vom Typ alpha
<u>TypeBool</u>	Sortierdaten vom Typ logic
<u>TypeByte</u>	Sortierdaten vom Typ byte
<u>TypeInt</u>	Sortierdaten vom Typ int
<u>TypeBigInt</u>	Sortierdaten vom Typ bigint
<u>TypeFloat</u>	Sortierdaten vom Typ float
<u>TypeDecimal</u>	Sortierdaten vom Typ decimal
<u>TypeDate</u>	Sortierdaten vom Typ date
<u>TypeTime</u>	Sortierdaten vom Typ time
<u>TypeWord</u>	Sortierdaten vom Typ word



Ist diese Eigenschaft gesetzt (ClmTypeSort != TypeNone) muss den Zeilen der Spalte mit WinLstCellSet() ein Sortierwert zugewiesen werden, da sonst das Resultat der Sortierung nicht definiert ist.

Kontakt

ClmWidth

Spaltenbreite

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Mit dieser Eigenschaft wird die Breite der Spalte festgelegt. Die Angabe erfolgt in Pixeln.

Kontakt

Collate

Collate-Modus

Typ int
Liste,

Siehe Objekte,

PrtPropGet(),
PrtPropSet()

Über diese Eigenschaft kann beim Ausdruck mehrerer Exemplare die Sortierung angegeben werden. Folgende Konstanten können gesetzt werden:

PrtCollateOn Kopien sortieren



PrtCollateOff Kopien nicht sortieren

Die Anzahl der Kopien kann in der Eigenschaft Copies gesetzt werden.

Kontakt

ColTheme



Darstellung der Image-Tiles

Typ int

Liste, Objekte,

Siehe WinPropGet(),
WinPropSet()

Über diese Eigenschaft wird die Darstellung der Image-Tiles gesteuert.

Es gibt folgende Einstellungen:

WinColorMonoGrayed Die Image-Tiles werden im Zustand "Disabled" in Grau gezeichnet.

WinColorTileSystem In hochauflösenden Grafik-Modi (> 256 Farben) werden Image-Tiles mit 256 Farben verwendet, ansonsten Image-Tiles mit 16 Farben.

WinColorTileC16Low Es werden die CONZEPT 16 definierten Grafik-Tiles mit 16 Farben verwendet. Bei benutzerdefinierten Tiles gilt das Verhalten von WinColorTileSystem.

WinColorTileUserLow Bei benutzerdefinierten Image-Tiles werden die mit 16 Farben verwendet. Bei den in CONZEPT 16 definierten Image-Tiles gilt das Verhalten von WinColorTileSystem.

WinColorTileLow Es werden generell die Image-Tiles mit 16 Farben verwendet.



Ist die Eigenschaft ColTheme auf WinColorTileLow oder WinColorTileC16Low gesetzt, wird die Eigenschaft TileTheme nicht ausgewertet.

Kontakt

Comment

Kommentar

Typ alpha(8192)

Liste,

Siehe Objekte,

PrtPropGet(),

PrtPropSet()

Diese Eigenschaft enthält eine Kurzbeschreibung des Druckers.

Die Eigenschaft kann über den Befehl PrtPropGet() gelesen werden.

Kontakt

ConnectionString

ODBC-Verbindungszeichenkette

Typ alpha

Liste, WinComOdbc,

Siehe

In dieser Eigenschaft wird die ODBC-Verbindungszeichenkette abgelegt, die durch den Auswahldialog (siehe WinComOdbc) erstellt wurde. Die Eigenschaft kann vor der Anzeige des Dialogs vorbelegt werden. Die einzelnen Parameter in der Zeichenkette müssen durch Semikolon voneinander getrennt werden. Die Zeichenkette setzt sich aus folgenden Parametern zusammen:

DSN Name der Datenquelle die voreingestellt bzw. ausgewählt wird.

DBQ Name der Datenbank die voreingestellt bzw. ausgewählt wird.

UID Name des Benutzers der voreingestellt bzw. ausgewählt wird.

PWD Kennwort des Benutzers der voreingestellt bzw. ausgewählt wird.

Die Argumente sind abhängig vom verwendeten ODBC-Treiber.

Beispiel:

```
tDlg->wpConnectionString # 'DSN=CodeLibrary;UID=user;DBQ=C:\DB\CodeLibrary';
```

ContentHeight

Höhe des Inhalts einer Gruppe

Typ int

Liste, Objekte,

WinPropGet(),

Siehe WinPropSet(),

ContentWidth, ContentLeft,

ContentTop

Mit dieser Eigenschaft kann die Höhe des Inhalts Gruppe Zeile ermittelt werden.

Die Eigenschaft kann nur im Ereignis EvtLstGroupArrange über die Befehle

WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

ContentHeightMax

Maximale Höhe des Inhalts einer Zeile

Typ int

Siehe Liste, Objekte, WinPropGet(),

Mit dieser Eigenschaft wird die maximale Höhe des Inhalts einer Zeile festgelegt.

Die Eigenschaft kann im Editor oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Mögliche Werte liegen im Bereich von **-65535 bis 65535**.

Werte < 0 Feste Höhe in Pixeln.

Wert = 0 Keine Höhenbegrenzung.

Werte > 0 Maximale Höhe in Pixeln. Verbraucht eine Zeile weniger Platz,
wird nur der benötigte Platz verwendet.

ContentLeft

Linke Position des Inhalts einer Gruppe

Typ int

Liste, Objekte, WinPropGet(),

Siehe WinPropSet(), ContentWidth,

ContentHeight, ContentTop

Mit dieser Eigenschaft wird die linke Position des Inhalts relativ zu der Gruppe festgelegt.

Die Eigenschaft kann nur im Ereignis EvtLstGroupArrange über die Befehle

WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Bei SubItems kann die Eigenschaft verändert werden. Bei Items ist sie nur lesbar.

Kontakt

ContentSource

Quelle des darzustellenden Inhaltes

Typ int

Siehe Liste, Objekte

Diese Eigenschaft definiert die Quelle des darzustellenden Inhaltes. Sie kann folgende Werte annehmen:

WinContentSourceField

Der Inhalt der Eigenschaft Caption oder der Inhalt des in DbFieldName angegebenen Feldes wird angezeigt.

WinContentSourceFileName Die Datei in der Eigenschaft FileName wird dargestellt. Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

ContentTop Obere Position des Inhalts einer
Gruppe Typ int

Liste, Objekte, WinPropGet(),
Siehe WinPropSet(), ContentWidth,
ContentHeight, ContentLeft

Mit dieser Eigenschaft wird die obere Position des Inhalts relativ zu der Gruppe festgelegt.

Die Eigenschaft kann nur im Ereignis EvtLstGroupArrange über die Befehle
WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Bei SubItems kann die Eigenschaft verändert werden. Bei Items ist sie nur lesbar.

Kontakt

ContentType Typ

des Inhalts Typ

int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Mit dieser Eigenschaft wird der Typ des Inhalts der Spalte festgelegt.

Die Eigenschaft kann im Editor oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Folgende Typen stehen zur Verfügung:

WinContentTypeText Der Inhalt wird als Plain-Text angezeigt.

WinContentTypeRTF Der Inhalt wird als RTF-Text interpretiert und angezeigt.

WinContentTypeImage Der Inhalt ist ein Bild.

ContentWidth

Breite des Inhalts einer Gruppe

Typ int

Liste, Objekte,

WinPropGet(),

Siehe WinPropSet(),

ContentHeight, ContentLeft,

ContentTop

Mit dieser Eigenschaft kann die Breite des Inhalts Gruppe Zeile ermittelt werden.

Die Eigenschaft kann nur im Ereignis EvtLstGroupArrange über die Befehle

WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Bei SubItems kann die Eigenschaft verändert werden. Bei Items ist sie nur lesbar.

Kontakt

Copies

--	--

Druck-Kopie Anzahl

Typ int

Siehe Liste, Objekte

Die Eigenschaft kann bei folgenden Objekten gesetzt werden:

- PrintDevice

Hier wird die Anzahl der zu druckenden Kopien angegeben. Der Druckjob wird dann die hier eingetragene Anzahl mal gedruckt, vorausgesetzt der entsprechende Drucker unterstützt das Drucken von Kopien.

Die Sortierung, in der die Seiten ausgedruckt werden, kann über die Eigenschaft Collate bestimmt werden.

- PrintDoc

Werden innerhalb des Druckjobs Kopien erstellt, kann die Anzahl der Kopien im PrintDoc-Objekt angegeben werden. Die Gesamtzahl der Seiten wird durch die hier angegebene Anzahl der Kopien geteilt. Dies ist nur wichtig, wenn die Anzahl der Seiten in einem Druckjob ausgegeben werden soll (Eigenschaft StyleCaption ist auf PrtStyleCapPageCount gesetzt).

Kontakt

CorrectionLevel

Fehlertoleranzlevel bei QR-Barcodes

Typ int

Siehe Liste, Objekte, TypeBarcode

In dieser Eigenschaft wird der Fehlertoleranzlevel bei QR-Codes (TypeBarcode = WinBarcode2dQRCode) bezüglich beschädigter Daten definiert.



Bei anderen Werten von TypeBarcode ist diese Eigenschaft unwirksam.
Folgende Werte können angegeben werden:

0 Entspricht Level L (7% der Daten können wiederhergestellt werden) (Standard)

1 Entspricht Level M (15% der Daten können wiederhergestellt werden)

2 Entspricht Level Q (25% der Daten können wiederhergestellt werden)

3 Entspricht Level H (30% der Daten können wiederhergestellt werden) Werden Werte außerhalb der Grenzen angegeben, wird der Wert auf die nähere Grenze (< 0 => 0 / > 3 => 3) gesetzt.

Kontakt

CroDevToolsSide



Anzeige der Entwicklerwerkzeuge

Typ int

Siehe Liste, Objekte, WinPropGet(),

WinPropSet()

Die Eigenschaft definiert die Position der Entwicklerwerkzeuge im Chromium-Objekt. Über die Eigenschaft CroDevToolsSize wird die Breite bzw. Höhe der Anzeige in Pixeln definiert.

Folgende Werte sind zulässig:

WinCroDevToolsSideNone

Anzeige abschalten

WinCroDevToolsSideLeft

Anzeige am linken Rand

WinCroDevToolsSideRight

Anzeige am rechten Rand

WinCroDevToolsSideTop

Anzeige am oberen Rand

WinCroDevToolsSideBottom

Anzeige am unteren Rand

Kontakt

Objekte von CroDevToolsSide

Folgende Objekte unterstützen die Eigenschaft CroDevToolsSide

CroDevToolsSide,

Liste der

Siehe Objekte, Liste

der

Eigenschaften

- Chromium

Kontakt

CroDevToolsSize Breite bzw. Höhe der Entwicklerwerkzeuge Typ **int**

Siehe [Liste](#), [Objekte](#), [WinPropGet\(\)](#),

Die Eigenschaft definiert die Höhe bzw. Breite der Entwicklerwerkzeuge im **Chromium**-Objekt. Über die Eigenschaft [CroDevToolsSide](#) kann die Position der Werkzeuge gesetzt werden.

Ein Wert größer 0 gibt die Größe (Breite bzw. Höhe) der Entwicklerwerkzeuge in Pixeln an. Der Wert 0 schaltet die Werkzeug-Anzeige ab. Werte kleiner 0 führen zum Laufzeitfehler [ErrValueInvalid](#).

Kontakt

Objekte von CroDevToolsSize

Folgende Objekte unterstützen die Eigenschaft CroDevToolsSize

CroDevToolsSize,

Liste der

Siehe Objekte, Liste

der

Eigenschaften

- Chromium

Current

Ausgewähltes Objekt

Typ alpha(8192)

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Über diese Eigenschaft kann das ausgewählte Objekt gesetzt werden. Bei einem Notebook-Objekt kann über diese Eigenschaft zur Laufzeit der Name der aktuellen Notizbuchseite ermittelt oder gesetzt werden.

Bei Setzen der Notizbuchseite erhält diese standardmäßig den Fokus. Ist das nicht gewünscht, kann mit der Option WinFlagNoFocusSet ein Wechsel des Fokus unterbunden werden.

Beispiel:

```
$NoteBook->wpCurrent(_WinFlagNoFocusSet) # 'Page1';
```

Kontakt



CurrentInt

Selektierte(r) Knoten / Zeile / Gruppe / Seitennummer

Typ int
Siehe Liste, Objekte, WinPropGet(), WinPropSet(),
PrtPropGet(), PrtPropSet()

Je nach Objekt hat CurrentInt folgende Bedeutung:

- TreeView

Mit Hilfe dieser Eigenschaft kann der Deskriptor des aktuell selektierten Knotens ermittelt und gesetzt werden. Beim Setzen werden dazugehörige Parent-Knoten automatisch aufgeklappt.

- DataList

Über diese Eigenschaft kann eine Zeile selektiert oder die aktuelle Zeile abgefragt werden. Die erste Zeile fängt bei 1 an. Mit 0 wird die Selektion aufgehoben.

- Toolbar, Statusbar

In Verbindung mit der Eigenschaft Group, des Toolbar- bzw. Statusbar-Buttons, ist eine Unterteilung der Schaltflächen in Gruppen möglich. Über die Gruppen kann gesteuert werden, ob eine Schaltfläche angezeigt oder ausgeblendet wird.

Ist die Eigenschaft CurrentInt gleich 0 (Vorgabe), werden alle Buttons angezeigt, deren Eigenschaft Group größer oder gleich 0 ist.

Ist die Eigenschaft CurrentInt größer 0, werden alle Schaltflächen angezeigt, deren Eigenschaft Group mit CurrentInt übereinstimmt oder gleich 0 ist.

- RtfEdit

Über die Eigenschaft kann zur Laufzeit die aktuelle Zeile des RtfEdit-Objekts abgefragt werden. Der Wert liegt zwischen 1 und dem im Objekt vorhandenen Zeilen.

- PrtJobPreview, PrintJob

Über die Eigenschaft CurrentInt wird die Nummer der anzuzeigenden Seite angegeben.

- Picture, PrtPicture, Column, Item

Wird ein mehrseitiges TIFF-Bild oder Icon angezeigt, wird in dieser Eigenschaft die Seiten- bzw. Bildnummer angegeben.

- DocView

Wird ein mehrseitiges Dokument angezeigt, wird in dieser Eigenschaft die Seitennummer angegeben. Die entsprechende Seite wird in dem aktiven View angezeigt.

Data

Verweis auf Daten

Typ int

Liste, Objekte,

Siehe WinPropGet(),
WinPropSet()

Die Eigenschaft wird bei verschiedenen Objekten unterschiedlich verwendet:

Verwendung beim Objekt DragData

In der Eigenschaft befinden sich die Verweise auf unterschiedliche DragDataFormat-Objekte. Für jedes Format kann ein eigenes DragDataFormat-Objekt hinterlegt werden:

```
tDragList # aDataObject->wpData(_WinDropDataUser); tDragList # aDataObject->wpData(_WinDropDataFil
```

Beispiel: Eine Datei wird vom Desktop in die Applikation gezogen. Das DragDataFormat-Objekt wird mit der Anweisung `tFormat # aDataObject->wpData(_WinDropDataFile);` ermittelt.

Mit einem DragData-Objekt können somit bis zu fünf Datenbereiche verknüpft sein.

Verwendung beim Objekt DragDataFormat

Nachdem das DragDataFormat-Objekt ermittelt wurde, kann auf dessen Data-Eigenschaft zugegriffen werden. In dieser Eigenschaft befindet sich der Deskriptor auf die Daten. Abhängig vom Format der Daten befindet sich folgender Inhalt in der Eigenschaft:

- Benutzerdefiniertes Format (WinDropDataUser)

In der Eigenschaft befindet sich ein Deskriptor auf eine CTE-Liste oder auf einen CTE-Tree. Die Liste oder der Tree enthält frei vom Benutzer definierte Daten, die nicht interpretiert werden. Ein Transport dieser Daten ist nur zwischen CONZEPT 16-Oberflächen-Objekten möglich. Windows-Anwendungen erkennen Daten dieses Formats nicht.

- Datei (WinDropDataFile)

In der Eigenschaft befindet sich ein Deskriptor auf eine CTE-Liste. Die Liste enthält CteItem's deren Eigenschaft Name den zu übertragenden Dateinamen enthalten muss. Eine Überprüfung, ob die Datei existiert findet nicht statt. Daten in diesem Format werden z. B. vom Windows Explorer übergeben.

- Inhalt (WinDropDialogContent)

In der Eigenschaft befindet sich ein Deskriptor auf eine CTE-Liste. Die Liste enthält CteItem's deren Eigenschaft Name den Namen für den Inhalt enthält. In der Eigenschaft ID wird der Deskriptor auf ein Memory-Objekt angegeben. In dem Memory-Objekt wird der Inhalt übergeben.

- Unformatierter Text (WinDropDataText)

Kontakt

In der Eigenschaft befindet sich ein Deskriptor auf einen Textpuffer. Daten in diesem Format können beispielsweise von und nach Word, Wordpad usw. transportiert werden.

- Formaterter Text (WinDropDataRtf)

In der Eigenschaft befindet sich ein Deskriptor auf einen Textpuffer. Die Daten liegen im RTF-Format vor. Dadurch ist formatierbarer Text möglich, der beispielsweise von Word oder Wordpad entgegengenommen werden kann.

Verwendung beim Objekt SelectionData

In der Eigenschaft befinden sich Verweise auf unterschiedliche dynamische Strukturen. Die selektierten Elemente liegen in einer Cte-Liste und in einem Cte-Tree vor. Das Ermitteln der jeweiligen Struktur erfolgt über einen entsprechenden Parameter:

```
tCteData # tSelectionData->wpData(_WinSelDataCteTree); // Cte-Tree tCteData # tSelectionData->wpDa
```

Die dynamische Struktur bleibt erhalten, bis das SelectionData-Objekt entfernt wird. Das erfolgt, wenn das übergeordnete Objekt (zum Beispiel das RecList) mit WinClose() aus dem Speicher gelöscht wird.

Beispiel:

```
// Ermitteln eines Dateinamens, der in ein Objekt gezogen wurde im Ereignis EvtDroptDragDataForma
```

DataOwner

Eigentümer der Daten

Typ logic

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In dieser Eigenschaft wird definiert, ob die übertragenen Daten nach der Abwicklung der Drag & Drop-Funktionen weiter zur Verfügung stehen sollen, oder automatisch freigegeben werden.

Ist diese Eigenschaft auf true gesetzt, werden die Daten nach Durchführung der Funktionen automatisch gelöscht. Ist die Eigenschaft auf false gesetzt, müssen in der EvtDragTerm die entsprechenden Deskriptoren freigegeben werden.

Wenn Daten von externen Applikationen (auch anderer CONZEPT 16-Client) übertragen werden, ist diese Eigenschaft standardmäßig true, ansonsten false.

Kontakt

DateWindow

Bereichsangabe für Jahrhundert

Typ int

Siehe Liste, Objekte,

WinPropGet(), WinPropSet()

Diese Eigenschaft gibt eine Bereichszahl für das Jahrhundert bei 2-stelligen
Jahreingaben an. Ist das eingegebene Jahr kleiner als DateWindow wird 2000 zum
Jahr ansonsten 1900 addiert.

Bei einem Wert von beispielsweise 20 werden zweistellige Jahreingaben von 00 bis 19 als 2000 bis
2019, Werte von 20 bis 99 als 1920 bis 1999 interpretiert.

Kontakt

DbaName (Eigenschaft) —

Name der Datenbank

Typ alpha

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In dieser Eigenschaft steht der Name und der Pfad der Datenbank, aus der die Informationen bei einer Drag & Drop-Operation stammen. Wird zum Beispiel ein Datensatz aus einer Datenbank gezogen, kann über diese Eigenschaft die Herkunft ermittelt werden.

Erfolgt das Drag & Drop nicht aus einer CONZEPT 16-Datenbank, ist diese Eigenschaft leer.

DbaUserID

User-ID des Datenbankbenutzers

Typ int

Liste, Objekte,

Siehe DbaUserName,

WinPropGet(), WinPropSet()

In dieser Eigenschaft ist die User-ID des Datenbank-Benutzers abgelegt, aus der die Drag & Drop-Daten stammen. Wird zum Beispiel ein Datensatz aus einer CONZEPT 16-Applikation herausgezogen, befindet sich die User-ID des Datenbankbenutzers in dieser Eigenschaft.

Erfolgt das Drag & Drop nicht aus einer CONZEPT 16-Datenbank, ist diese Eigenschaft leer.

Kontakt

DbaUserName

User-Name des Datenbankbenutzers

Typ alpha(20)

Siehe Liste, Objekte, DbaUserID,

In dieser Eigenschaft ist der User-Name des Datenbank-Benutzers abgelegt, aus der die Drag & Drop-Daten stammen. Wird zum Beispiel ein Datensatz aus einer CONZEPT 16-Applikation herausgezogen, befindet sich der User-Name des Datenbankbenutzers in dieser Eigenschaft.

Erfolgt das Drag & Drop nicht aus einer CONZEPT 16-Datenbank, ist diese Eigenschaft leer.

Decimals

Anzahl der Nachkommastellen

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In dieser Eigenschaft wird die Anzahl der Nachkommastellen angegeben, die in dem Objekt ausgegeben bzw. eingegeben werden können. Bei der Angabe von -1 werden keine Nachkommastellen verwendet. Ist die Eigenschaft auf 0 gesetzt werden soviele Nachkommastellen ausgegeben, wie vorhanden sind.

Besitzt der Wert mehr signifikante Nachkommastellen als angegeben, findet eine kaufmännische Rundung statt.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

DefaultDate

Standardwert des Eingabeobjektes

Typ date

Liste, Objekte, MinDate,

Siehe MaxDate, WinPropGet(),

WinPropSet()

Diese Eigenschaft bestimmt den Wert, der angezeigt wird, wenn das DateEdit-Objekt keinen Inhalt (Nullwert) besitzt und der Benutzer eine Tastatureingabe macht. Der Nullwert dieser Eigenschaft ist gleichbedeutend mit dem 1.1. des aktuellen Jahres.

Soll für alle DateEdit-Objekt das aktuelle Datum als Vorgabewert verwendet werden, kann das über die Flags-Eigenschaft des App-Objekts durch Setzen der Ausprägung WinAppEditDateDefaultCurr erfolgen.

In Verbindung mit der Eigenschaft FormatDate kann eine Schnelleingabe für die Eingabe des Datums vorgenommen werden.

Kontakt

DialogBoxFlags Optionen

der Dialogbox Typ int

Liste, Objekte,
Siehe WinPropGet(),
WinPropSet()

Diese Eigenschaft bestimmt die verwendete Schriftart sowie die Beschriftung der Schaltflächen in mit WinDialogBox() angezeigten Nachrichtenfenstern.

Folgende Werte sind zulässig:

- WinDialogBoxDefault – Standard-Einstellung.
- WinDialogBoxUseFont – Legt die Schriftart für die Textdarstellung fest.
- WinDialogBoxUseFontButton – Legt die Schriftart für die Schaltflächen fest.
- WinDialogBoxUseTextButton – Legt die Beschriftung für die Schaltflächen fest.

Kontakt

Objekte von **DialogBoxFlags**

Folgende Objekte unterstützen die Eigenschaft **DialogBoxFlags**

DialogBoxFlags,

Liste der

Siehe **Objekte, Liste**

der

Eigenschaften

- **Application**

Kontakt

DialogBoxTextButton Beschriftung der Schaltflächen
der Dialogbox Typ alpha

Siehe Liste, Objekte, WinPropGet(),

Die Eigenschaft bestimmt den Text, der für die Schaltflächen in Nachrichtenfenstern (WinDialogBox()) verwendet werden soll. Der Text für die jeweilige Schaltfläche wird über die ID der Schaltfläche definiert.

Der Text wird nur berücksichtigt, wenn die Option WinDialogBoxUseTextButton der Eigenschaft DialogBoxFlags des App-Objektes gesetzt ist.

Beispiele

```
_App->wpDialogBoxTextButton(_WinIdOk) # 'Schließen'; // Text für die Ok-Schaltfläche._App->wpDial
```

Kontakt

Objekte von DialogBoxTextButton

Folgende Objekte unterstützen die Eigenschaft [DialogBoxTextButton](#)

Siehe [DialogBoxTextButton](#),
[Liste der Objekte](#),

[Liste der](#)
[Eigenschaften](#)

- [Application](#)

Kontakt

Dictionaries



Ausgewählte Wörterbücher für Rechtschreibprüfung

Typ int

Liste, Objekte, SpellSplitWords,

Siehe SpellIgnoreAllCapsWords, WinPropGet(),

WinPropSet()

Über diese Eigenschaft kann gesteuert werden, welche Wörterbücher für die Rechtschreibprüfung im CtxDocEdit-Objekt verwendet werden sollen.

Es gibt folgende Ausprägungen:

WinDictEnglishUS Wörterbuch Englisch (USA) verwenden.

WinDictEnglishUK Wörterbuch Englisch (Großbritannien) verwenden.

WinDictEnglishCA Wörterbuch Englisch (Kanada) verwenden.

WinDictEnglishTech Wörterbuch technisches Englisch verwenden.

WinDictGermanDE Wörterbuch Deutsch (Deutschland) verwenden.

Die Ausprägungen können wie folgt überprüft und gesetzt werden:

```
// Ausprägung auswerten: if ($ctxDocEdit->wpDictionaries & _WinDictGermanDE > 0){ ...}// Auspräg
```

DocMixMarker

Markierungszeichen für Platzhalter

Typ alpha(1)
Liste, Objekte, WinPropGet(),
Siehe

In dieser Eigenschaft wird das Markierungszeichen für das Text und Daten mischen bei CtxDocEdit-Objekten definiert. Als Markierungszeichen sollte ein Zeichen gewählt werden, das ausschließlich zur Markierung dient und sonst nicht im Text vorkommt.

Das Markierungszeichen muss in seiner Form auch im jeweiligen Datei-Format (also im "Source-Code" des Textes) vorkommen. Einige Sonderzeichen werden in ihre Tabellenposition im Zeichensatz übersetzt (z. B. "°" in "\b0" oder "\|" in "\\" bei RTF-Texten). Solche Zeichen können nicht als Markierungszeichen verwendet werden.

Ist kein Markierungszeichen definiert, wird die Tilde "~" als Markierungszeichen verwendet.

Die Eigenschaft kann über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

DragSource

Deskriptor des Quell-Objekts

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In dieser Eigenschaft steht der Deskriptor des Quell-Objekts bei einer Drag & Drop-Operation. Befindet sich das Quell-Objekt außerhalb der Applikation, ist die Eigenschaft leer (0).

Kontakt

DrawBkgAlways Hintergrundfarbe des Items immer
anzeigen Typ logic

Siehe [Liste](#), [Objekte](#), [WinPropGet\(\)](#),

Mit dieser Eigenschaft kann definiert werden, ob die Hintergrundfarbe (ClmColBkg) des Items auch angezeigt werden soll, wenn die Gruppe selektiert ist (true) oder nicht (false).

Standardmäßig ist die Eigenschaft auf false gesetzt.

DropFiles

Drag & Drop von Dateien unterbinden

Typ [logic](#)

Siehe [Liste, Objekte](#)

Die neue Eigenschaft **DropFiles** des [WebNavigator](#) bestimmt, ob Dateien (z. B. aus dem Explorer) per Drag & Drop im [WebNavigator](#) geöffnet werden können ([true](#)) oder nicht ([false](#)), während eine HTML-Seite angezeigt wird. Wird bereits eine HTML-Seite angezeigt, dann wirkt sich das Setzen der Eigenschaft erst nach dem Zurücksetzen und erneuten Setzen der Eigenschaft [Caption](#) aus.



Das [WebNavigator](#)-Objekt ist in der Lage unterschiedliche Inhalte (HTML, Dateisystem, Dokumente) anzuzeigen. Das Setzen der Eigenschaft **DropFiles** wirkt sich nur bei HTML-Inhalt aus.

Beispiel:

```
// Drag & Drop von Dateien verhindern:$web->wpDropFiles # false;// Seite erneut laden.tCaption #
```

DropPlace

Position der einzufügenden Daten

Typ int
Siehe Liste, Objekte, WinPropGet(),

In dieser Eigenschaft wird die Position der Daten im Ziel-Objekt bestimmt. Dabei gibt es bei verschiedenen Objekten verschiedene Möglichkeiten:

- DataPlace-Objekt

Der Inhalt der Eigenschaft kann mit dem Befehl WinPropGet() gelesen werden. Dabei kann der ausgelesene Wert mit folgenden Konstanten verglichen werden:

<u>WinDropPlaceNone</u>	keine genaue Position.
<u>WinDropPlaceThis</u>	auf dem hervorgehobenen Eintrag.
<u>WinDropPlaceBefore</u>	vor dem hervorgehobenen Eintrag.
<u>WinDropPlaceAppend</u>	nach dem hervorgehobenen Eintrag.

- Oberflächen-Objekt

Die Eigenschaft kann über die Befehle WinPropSet() und WinPropGet() gesetzt bzw. gelesen werden. Sie beeinflusst die Einfügemarkierung bei einer Drag & Drop-Operation. Folgende Konstanten stehen dabei zur Auswahl:

<u>WinDropPlaceNone</u>	Keine Einfügemarkierung anzeigen.
<u>WinDropPlaceThis</u>	Nur den Selektionsbalken für bestehende Einträge anzeigen.
<u>WinDropPlaceDynamic</u>	Selektionsbalken für bestehende Einträge und Einfügemarken zwischen den Einträgen anzeigen.

Kontakt

Duplex

Duplex-Modus

Typ int
Liste,

Siehe Objekte,
PrtPropGet(),
PrtPropSet()

Die Eigenschaft gibt an, ob der Duplex-Druck (beidseitiger Druck) gesetzt ist oder nicht.

Folgende Werte sind möglich:

- 0 - Duplex wird nicht unterstützt.
- 1 - Duplex wird unterstützt, ist jedoch ausgeschaltet.
- 2 - Duplex ist angeschaltet. Die Seite wird vertikal gedreht.
- 3 - Duplex ist angeschaltet. Die Seite wird horizontal gedreht.

EditMask

Eingabemaske

Typ alpha(8192)

Siehe Liste,

Objekte

Mit der Eingabemaske kann die Eingabe des Benutzers beschränkt werden. Dadurch ist es möglich falsche Eingaben des Benutzers schon bei der Eingabe zu unterbinden.

In der Zeichenkette kann angeben werden, wo der Benutzer an welcher Stelle welche Zeichen angeben kann.

Eine Eingabemaske kann aus einer oder mehreren Formatierungen bestehen.

Das Format kann wie folgt zusammengesetzt werden:

Format = *

***>**

%<Anzahl><Optionen><Zeichen>

Bedeutung der Bestandteile:

Wird als Eingabemaske * oder *> verwendet, können beliebige Zeichen in beliebiger Menge (bis zur in LengthMax eingetragenen Anzahl) eingegeben werden. Durch den Zusatz > werden diese Zeichen zusätzlich in Großbuchstaben gewandelt. Diese Formatierungen können nicht mit Formatierungen mit fester Länge kombiniert werden.

Formatierungen mit fester Länge werden immer mit einem % eingeleitet.

% - Beginn einer Formatierung

Anzahl - (optional; Standard = 1) Anzahl der Zeichen, die eingegeben werden können

Optionen - (optional) option specifier:

* - Löschen bei Eintritt

Zeichen - Erlaubte Zeichen n

Zahlen (0...9)

a Kleine Buchstaben (a...z) A

Große Buchstaben (A...Z)

I Alle Buchstaben (a...z, A...Z)

L Buchstaben und Zahlen (a...z, A...Z, 0...9)

! Alle Buchstaben (a...z, A...Z), die Eingabe wird in Großbuchstaben gewandelt

? Alle Zeichen

Für die erlaubten Zeichen kann auch eine Menge von Zeichen angegeben werden. Diese Menge muss in [] eingeschlossen sein und alle erlaubten Zeichen durch Komma getrennt beinhalten.

Innerhalb der Eigenschaft können mehrere Bereiche mit Eingaben fester Länge definiert werden. Zwischen den Bereichen kann ebenfalls fester Text angegeben

Kontakt

werden. Soll innerhalb des festen Textes ein Prozent-Zeichen erscheinen, muss es zweimal angegeben werden.

Beispiele:

- *>
 - Alle eingegebenen Zeichen werden von Klein- in Großbuchstaben gewandelt
- %5*n
 - Eingabebeschränkung für ein PLZ-Feld
- %3*A
 - Eingabebeschränkung für ein LKZ-Feld
- %5!-%8*n
 - Eingabebeschränkung für Artikelnummern, die sich aus Lieferantenkennung und eigener Nummer zusammensetzen (zwei Eingabebereiche)
- +%2*n(%4[,0-9])%12[,0-9,-]
 - Eingabebeschränkung für ein Telefonnummern-Feld (inklusive Festtext)

Sind in der Eingabemaske mehrere Bereiche (zum Beispiel bei der zusammengesetzten Artikelnummer weiter oben) definiert, kann durch Betätigen der

Tastenkombination  +  der Inhalt des aktuell editierten Maskensegmentes auf den Initialwert zurückgesetzt werden. Bei Betätigung ohne -Taste wird lediglich der Inhalt des Zeichens an der Cursorposition auf den Initialwert zurückgesetzt.

Kontakt

EditorAclFlags



Anzuzeigende Elemente in der Auto vervollständigungsliste des CodeEdit-Objektes Typ int

Siehe Liste, Objekte

Diese Eigenschaft definiert die anzuzeigenden Elemente in der Auto vervollständigungsliste im CodeEdit-Objekt. Standardmäßig ist diese Eigenschaft auf WinEditorAclFlagC16 | WinEditorAclFlagDataStruct gesetzt.

Folgende Werte können angegeben werden:

Konstante	Bedeutung
<u>WinEditorAclFlagNone</u>	Keine Elemente
<u>WinEditorAclFlagC16</u>	CONZEPT 16 eigene und mit <u>WinEditorKeywordsAdd()</u> hinzugefügte Schlüsselwörter
<u>WinEditorAclFlagDataStruct</u>	Elemente der Datenstruktur

Die Werte können mittels Pipe () miteinander kombiniert werden.



Diese Eigenschaft kann frühestens im EvtCreated des Elternfensters gesetzt werden.

Indizes:

Das Setzen der Eigenschaft ist prozedural über Indizes pro View des CodeEdit-Objektes möglich. Wird kein Index, oder der Index 0 angegeben, wird die Eigenschaft in allen Views verändert. Beim Ermitteln wird das erste View abgefragt. Mit den Indizes 1 bis 4 können die einzelnen Views angesprochen werden.

Beispiel:

```
// Eigenschaft bei allen Views des Objektes setzen$CodeEdit->wpEditorAclFlags # _WinEditorAclFlag
```

Kontakt

EditorAllowControlChars

Eingabe von Steuerzeichen erlauben

Typ logic

Siehe Liste, Objekte

Wenn diese Eigenschaft auf true gesetzt ist, wird im CodeEdit-Objekt die Eingabe von Steuerzeichen (ASCII-Code < 32) erlaubt. Standardmäßig ist diese Eigenschaft auf false gesetzt.



Es ist zu beachten, dass nicht alle Steuerzeichen eingegeben und angezeigt werden können.



Diese Eigenschaft kann frühestens im EvtCreated des Elternfensters gesetzt werden.

Indizes:

Das Setzen der Eigenschaft ist prozedural über Indizes pro View des CodeEdit-Objektes möglich. Wird kein Index, oder der Index 0 angegeben, wird die Eigenschaft in allen Views verändert. Beim Ermitteln wird das erste View abgefragt. Mit den Indizes 1 bis 4 können die einzelnen Views angesprochen werden.

Beispiel:

```
// Eigenschaft bei allen Views des Objektes setzen$CodeEdit->wpEditorAllowControlChars # true;//
```

Kontakt

EditorFixedHighlighting Wortvorkommen im CodeEdit-
Objekt beibehalten Typ logic

Siehe Liste, Objekte

Ist der Wert der Eigenschaft auf true gesetzt, werden die Wortvorkommen im CodeEdit-Objekt nicht gelöscht, wenn der Cursor an eine andere Stelle gesetzt wird. Standardmäßig ist diese Eigenschaft auf true gesetzt.



Diese Eigenschaft kann frühestens im EvtCreated des Elternfensters gesetzt werden.

Indizes:

Das Setzen der Eigenschaft ist prozedural über Indizes pro View des CodeEdit-Objektes möglich. Wird kein Index, oder der Index 0 angegeben, wird die Eigenschaft in allen Views verändert. Beim Ermitteln wird das erste View abgefragt. Mit den Indizes 1 bis 4 können die einzelnen Views angesprochen werden.

Beispiel:

```
// Eigenschaft bei allen Views des Objektes setzen$CodeEdit->wpEditorFixedHighlighting # true;//
```

EditorShowEOLCharacters

Zeilenende-Zeichen anzeigen

Typ [logic](#)

Siehe [Liste, Objekte](#)

Wenn diese Eigenschaft auf `true` gesetzt ist, werden im [CodeEdit](#)-Objekt Zeilenende-Zeichen angezeigt. Standardmäßig ist diese Eigenschaft auf `false` gesetzt.



Diese Eigenschaft kann frühestens im [EvtCreated](#) des Elternfensters gesetzt werden.

Indizes:

Das Setzen der Eigenschaft ist prozedural über Indizes pro View des [CodeEdit](#)-Objektes möglich. Wird kein Index, oder der Index 0 angegeben, wird die Eigenschaft in allen Views verändert. Beim Ermitteln wird das erste View abgefragt. Mit den Indizes 1 bis 4 können die einzelnen Views angesprochen werden.

Beispiel:

```
// Eigenschaft bei allen Views des Objektes setzen$CodeEdit->wpEditorShowEOLCharacters # true;//
```

Kontakt

EditorShowFlags Anzuzeigende Elemente des CodeEdit-Objektes Typ int

Siehe Liste, Objekte

Diese Eigenschaft definiert die anzuzeigenden Elemente im CodeEdit-Objekt.
Standardmäßig ist diese Eigenschaft auf WinEditorShowAnnotationBar | WinEditorShowMargin gesetzt.

Folgende Werte können angegeben werden:

Konstante	Bedeutung
<u>WinEditorShowNone</u>	Alle ausschaltbaren Elemente ausblenden
<u>WinEditorShowAnnotationBar</u>	Anmerkungsleiste anzeigen
<u>WinEditorShowMargin</u>	Informationsrand anzeigen

Die Werte können mittels Pipe () miteinander kombiniert werden.



Diese Eigenschaft kann frühestens im EvtCreated des Elternfensters gesetzt werden.

Indizes:

Das Setzen der Eigenschaft ist prozedural über Indizes pro View des CodeEdit-Objektes möglich. Wird kein Index, oder der Index 0 angegeben, wird die Eigenschaft in allen Views verändert. Beim Ermitteln wird das erste View abgefragt. Mit den Indizes 1 bis 4 können die einzelnen Views angesprochen werden.

Beispiel:

```
// Eigenschaft bei allen Views des Objektes setzen$CodeEdit->wpEditorShowFlags # _WinEditorShowAn
```

Kontakt

EditorShowLineLimit

Markierung für Zeilenlimitierung anzeigen

Typ [int](#)

Siehe [Liste, Objekte](#)

Wird diese Eigenschaft im [CodeEdit](#)-Objekt auf einen Wert größer als 0 gesetzt, wird an der entsprechenden Spalte eine vertikale Linie angezeigt, die die maximale Länge einer Zeile markiert. Standardmäßig wird das Zeilenlimit nicht dargestellt.



Diese Eigenschaft kann frühestens im [EvtCreated](#) des Elternfensters gesetzt werden.

Beispiel:

```
// Eigenschaft setzen$CodeEdit->wpEditorShowLineLimit # 80;// Eigenschaft ermittelntLineLimit # $
```

Kontakt

EditorShowLineNumbers 

Zeilennummern anzeigen

Typ [logic](#)

Siehe [Liste, Objekte](#)

Wenn diese Eigenschaft auf `true` gesetzt ist, werden im [CodeEdit](#)-Objekt

Zeilennummern angezeigt. Standardmäßig ist diese Eigenschaft gesetzt.



Diese Eigenschaft kann frühestens im [EvtCreated](#) des Elternfensters gesetzt werden.

Indizes:

Das Setzen der Eigenschaft ist prozedural über Indizes pro View des [CodeEdit](#)-Objektes möglich. Wird kein Index, oder der Index 0 angegeben, wird die Eigenschaft in allen Views verändert. Beim Ermitteln wird das erste View abgefragt. Mit den Indizes 1 bis 4 können die einzelnen Views angesprochen werden.

Beispiel:

```
// Eigenschaft bei allen Views des Objektes setzen$CodeEdit->wpEditorShowLineNumbers # true;// Ei
```

EditorShowWSCharacters

Whitespace-Zeichen anzeigen

Typ [logic](#)

Siehe [Liste, Objekte](#)

Wenn diese Eigenschaft auf true gesetzt ist, werden im [CodeEdit](#)-Objekt Leerzeichen und Tabulatorzeichen angezeigt. Standardmäßig ist diese Eigenschaft auf false gesetzt.



Diese Eigenschaft kann frühestens im [EvtCreated](#) des Elternfensters gesetzt werden.

Indizes:

Das Setzen der Eigenschaft ist prozedural über Indizes pro View des [CodeEdit](#)-Objektes möglich. Wird kein Index, oder der Index 0 angegeben, wird die Eigenschaft in allen Views verändert. Beim Ermitteln wird das erste View abgefragt. Mit den Indizes 1 bis 4 können die einzelnen Views angesprochen werden.

Beispiel:

```
// Eigenschaft bei allen Views des Objektes setzen$CodeEdit->wpEditorShowWSCharacters # true;// E
```

EditorTabSize

Länge eines Tabulators

Typ [int](#)

Siehe [Liste, Objekte](#)

Über diese Eigenschaft kann die Länge von Tabulatoren im [CodeEdit](#)-Objekt definiert werden.

Standardmäßig steht ein Tabulator für 2 Leerzeichen.



Diese Eigenschaft kann frühestens im [EvtCreated](#) des Elternfensters gesetzt werden.

Indizes:

Das Setzen der Eigenschaft ist prozedural über Indizes pro View des [CodeEdit](#)-Objektes möglich. Wird kein Index, oder der Index 0 angegeben, wird die Eigenschaft in allen Views verändert. Beim Ermitteln wird das erste View abgefragt. Mit den Indizes 1 bis 4 können die einzelnen Views angesprochen werden.

Beispiel:

```
// Eigenschaft bei allen Views des Objektes setzen$CodeEdit->wpEditorTabSize # 2;// Eigenschaft v
```

Kontakt

EditorTextModified

Objektinhalt verändert?

Typ logic

Siehe Liste, Objekte

Über diese Eigenschaft kann ermittelt werden, ob der Inhalt eines CodeEdit-Objektes verändert wurde. Diese Eigenschaft kann nur gelesen werden.

Indizes:

Das Ermitteln der Eigenschaft ist prozedural über Indizes pro View des CodeEdit-Objektes möglich. Wird kein Index, oder der Index 0 angegeben, werden alle Views abgefragt. Mit den Indizes 1 bis 4 können die einzelnen Views angesprochen werden.

Beispiel:

```
// Eigenschaft bei dem ersten Views des Objektes ermittelntModified # $CodeEdit->wpEditorTextModi
```

Kontakt

EditorTextType



Typ des dargestellten Textes im CodeEdit-Objekt Typ int

Siehe Liste, Objekte

Diese Eigenschaft definiert den Typ des angezeigten Textes im CodeEdit-Objekt.

Standardmäßig ist diese Eigenschaft auf WinEditorTextTypePrc gesetzt.

Folgende Werte können angegeben werden:

Konstante	Bedeutung
<u>WinEditorTextTypePrc</u>	Darzustellender Text ist eine Prozedur
<u>WinEditorTextTypeText</u>	Darzustellender Text ist ein Text
<u>WinEditorTextTypeXml</u>	Darzustellender Text ist ein XML-Dokument
<u>WinEditorTextTypeJson</u>	Darzustellender Text ist ein JSON-Dokument



Diese Eigenschaft kann frühestens im EvtCreated des Elternfensters gesetzt werden.

Indizes:

Das Setzen der Eigenschaft ist prozedural über Indizes pro View des CodeEdit-Objektes möglich. Wird kein Index, oder der Index 0 angegeben, wird die Eigenschaft in allen Views verändert. Beim Ermitteln wird das erste View abgefragt. Mit den Indizes 1 bis 4 können die einzelnen Views angesprochen werden.

Beispiel:

```
// Eigenschaft bei allen Views des Objektes setzen$CodeEdit->wpEditorTextType # _WinEditorTextTyp
```

Kontakt

EmfFileName

Dateiname des EMF-Dokuments

Typ alpha(8192)

Siehe Liste, Objekte,

PrtJobClose(), PrtPropSet()

Über diese Eigenschaft wird bei der Erzeugung eines EMF-Dokuments (Enhanced Metafile) der Name des zu erzeugenden Dokuments definiert. Die Eigenschaft muss angegeben werden, wenn von CONZEPT 16 aus eine EMF-Datei erzeugt werden soll.

Besteht der Druckjob aus mehreren Seiten, werden mehrere Dateien erzeugt. Die Namen der Dateien werden dann um die Seitennummer und gegebenenfalls um die Kopienummer erweitert. Der Name bleibt unverändert, wenn nur eine Seite gedruckt wird.

Beispiel

```
tHdlPrintJob->ppEmfFileName # _Sys->spPathTemp + '\rechnung.emf';tErr # tHdlPrintJob->PrtJobClose
```

Kontakt

Erg



Ergebnis der letzten Datensatzoperation

Typ int

Siehe Liste, Objekte, WinPropGet()

In dieser Eigenschaft kann das Ergebnis der letzten Datensatzoperation abgefragt werden. Die Abfrage erfolgt über den Befehl WinPropGet().

Kontakt

ExpireTime



Zeit in Sekunden, wann frühestens das nächste mal das Ereignis EvtLstGroupInit für die Gruppe aufgerufen wird.

Typ int

Siehe Liste, Objekte, WinPropGet(), WinPropSet(), EvtLstGroupInit

Die Eigenschaft kann bei einer Gruppe im Ereignis EvtLstGroupInit gesetzt werden, um zu definieren, in wie vielen Sekunden das Ereignis frühestens das nächste Mal für diese Gruppe ausgelöst wird. Dadurch kann verhindert werden, dass die Gruppe gelöscht wird, wenn die Gruppe aus dem sichtbaren Bereich austritt, weil beispielsweise der Benutzer diese aus dem sichtbaren Bereich scrollt.

Wird die Eigenschaft in dem Ereignis nicht manuell gesetzt hat sie standardmäßig einen Wert von 60 Sekunden. Wird die Eigenschaft auf 0 gesetzt, wird die Gruppe nicht gecached. Die Eigenschaft kann Werte zwischen 0 und 65535 annehmen. Bei anderen Werten wird der Laufzeitfehler ErrValueInvalid ausgelöst.



Wird die Größe einer Spalte verändert, werden alle Gruppen aus dem Cache entfernt.

Einzelne oder alle Gruppen können aus dem Cache entfernt werden, indem der Befehl WinRvwUpdate() mit dem Update-Modus WinRvwUpdateOptClearCache oder WinRvwUpdateOptClearSelected aufgerufen wird.

Kontakt

Special (Objektspezifische Eigenschaften) - F bis J

Eigenschaften sortiert

nach Gruppen,

Alphabetische Liste

Siehe aller Eigenschaften,

Liste aller

Special-Eigenschaften

<u>FileCount</u>	Anzahl der selektierten Dateien
<u>FileName</u>	Name der Datei
<u>FileNameExt</u>	Standardmäßige Dateiendung
<u>FmtBigIntFlags</u>	Formatierungsattribute für bigint-Werte
<u>FmtBoolString</u>	Formatierungsattribut für logic-Werte
<u>FmtBoolStyle</u>	Formatierungsattribute für logic-Werte
<u>FmtDateString</u>	Formatierungsattribute für date-Werte
<u>FmtDateStyle</u>	Formatierungsattribute für date-Werte
<u>FmtDecimalFlags</u>	Formatierungsattribute für decimal-Werte
<u>FmtFloatFlags</u>	Formatierungsattribute für float-Werte
<u>FmtIntFlags</u>	Formatierungsattribute für int-Werte
<u>FmtOutput</u>	Ausgabeformatierung
<u>FmtPostComma</u>	Anzahl der Nachkommastellen
<u>FmtTimeFlags</u>	Formatierungsattribute für time-Werte
<u>FocusByMouse</u>	Setzen des Fokus beim Mausklick
<u>Format</u>	Format der Daten
<u>FormatDate</u>	Formatierung der Ausgabe
<u>FormatEnum</u>	Datenformate bei Drag & Drop-Operationen
<u>FormatTime</u>	Formatierung der Ausgabe
<u>FormatWrap</u>	Automatischen Umbruch speichern
<u>FormType</u>	Form eines CanvasGraphic-Objektes im Canvas-Objekt
<u>Frame</u>	Darstellungsart des Rahmens
<u>FrameEnd</u>	Nummer des letzten Bildes
<u>FrameStart</u>	Nummer des ersten Bildes
<u>GanttFlags</u>	Optionen
<u>GrayingModeUser</u>	Darstellung ausgegrauter Symbole
<u>GridHeight</u>	Vertikaler Abstand zwischen Rasterpunkten
<u>GridWidth</u>	Horizontaler Abstand zwischen Rasterpunkten
<u>Group</u>	Gruppe
<u>GroupDrawSelect</u>	Selektionsrahmen zeichnen
<u>GsEncryption</u>	Verschlüsselungstyp von PDF-Dokumenten
<u>GsOptions</u>	Optionen für GhostScript
<u>GsPasswordOwner</u>	Passwort des Eigentümers
<u>GsPasswordUser</u>	Passwort des Benutzers
<u>GsPdfColorProfile</u>	Farbprofil für PDF/A-Dokumente
<u>GsPermissions</u>	Rechtebeschränkung für PDF-Dokumente

Kontakt

<u>GsVersion</u>	Versionsnummer von GhostScript
<u>HeightIvl</u>	Höhe des Intervalls
<u>HeightMode</u>	Höhenmodus
<u>HyphMarginTwips</u>	Bereich ohne Silbentrennung
<u>ID</u>	ID des Objektes
<u>ImageSource</u>	Quelle des Bildes
<u>IndexColBkg</u>	Hintergrundfarbe für untergeordnete Objekte
<u>IndexColFg</u>	Vordergrundfarbe für untergeordnete Objekte
<u>IndexData</u>	Darstellungsindex des Eltern-Objektes
<u>IndexFont</u>	Schriftattribute für untergeordnete Objekte
<u>InputCheck</u>	Überprüfung der Eingabe
<u>InputMode</u>	Eingabemodus
<u>InstallCtxOffice</u>	Installation der Office-Erweiterung
<u>JobCount</u>	Druckjob-Anzahl
<u>JobPriority</u>	Druckjob-Priorität
<u>Justify</u>	Ausrichtung, wenn das Objekt fokussiert ist
<u>JustifyVert</u>	Angabe der vertikalen Textausrichtung
<u>JustifyView</u>	Ausrichtung, wenn das Objekt nicht fokussiert ist

Kontakt

FileCount

Anzahl der selektierten Dateien

Typ int

Siehe Liste, Objekte,

WinPropGet(), WinPropSet()

In dieser Eigenschaft kann die Anzahl der selektierten Objekte ermittelt werden. Der hier ermittelte Wert gibt die Obergrenze für den Index zur Ermittlung der selektierten Objekte an.

Beispiel:

```
// Alle selektierten Dateinamen ermitteln
for    tCount # 1;loop  Inc(tCount);while (tCount <= tHdl
```

FileName

Name der Datei

Typ alpha(8192)

Liste, Objekte,

WinPropGet(),

Siehe WinPropSet(),

PrtPropGet(),

PrtPropSet()

In dieser Eigenschaft kann eine externe Datei, ein binäres Objekt oder ein importiertes Bild angegeben werden. Ist die Angabe einer externen Datei möglich, erscheint eine ComboBox, sobald der Fokus in die Eigenschaft gesetzt wird. Wird der Eintrag "Externe Datei..." ausgewählt, kann in einem anschließenden Dateiauswahl-Dialog die Datei gewählt werden.

Bei der Angabe einer externen Datei muss der Präfix '!' oder '*' angegeben werden, um einen relativen oder absoluten Pfad anzuzeigen. Binäre Objekte werden mit dem Präfix '>' gefolgt von der Nummer der verbundenen Datenbank, gekennzeichnet. Mit '>0' wird ein binäres Objekt aus der aktuellen Datenbank geladen. Ist kein Präfix angegeben, wird ein importiertes Bild geladen.

Je nach Objekt hat die Eigenschaft folgende Bedeutung:

- Animation

Angabe einer externen Datei.

- CtxAdobeReader und CtxOffice

In der Eigenschaft wird die externe Datei angegeben, die in dem Objekt angezeigt werden soll.

Soll im CtxOffice-Objekt ein neues Dokument angelegt werden, kann das Dokument in der ComboBox ausgewählt werden. Folgende Dokumente können angegeben werden:

Wert der Eigenschaft Anzeige in der ComboBox

'#Word.Document' Word-Dokument

'#Excel.Sheet' Excel-Sheet

'#Excel.Chart' Excel-Diagramm

'#PowerPoint.Show' PowerPoint-Präsentation

'#MSProject.Project' Project-Projekt

'#Visio.Drawing' Visio-Zeichnung

'#MapPoint.Map' MapPoint-Karte

- Icon

Angabe einer externen oder in die Datenbank importierten Datei. Bei Angabe des Namens ohne Präfix ('!' und '*') wird eine Icon-Datei in der Datenbank referenziert.

- RtfEdit und PrtRtf

Kontakt

Angabe einer externen oder in der Datenbank gespeicherten Datei.

Die Quelle des Textes wird in der Eigenschaft StreamSource spezifiziert. Bei externen Dateien darf in der Eigenschaft dem Dateinamen kein '*' vorangestellt werden.

- **DocView und PrtPdf**

Angabe einer externen Datei oder eines binären Objekts.

- **PpcObject**

Dateiname des Druckjobs.

Die Eigenschaft kann bei diesem Objekt nur gelesen werden.

- **Systemdialog "Datei öffnen"**

FileName kann als Vorgabe und Resultat verwendet werden. In der Eigenschaft steht immer der Name einer externen Datei. Es darf kein Präfix angegeben werden.

Bei Verwendung als Vorgabe wird die zu öffnende Datei im Dialog vorgeschlagen. Die vom Anwender geöffnete Datei wird über FileName ermittelt. Ist in der Eigenschaft Flags des Dialogs die Ausprägung WinComAllowMultiSelect gesetzt, können mehrere Dateien ausgewählt werden. Auf die einzelnen Dateinamen kann dann über einen Index (zum Beispiel tFileName # tHdIDlg->wpFileName(2)) zugegriffen werden. Die Anzahl der selektierten Dateien befindet sich in der Eigenschaft FileCount. Über die Indexierung können nicht mehrere Vorgabedateien angegeben werden. Der Pfad kann über die Eigenschaft PathName ermittelt werden.



Bei Verwendung des alten Windows-Dialog Stiles (Ausprägung

WinComExplorer ist in der Eigenschaft Flags nicht gesetzt) werden Dateinamen vom Dialog in der Kurzform (8.3-Form; Leerzeichen ersetzt durch "~") angezeigt. Die Eigenschaft FileName gibt in jedem Fall die Langform des Dateinamens zurück.

- **Systemdialog "Datei speichern"**

FileName kann als Vorgabe und Resultat verwendet werden. In der Eigenschaft steht immer der Name einer externen Datei. Es darf kein Präfix angegeben werden.

Bei Verwendung als Vorgabe wird die zu speichernde Datei im Dialog vorgeschlagen. Die vom Anwender gespeicherte Datei wird über FileName ermittelt. Auch in diesem Dialog ist eine Mehrfachauswahl analog zum "Datei öffnen"-Dialog möglich. Der Pfad kann über die Eigenschaft PathName ermittelt werden.



Bei Verwendung des alten Windows-Dialog Stiles (Ausprägung

WinComExplorer ist in der Eigenschaft Flags nicht gesetzt) werden Dateinamen vom Dialog in der Kurzform (8.3-Form; Leerzeichen ersetzt durch "~") angezeigt. Die Eigenschaft FileName gibt in jedem Fall die Langform des Dateinamens zurück.

- **(Sub-)Item eines RecView**

Kontakt

In (Sub-)Items eines RecView-Objektes kann je nach Wert der Eigenschaft ContentType ein Plain-, ein RTF-Text oder ein Bild angezeigt werden. Um den Inhalt aus BLObs zu lesen, muss '>0' vor den Dateinamen geschrieben werden. Soll eine externe Datei geladen werden muss '*' vor dem Pfad angegeben werden. Wurde kein Präfix verwendet, wird je nach ContentType der interne Text oder das importierte Bild verwendet.

- CtxDocEdit und CodeEdit

Angabe einer externen oder in der Datenbank gespeicherten Datei.

Bei Angabe des Namens ohne Präfix ('!' und '*') wird ein interner Text in der Datenbank referenziert.

Beispiele:

```
// externe Datei relativ angegeben '!Icons\Artikel.ico' // externe Datei absolut angegeben '*C:\C16\
```

Kontakt

FileNameExt



Standardmäßige Dateiendung.

Typ alpha(8192)

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Über diese Eigenschaft wird eine standardmäßige Dateiendung festgelegt. Wird beim Öffnen oder Speichern einer Datei keine Erweiterung angegeben, wird die in FileNameExt definierte Erweiterung verwendet.

FlagsExt

Erweiterte Optionen für Objekte

Typ [int](#)

Siehe [Liste](#), [Objekte](#)

Über diese Eigenschaft werden erweiterte Optionen für das [Application](#)-Objekt festgelegt:

[WinAppExtFlowKeySupport](#)

EvtKeyItem für Steuerungstasten auslösen.

[WinAppExtTransparentBkg](#)

Überdeckungsmodus für untergeordnete Objekte einer [GroupBox](#).

[WinAppExtLayoutComplete](#)

Aufrufzeitpunkt für EvtPosChanged.

[WinAppExtUmlautKeySupport](#)

EvtKeyItem für Umlaute auslösen.

[WinAppExtLstPosSelected](#)

Neupositionierung der Einträge in Listen bei Veränderung von Spalteneigenschaften verhindern. Parallel Druckjobs synchronisieren. Diese Option kann die Performance der Druckausgabe beeinflussen.

[WinAppExtSyncPrint](#)

Tastatureingaben während laufender Ereignisse behandeln.

[WinAppExtWaitKeyInput](#)

[WinAppExtContextMenuSelect](#) Ausführung von [EvtMenuContext](#) während [EvtLstSelect](#) bzw. [EvtNodeSelect](#) blockieren.

Kontakt

FmtBigIntFlags  Formatierungsattribute für große Integer-Werte Typ int

Siehe
Liste, Objekte, WinPropGet(), WinPropSet(),
PrtPropGet(), PrtPropSet()
Mit dieser Eigenschaft können große Integer-Werte formatiert werden. Folgende Einstellungen können vergeben werden:

- Numerische Darstellungen:
 - ◆ FmtNumNoGroup Keine Tausendertrennung
 - ◆ FmtNumNoZero Nullunterdrückung
 - ◆ FmtNumLeadZero Umwandlung mit führenden Nullen
 - ◆ FmtNumHex Darstellung in Hexadezimalform
- Datums- und Zeitdarstellungen:
 - ◆ FmtCaltimeRFC Darstellung nach RFC 2822
 - ◆ FmtCaltimeISO Darstellung nach ISO 8601
 - ◆ FmtCaltimeDHMS Darstellung als Tage, Stunden, Minuten, Sekunden und Millisekunden als Zeitdifferenz zum 01.01.1601 00:00:00.000
 - ◆ FmtCaltimeUTC Umwandlung nach UTC
 - ◆ FmtCaltimeDate Konvertierung des Datums mit 'T' als Trenner zwischen Datum und Zeit
 - ◆ FmtCaltimeDateBlank Konvertierung des Datums mit Leerzeichen als Trenner zwischen Datum und Zeit
 - ◆ FmtCaltimeTimeHMS Konvertierung der Zeit im Format "HH:mm:ss"
 - ◆ FmtCaltimeTimeHM Konvertierung der Zeit im Format "HH:mm"
 - ◆ FmtCaltimeTimeFull Konvertierung der Zeit im Format "HH:mm:ss.nnnnnnnn"

Optionen der Numerischen Darstellung können nicht mit Optionen der Datums- und Zeitdarstellung kombiniert werden.

Bei der Option FmtCaltimeISO werden die Bestandteile des formatierten Wertes durch die Kombination mit den Optionen FmtCaltimeDate(...) und FmtCaltimeTime... definiert. Aus jeder dieser zwei Gruppen kann nur eine Option mit FmtCaltimeISO kombiniert werden.

Die Option FmtCaltimeUTC kann nur in der Kombination mit einer der Konstanten FmtCaltimeRFC bzw. FmtCaltimeISO angegeben werden. Ohne die Option FmtCaltimeUTC wird die lokale Zeitzone verwendet.

Bei einem Edit-Objekt wirkt sich die Eigenschaft nur dann aus, wenn FmtOutput auf true gesetzt ist.

Die Flags können wie folgt überprüft und gesetzt werden:

```
// Flag auswerten if ($edBigInt->wpFmtBigIntFlags & _FmtNumNoGroup > 0) { ... } // Flag setzen $edBig
```

FmtBoolString

Formatierungsattribut für Bool-Werte

Typ [alpha\(8192\)](#)

[Liste](#), [Objekte](#), [WinPropGet\(\)](#),

Siehe [WinPropSet\(\)](#), [PrtPropGet\(\)](#),

[PrtPropSet\(\)](#)

Mit dieser Eigenschaft können logic-Werte formatiert werden. Der String wird durch ein Semikolon getrennt 'true;false'.

Beispiele:

```
tObj->wpFmtBoolString # 'ja;nein' // StandardeinstellungtObj->wpFmtBoolString # 'J;N'tObj->ppFmtB
```

Kontakt

FmtBoolStyle



Formatierungsattribute für logische Werte

Typ int

Siehe Liste, Objekte, WinPropGet(),

WinPropSet()

Mit dieser Eigenschaft können logic-Werte formatiert werden. Folgende Einstellungen können vergeben werden:

<u>WinFmtBoolCheck</u>	Darstellung als Checkbox	<input checked="" type="checkbox"/>
<u>WinFmtBoolCheckLight</u>	dünnere Darstellung der Checkbox	<input checked="" type="checkbox"/>
<u>WinFmtBoolCheckXp</u>	Betriebssystemabhängige Darstellung der Checkbox	<input checked="" type="checkbox"/>
<u>WinFmtBoolRadio</u>	Darstellung als Radiobox	<input checked="" type="radio"/>
<u>WinFmtBoolRadioLight</u>	dünnere Darstellung der Radiobox	<input checked="" type="radio"/>
<u>WinFmtBoolRadioXp</u>	Betriebssystemabhängige Darstellung der Radiobox	<input checked="" type="radio"/>
<u>WinFmtBoolString</u>	benutzerdefinierte Darstellung	

Wird diese Eigenschaft im Ereignis EvtLstDataInit gesetzt, wirkt sich das nur auf die aktuelle Zelle aus. Die Eigenschaft wird nicht für die gesamte Spalte geändert.

Kontakt

FmtDateString



Formatierungsattribute für Datums-Werte

Typ alpha(8192)

Liste, Objekte, WinPropGet(),

Siehe WinPropSet(), PrtPropGet(),

PrtPropSet()

Mit dieser Eigenschaft können Zeit-Werte formatiert werden.

Beispiele:

```
tObj->wpFmtDateString # 'dd.MM.yyyy'      // 15.11.2000 (Standardeinstellung) tObj->wpFmtDateString
```

Es werden die gleichen Formatierungsstrings verwendet wie bei den
Ländereinstellungen unter Windows.

Kontakt

FmtTextStyle | Formatierungsattribute für

Datums-Werte Typ int

Liste, Objekte, FormatDate,

Siehe WinPropGet(), WinPropSet(),

PrtPropGet(), PrtPropSet()

Mit dieser Eigenschaft kann das Ausgabeformat von Datums-Werten festgelegt werden. Bei Oberflächen-Objekten wird das Format angewendet, wenn das Objekt nicht den Eingabefokus besitzt. Folgende Einstellungen können vergeben werden:

WinFmtDateShort Darstellung des Datums im Kurzformat

WinFmtDateLong Darstellung des Datums im Langformat

WinFmtDateString benutzerdefinierte Darstellung



Bei einem DateEdit-Objekt wirkt sich die Eigenschaft nur dann aus, wenn FmtOutput auf true gesetzt ist.

Kontakt

FmtDecimalFlags | Formatierungsattribute für große
Dezimal-Werte Typ int

Siehe
Liste, Objekte, WinPropGet(), WinPropSet(),
PrtPropGet(), PrtPropSet()
Mit dieser Eigenschaft können große Dezimal-Werte formatiert werden. Folgende Einstellungen
können vergeben werden:

FmtNumNoZero Nullunterdrückung

FmtNumNoGroup Ohne Tausender trennung

FmtNumLeadZero Führende Nullen

FmtNumPoint Punkt als Dezimaltrennzeichen

Bei einem DecimalEdit-Objekt wirkt sich die Eigenschaft nur dann aus, wenn FmtOutput
auf true gesetzt ist.

Die Flags können wie folgt überprüft und gesetzt werden:

```
// Flag auswerten:if ($edDecimal->wpFmtDecimalFlags & _FmtNumNoGroup > 0) { . . . } // Flag setzen$ed
```

Kontakt

FmtFloatFlags Formatierungsattribute für

Gleitkomma-Werte Typ int

Siehe

Liste, Objekte, WinPropGet(), WinPropSet(),
PrtPropGet(), PrtPropSet()

Mit dieser Eigenschaft können Gleitkomma-Werte formatiert werden. Folgende Einstellungen können vergeben werden:

FmtNumNoZero Nullunterdrückung

FmtNumNoGroup Ohne Tausender trennung

FmtNumLeadZero Führende Nullen

FmtNumPoint Punkt als Dezimaltrennzeichen

Bei einem FloatEdit-Objekt wirkt sich die Eigenschaft nur dann aus, wenn FmtOutput auf true gesetzt ist.

Die Flags können wie folgt überprüft und gesetzt werden:

```
// Flag auswerten: if ($edFloat->wpFmtFloatFlags & _FmtNumNoGroup > 0) { ... } // Flag setzen $edFlo
```

Kontakt

FmtIntFlags Formatierungsattribute für Integer-Werte Typ int

Liste, Objekte, WinPropGet(),
Siehe WinPropSet(), PrtPropGet(),
PrtPropSet()

Mit dieser Eigenschaft können Integer-Werte formatiert werden. Folgende Einstellungen können vergeben werden:

FmtNumNoZero Nullunterdrückung

FmtNumNoGroup Ohne Tausendertrennung

FmtNumLeadZero Führende Nullen

FmtNumHex Ausgabe als Hexadezimalzahl

Bei einem IntEdit-Objekt wirkt sich die Eigenschaft nur dann aus, wenn FmtOutput auf true gesetzt ist.

Die Flags können wie folgt überprüft und gesetzt werden:

```
// Flag auswerten: if ($edInt->wpFmtIntFlags & _FmtNumNoGroup > 0) { ... } // Flag setzen $edInt->wpF
```

Kontakt

FmtNumLen Anzahl der Stellen mit führenden
Nullen Typ int

Liste, Objekte, WinPropGet(),
Siehe WinPropSet(), PrtPropGet(),
PrtPropSet()

In dieser Eigenschaft wird die Anzahl der Stellen eines numerischen Wertes angegeben, die in dem Objekt ausgegeben wird.



Diese Eigenschaft wirkt sich nur aus, wenn das Flag FmtNumLeadZero in der jeweiligen Eigenschaft FmtIntFlags, FmtBigIntFlags, FmtFloatFlags oder FmtDecimalFlags des Objektes gesetzt ist.

Kann die resultierende Zeichenkette nicht in den angegebenen Stellen angezeigt werden, oder wird das Ergebnis in einen zu kurzen String zurückgegeben, wird er mit '*' aufgefüllt.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet(), WinPropSet(), PrtPropGet() und PrtPropSet() gelesen bzw. geändert werden.

Kontakt

FmtOutput



Ausgabeformatierung

Typ logic

Liste, Objekte,

WinPropGet(),

Siehe WinPropSet(),

PrtPropGet(),

PrtPropSet()

Hat diese Eigenschaft den Wert true wird die Ausgabeformatierung (FmtFloatFlags, FmtIntFlags usw.) für das Objekt vorgenommen, andernfalls nicht. Die Formatierung der Ausgabe wird dann aktiviert, wenn das Eingabeobjekt nicht den Eingabefokus besitzt, andernfalls wird in den Eingabemodus umgeschaltet.

FmtPostComma

Anzahl der Nachkommastellen

Typ int

Liste, Objekte, WinPropGet(),

Siehe WinPropSet(), PrtPropGet(),

PrtPropSet()

Mit dieser Eigenschaft wird die Anzahl der Nachkommstellen festgelegt.

Die Anzahl der Nachkommastellen kann im Bereich von 0 bis 15 liegen. Bei der Angabe von -1 werden die Ländereinstellungen von Windows verwendet. Bei der Angabe von -2 werden soviele Nachkommastellen dargestellt, wie der Wert benötigt.

Kontakt

FmtTimeFlags



Formatierungsattribute für Zeit-Werte

Typ **int**

Liste, Objekte, WinPropGet(),

Siehe WinPropSet(), PrtPropGet(),
PrtPropSet()

Mit dieser Eigenschaft können Zeit-Werte formatiert werden. Folgende Einstellungen können vergeben werden:

FmtTimeHSeconds Darstellung mit Hundertstelsekunden

FmtTimeLFormat Darstellung aus Lcl**TimeLFormat**

FmtTimeNoHours Darstellung ohne Stunden

FmtTimeNoMinutes Darstellung ohne Minuten

FmtTimeNoSep Darstellung ohne Trennzeichen

FmtTimeSeconds Darstellung mit Sekunden

FmtTimeSFormat Darstellung aus Lcl**TimeSFormat**

Bei einem TimeEdit-Objekt wirkt sich die Eigenschaft nur dann aus, wenn FmtOutput auf true gesetzt ist.

Die Flags können wie folgt überprüft und gesetzt werden:

```
// Flag auswerten:if ($edTime->wpFmtTimeFlags & _FmtNumNoGroup > 0){ ...}// Flag setzen$edTime->
```

FocusByMouse

Setzen des Fokus beim Mausklick

Typ logic
Siehe Liste, Objekte, WinPropGet(),

Durch Setzen dieser Eigenschaft werden Mausklicks mit der rechten oder mittleren Maustaste gleich behandelt wie ein Mausklick mit der linken Maustaste, d. h. das Objekt bekommt den Fokus und der unter dem Mauszeiger liegende Datensatz, Zeile oder Knoten wird automatisch selektiert (das Ereignis EvtLstSelect bzw. EvtNodeSelect wird ausgelöst).



Ist die Zeile oder der Knoten bereits selektiert, erhält das Objekt nur den Fokus. Es wird nicht erneut das Ereignis EvtLstSelect bzw. EvtNodeSelect für die Zeile oder den Knoten ausgelöst.

Kontakt

FocusSelect



Inhalt des Eingabeobjektes bei Erhalt des Fokus selektieren Typ logic

Siehe Liste, Objekte, WinPropGet(), WinPropSet()

Mit dieser Eigenschaft wird definiert, ob der Inhalt des Eingabeobjektes beim Erhalt des Fokus komplett selektiert (true), oder der Cursor vor das erste Zeichen gesetzt (false) wird. Standardmäßig ist die Eigenschaft auf true gesetzt.



Ausgenommen von dieser Regel sind Edit-Objekte, bei denen eine Eingabemaske (EditMask) gesetzt ist und DateEdit- und TimeEdit-Objekte, wenn in der Eigenschaft Flags des App-Objektes WinAppEditDateFast bzw. WinAppEditTimeFast gesetzt ist.

Kontakt

Format

Format der Daten

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In dieser Eigenschaft ist das Format abgelegt, in dem die Daten bei dem Objekt hinterlegt sind. Der Wert der Eigenschaft kann mit folgenden Konstanten verglichen werden:

WinDropDataUser benutzerdefinierte Daten

WinDropDataFile Daten im Format "file"

WinDropDialogContent Daten werden mit übergeben

WinDropDataText Daten im Format "text"

WinDropDataRtf Daten im Format "rtf"

WinDropDataUnknown Daten unbekannten Formats

In der Eigenschaft Data ist der Deskriptor auf die Daten hinterlegt.

Kontakt

FormatDate



Formatierung der Ausgabe

Typ int

Liste, Objekte,
FmtDateFormat(),
Siehe WinPropGet(),
WinPropSet(),
PrtPropGet(),
PrtPropSet()

Diese Eigenschaft bestimmt die Darstellung des Datums zum Zeitpunkt der Eingabe. Das eingestellte Datumsformat ist dann wirksam, wenn sich der Eingabefokus in dem Objekt befindet.

Folgende Darstellungen können gewählt werden:

<u>WinFmtDateAutoYY</u>	Betriebssystemabhängige Reihenfolge mit zweistelligem Jahr
<u>WinFmtDateDDMMYY</u>	Tag/Monat/Jahr
<u>WinFmtDateMMDDYY</u>	Monat/Tag/Jahr
<u>WinFmtDateYYMMDD</u>	Jahr/Monat/Tag
<u>WinFmtDateAutoYYYY</u>	Betriebssystemabhängige Reihenfolge mit vierstelligem Jahr
<u>WinFmtDateDDMMYYYY</u>	Tag/Monat/Jahr(4-stellig)
<u>WinFmtDateMMDDYYYY</u>	Monat/Tag/Jahr(4-stellig)
<u>WinFmtDateYYYYMMDD</u>	Jahr(4-stellig)/Monat/Tag
<u>WinFmtDateDDMM</u>	Format für Schnelleingabe

 Die Ausprägungen WinFmtDateAutoYY und WinFmtDateAutoYYYY definieren nur die Reihenfolge der Datumskomponenten.

Beispiele:

- In den regionalen Einstellungen der Systemsteuerung ist das kurze Datumsformat JJJJ-MM-TT definiert.
Daraus ergibt sich die Eingabereihenfolge "Jahr - Monat - Tag".
- In den regionalen Einstellungen ist das kurze Datumsformat MM/TT/JJ definiert. Daraus ergibt sich die Eingabereihenfolge "Monat - Tag - Jahr".

Kontakt

FormatEnum Datenformate bei Drag & Drop-

Operationen Typ logic[5]

Siehe Liste, Objekte, Data

In dieser Eigenschaft wird das Format festgelegt, in der die Informationen aus einer Drag & Drop-Operation vorliegen. Dazu muss der Eigenschaft das Datenformat übergeben werden. Folgende Konstanten können dafür verwendet werden:

- WinDropDataUser
- WinDropDataFile
- WinDropDataContent
- WinDropDataText
- WinDropDataRtf

Auf die Daten kann über die Eigenschaft Data zugegriffen werden.

Beispiele:

```
// Sind Daten im Text-Format vorhanden?if (aDataObject->wpFormatEnum(_WinDropDataText))...// Setz
```

Kontakt

FormatTime



Formatierung der Ausgabe

Typ int

Liste, Objekte,

WinPropGet(),

Siehe WinPropSet(),

PrtPropGet(),

PrtPropSet()

Diese Eigenschaft bestimmt die Darstellung der Uhrzeit.

Folgende Darstellungen gibt es:

WinFmtTimeHHMMSSCC Anzeige von Stunden, Minuten, Sekunden und hunderstel

Sekunden

Anzeige von Stunden, Minuten und Sekunden

WinFmtTimeHHMMSS

Anzeige von Stunden und Minuten

WinFmtTimeHHMM

Anzeige von Minuten und Sekunden

WinFmtTimeMMSS

FormatWrap

Automatischen Umbruch speichern

Typ logic
Siehe Liste, Objekte, WinPropGet(),



Diese Eigenschaft kann nur gesetzt werden, wenn die Eigenschaft AutoWrap gesetzt ist.

Ist während der Laufzeit die Eigenschaft DbTextBuf auf einen gültigen Deskriptor eines internen Textes gesetzt, hat FormatWrap bei Anwendung des WinUpdate()-Befehls folgende Funktionen:

- \$TextEdit->WinUpdate(WinUpdObj2Fld)

Beim Übertragen des Textes in die Datenbank werden im Text dort Formatierungszeichen (dezimal 31) eingefügt, wo dieser am rechten Rand des Eingabeobjektes umgebrochen wurde.

- \$TextEdit->WinUpdate(WinUpdFld2Obj)

Beim Übertragen des Textes aus der Datenbank werden evtl. vorhandene Formatierungszeichen entfernt.

Standardmäßig ist FormatWrap zurückgesetzt, so dass keine Formatierungszeichen eingefügt oder entfernt werden.

Kontakt

FormType



Form eines CanvasGraphic-Objektes im Canvas-Objekt Typ int

Siehe
Liste, Objekte, WinPropGet(), WinPropSet(),
CanvasGraphic
Diese Eigenschaft definiert die Form von CanvasGraphic-Objekten im Canvas-Objekt.
Folgende Formen sind möglich:

- _WinFormTypeRectangle (0) - Rechteck
- _WinFormTypeSquare (1) - Quadrat
- _WinFormTypeLine (2) - Linie
- _WinFormTypeOval (3) - Oval
- _WinFormTypeCircle (4) - Kreis

Kontakt

Frame



Darstellungsart des Rahmens

Typ int

Liste, Objekte,

Siehe WinPropGet(),
WinPropSet()

Diese Eigenschaft bestimmt das Aussehen des Rahmens.

Es gibt folgende Stile:

WinFrameNone Kein Rahmen

WinFrameSingle Einfacher Rahmen

WinFrameDouble Doppelter Rahmen

Die Darstellung des Rahmens ist abhängig von der Darstellung des Betriebssystems. Ein doppelter Rahmen kann nur dann gezeichnet werden, wenn eine klassische Darstellung gewählt wurde.

Kontakt

FrameEnd



Nummer des letzten Bildes

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In dieser Eigenschaft wird die Bildnummer angegeben, an der das Abspielen des Films endet. Bei der Angabe von -1 werden alle Bildsequenzen durchlaufen.

Kontakt

FrameStart



Nummer des ersten Bildes

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In dieser Eigenschaft wird die Bildnummer angegeben, an der das Abspielen des Films beginnt.

GanttFlags 

Optionen

Typ int
Liste,
Siehe Objekte

In dieser Eigenschaft werden die Optionen der Objekte GanttGraph und PrtGanttGraph gesetzt. Folgende Optionen können gesetzt werden:

- WinGanttIvlFontSizeCustom bzw. PrtGanttIvlFontSizeCustom

Ist diese Ausprägung gesetzt, findet keine automatische Anpassung der Schriftgröße innerhalb des Interval-Objekts an die Höhe des Objekts statt. Die Größe der Schriftart wird aus der Eigenschaft FontIvl bezogen.

- WinGanttZOrderDraw bzw. PrtGanttZOrderDraw

Mit dieser Ausprägung kann die Reihenfolge bestimmt werden, in der die Intervalle gezeichnet werden. Die Reihenfolge bestimmt ebenfalls welches Intervall angeklickt wurde, wenn sich die entsprechenden Bereiche überlappen.

Ist kein Flag gesetzt wird die Reihenfolge beim Zeichnen durch die Reihenfolge der Erzeugung der Intervalle bestimmt. Die Reihenfolge kann nicht während der Laufzeit verändert werden.

Ist das Flag WinGanttZOrderDraw bzw. PrtGanttZOrderDraw gesetzt, wird die Eigenschaft ZOrder bei den Intervallen ausgewertet.

- WinGanttIvlBoxZOrderDraw bzw. PrtGanttIvlBoxZOrderDraw

Mit dieser Ausprägung kann die Reihenfolge bestimmt werden, in der die Box-Objekte gezeichnet werden.

Ist kein Flag gesetzt wird die Reihenfolge beim Zeichnen durch die Reihenfolge der Erzeugung der Objekte bestimmt. Die Reihenfolge kann nicht während der Laufzeit verändert werden.

Ist das Flag WinGanttIvlBoxZOrderDraw bzw. PrtGanttIvlBoxZOrderDraw gesetzt, wird die Eigenschaft ZOrder bei den Box-Objekten ausgewertet.

- WinGanttGridDrawFirst

Die Gitterlinien des Gantt-Graphen werden zuerst gezeichnet. Die nachfolgend gezeichneten Objekte (Interval-, Box-Objekte) verdecken die Linien des Gitters.

- WinGanttScrollDragMove

Der View-Bereich des GanttGraph-Objekts wird beim Verschieben von Intervallen mit verschoben.

- WinGanttScrollDragSize

Der View-Bereich des GanttGraph-Objekts wird beim Vergrößern von Intervallen mit verschoben.

- WinGanttIvlDragShowRect

Kontakt

Bei einer Drag & Drop Aktion eines Intervalls wird die Größe des Intervalls durch ein Drag-Rechteck angezeigt, das der Breite des Intervalls entspricht.

Die Ausprägung wirkt sich nur dann aus, wenn die Quelle und das Ziel der Drag & Drop Aktion ein GanttGraph-Objekt ist.

- WinGanttDropLimitBound

Wird bei einer Drag & Drop Aktion das Intervall aus dem angezeigten Raster hinaus geschoben, wird der NoDrop-Mauszeiger angezeigt.

- WinGanttScrollAccelerate

Die Scrollgeschwindigkeit wird nach einiger Zeit erhöht, wenn die Optionen WinGanttScrollDragMove und/oder WinGanttScrollDragSize gesetzt sind.

- WinGanttIvlBoxClick

Auslösen des Ereignisses EvtMenuItem beim Anklicken eines Box-Objekts.

- WinGanttIvlBoxShowText bzw. PrtGanttIvlBoxShowText

Ist diese Ausprägung gesetzt, werden in den Box-Objekten innerhalb des GanttGraph-Objekts die Texte angezeigt. Die Schriftart wird über die Eigenschaft FontBox gesetzt.

- WinGanttIvlBoxInflate

Rand des Box-Objekts transparent oder mit der Hintergrundfarbe des Objekts zeichnen.

- WinGanttIvlBoxLineNoHelpTip

Wird ein Box-Objekt als Linie gezeichnet (Eigenschaft StyleIvl = WinStyleIvlBoxHorz oder WinStyleIvlBoxVert), wird kein HelpTip angezeigt.

- WinGanttCellActiveDrawCellOnly bzw. PrtGanttCellActiveDrawCellOnly

Der Rahmen, in der Farbe ColCellActive, wird nur um die aktive Zelle (Eigenschaften CellActiveHorz und CellActiveVert) gezeichnet, statt um die komplette Zeile und Spalte.

- WinGanttIvlBoxDragDropAllowed

Drag & Drop für Box-Objekte ermöglichen.

- WinGanttIvlBoxSizingAllowed

Größenänderungen per Drag & Drop für Box-Objekte ermöglichen.

Das Setzen der Ausprägungen WinGanttScroll..., WinGanttIvlBoxInflate und WinGanttIvlBoxLineNoHelpTip kann nur beim GanttGraph-Objekt erfolgen.

GrayingModeUser

Darstellung ausgegrauster Symbole

Typ int

Siehe Liste, Objekte

Benutzerdefinierte Kacheln (Tiles) in der gesperrten Darstellung werden von CONZEPT 16 automatisch in Graustufen umkodiert. Die Eigenschaft kann gesetzt werden, um eine Graufärbung zu erreichen, die sich besser vom Normal-Zustand abhebt.

Einer der folgenden Werte kann gesetzt werden:

WinGrayingDefault Standard für die ausgegraute Darstellung.

WinGrayingLight Etwas hellere ausgegraute Darstellung.

WinGrayingDark Etwas dunklere ausgegraute Darstellung.

Bei der Zuweisung anderer Werte, wird der Laufzeitfehler ErrValueInvalid ausgelöst.

Kontakt

Objekte von GrayingModeUser

Folgende Objekte unterstützen die Eigenschaft [GraryingModeUser](#)

[GraryingModeUser](#),

[Liste der Objekte](#),

[Liste der](#)

[Eigenschaften](#)

- [App](#)

Siehe

Kontakt

GridHeight  Vertikaler Abstand zwischen
Rasterpunkten Typ int

Siehe [Liste](#), [Objekte](#), [WinPropGet\(\)](#),

Diese Eigenschaft definiert den vertikalen Abstand zwischen zwei Rasterpunkten in logischen Einheiten. Für die Umrechnung von Einheiten (z. B. Millimeter, Zentimeter) in logische Einheiten kann der Befehl [PrtUnitLog\(\)](#) verwendet werden.



Die Rasterpunkte werden nur dargestellt, wenn sowohl die Eigenschaft [GridWidth](#), als auch die Eigenschaft GridHeight einen Wert größer 0 haben.

Kontakt

GridWidth Horizontaler Abstand zwischen
Rasterpunkten Typ int

Siehe [Liste](#), [Objekte](#), [WinPropGet\(\)](#),

Diese Eigenschaft definiert den horizontalen Abstand zwischen zwei Rasterpunkten in logischen Einheiten. Für die Umrechnung von Einheiten (z. B. Millimeter, Zentimeter) in logische Einheiten kann der Befehl [PrtUnitLog\(\)](#) verwendet werden.



Die Rasterpunkte werden nur dargestellt, wenn sowohl die Eigenschaft GridWidth, als auch die Eigenschaft GridHeight einen Wert größer 0 haben.

Kontakt

Group

Gruppe

Typ int

Liste,
Objekte,
Siehe WinPropGet(),
 WinPropSet(),
PrtPropGet(),
PrtPropSet()

Mit dieser Eigenschaft können mehrere Objekte einer Gruppe zugeordnet werden.
Alle Objekte mit der gleichen Nummer gehören zu einer Gruppe.

Folgende Objekte besitzen diese Eigenschaft:

- Statusbar-Button und Toolbar-Button

Alle Schaltflächen einer Gruppe können über die Eigenschaft CurrentInt des übergeordneten Objekts ein- oder ausgeblendet werden. Buttons der Gruppe 0 sind davon nicht betroffen.

- PrtRtf

Ein RTF-Text kann in mehreren PrtRtf-Objekten dargestellt werden. Dazu muss in allen beteiligten PrtRtf-Objekten die Eigenschaft Group auf die gleiche Nummer gesetzt werden. Im ersten Objekt wird die Quelle des Textes angegeben, während in allen anderen Objekten der Gruppe die Eigenschaft StreamSource auf den Wert WinStreamNone gesetzt wird. Der Bereich des angezeigten Textes wird in der Eigenschaft Range angezeigt und kann nicht mehr gesetzt werden.

Beispiel:

```
$Toolbar->wpCurrentInt # 2;
```

Kontakt

GroupDrawSelect



Gruppe selektiert darstellen, wenn sie ausgewählt ist Typ logic

Siehe

Liste, Objekte, WinPropGet(), WinPropSet(),

EvtLstGroupInit

Die Eigenschaft kann bei einer Gruppe im Ereignis EvtLstGroupInit gesetzt werden, um zu definieren, ob ihr Selektionsrahmen gezeichnet wird, wenn sie ausgewählt wird. Standardmäßig hat die Eigenschaft den Wert true.

Kontakt

GsEncryption



Verschlüsselungstyp von PDF-Dokumenten

Typ int
Siehe Liste, Objekte, PrtPropGet(),
 PrtPropSet()

Diese Eigenschaft definiert die Schlüssellänge, mit der das PDF-Dokument verschlüsselt wird, sofern ein Passwort in der Eigenschaft GsPasswordOwner angegeben ist. Das Passwort wird dann zur Verschlüsselung verwendet.

Folgende Verschlüsselungen können angegeben werden:

- PpcGsEncrypt40
- PpcGsEncrypt128



Die Eigenschaft wird nur ausgewertet, wenn mit der Anweisung PpcMakePdf() ein PDF-Dokument mit Hilfe von GhostScript erzeugt wird.

GsOptions

Optionen für GhostScript

Typ [alpha](#)

Siehe [Liste, Objekte](#)

Hier können benutzerdefinierte Optionen angegeben werden. Die Optionen werden unverändert an Ghostscript weitergeleitet.

Mit folgender Anweisung wird ein PDF/A-konformes Dokument erzeugt:

```
aPpcObject->ppGsOptions # '-dPDFA;-dNOOUTERSAVE;-dUseCIEColor;-sProcessColorModel=DeviceCMYK';aPp
```

In diesem Beispiel wird das Standard-Farbprofil (sRGB.icc) verwendet. Soll ein anderes Farbprofil benutzt werden, kann es in der Eigenschaft [GsPdfColorProfile](#) angegeben werden. Alternativ kann ein PDF/A auch mit Hilfe des Übergabeparameters [PpcPdfA](#) bei der Anweisung [PpcMakePdf\(\)](#) erzeugt werden.



Die Eigenschaft wird nur ausgewertet, wenn mit der Anweisung [PpcMakePdf\(\)](#) ein PDF-Dokument mit Hilfe von GhostScript erzeugt wird.

GsPasswordOwner

Passwort des Eigentümers

Typ alpha

Liste, Objekte,

Siehe PrtPropGet(),

PrtPropSet()

Diese Eigenschaft definiert ein Passwort für den Eigentümer des PDF-Dokuments. Das Password muss gesetzt sein, um eine Verschlüsselung (siehe Eigenschaft GsEncryption), das Setzen eines Benutzerkennworts (GsPasswordUser) oder das Setzen von Rechten (GsPermissions) zu ermöglichen. Das Passwort muss dann beim Ändern der Sicherheitseinstellungen des Dokuments angegeben werden.



Die Eigenschaft wird nur ausgewertet, wenn mit der Anweisung PpcMakePdf() ein PDF-Dokument mit Hilfe von GhostScript erzeugt wird.

GsPasswordUser

Passwort des Benutzers

Typ alpha

Liste, Objekte,

Siehe PrtPropGet(),

PrtPropSet()

Diese Eigenschaft definiert das Passwort des Benutzers eines PDF-Dokuments. Das Passwort muss beim Öffnen des Dokuments angegeben werden.



Die Eigenschaft wird nur ausgewertet, wenn mit der Anweisung PpcMakePdf() ein PDF-Dokument mit Hilfe von GhostScript erzeugt wird.

GsPdfColorProfile

Farbprofil für PDF/A-Dokumente

Typ [alpha](#)

Siehe [Liste, Objekte](#)

Soll ein PDF/A-konformes Dokument erstellt werden, muss ein Farbprofil angegeben werden. Ist diese Eigenschaft leer, bindet CONZEPT 16 das Farbprofil sRGB.icc ein. Soll ein anderes Farbprofil verwendet werden, kann es in dieser Eigenschaft angegeben werden.

Die Eigenschaft wird nur ausgewertet, wenn in der Eigenschaft [GsOptions](#) der Parameter -dPDFA; übergeben wird.



Die Eigenschaft wird nur ausgewertet, wenn mit der Anweisung [PpcMakePdf\(\)](#) ein PDF-Dokument mit Hilfe von GhostScript erzeugt wird.

Kontakt

GsPermissions



Rechtebeschränkung für PDF-Dokumente

Typ **int**
Siehe [Liste](#), [Objekte](#), [PrtPropGet\(\)](#),
[PrtPropSet\(\)](#)

Diese Eigenschaft definiert die Zugriffs-Beschränkungen, die bei der PDF-Erstellung durch die Anweisung [PpcMakePdf\(\)](#) berücksichtigt werden. Damit die Beschränkungen gesetzt werden können, muss in der Eigenschaft [GsPasswordOwner](#) ein Kennwort gesetzt sein.

Je nach Revision der PDF-Sicherheits-Unterstützung haben die Konstanten unterschiedliche Bedeutungen. Zum einfacheren Setzen der Einschränkungen können über die Konstanten [PpcGsDenyAll](#), [PpcGsDenyPdf12](#) und [PpcGsDenyPdf14](#) alle Rechte, alle Rechte der Revision 2 bzw. alle Rechte der Revision 3 entzogen werden.

Konstante	PDF-Revision	Beschreibung
PpcGsDenyAll	2 und größer	Alle Rechte entziehen
PpcGsDenyPdf12	2 und größer	Rechte der Revision 2 unterbinden
PpcGsDenyPrint	2 und größer	Drucken des PDF-Dokuments unterbinden
PpcGsDenyModify	2 und größer	Änderungen am PDF-Dokument unterbinden
PpcGsDenyCopy	2 und größer	PDF-Inhalt kopieren unterbinden
PpcGsDenyAnnotation	2 und größer	Eintragen von Kommentaren und unterbinden
PpcGsDenyPdf14	3 und größer	Rechte der Revision 3 und größer unterbinden
PpcGsDenyFormFields	3 und größer	Ausfüllen von Formularen unterbinden
PpcGsDenyExtractText	3 und größer	Zugriff von Programmen auf den Text unterbinden
PpcGsDenyAssemble	3 und größer	Seitenänderungen im Dokument unterbinden
PpcGsDenyPrintHigh	3 und größer	Drucken in hoher Auflösung unterbinden



Die Eigenschaft wird nur ausgewertet, wenn mit der Anweisung [PpcMakePdf\(\)](#) ein PDF-Dokument mit Hilfe von GhostScript erzeugt wird.

Beim Setzen dieser Eigenschaft werden bestimmte Optionen innerhalb des PDF-Dokuments gesetzt. Die Interpretation dieser Optionen obliegt dem Programm, das zur Anzeige des PDF-Dokuments verwendet wird.

Beispiele:

```
tPpcObject->ppGsPermissions # _PpcGsDenyPrint | _PpcGsDenyPrintHigh; // Dokument nicht druckt
```

Kontakt

GsVersion



Versionsnummer von GhostScript

Typ [alpha](#)

Siehe [Liste](#), [Objekte](#), [PrtPropGet\(\)](#)

Diese Eigenschaft enthält die Versionsnummer von GhostScript. Ist GhostScript nicht installiert, enthält die Eigenschaft den Wert ". Die Eigenschaft kann nur gelesen werden.

Kontakt

HdrVerticalText



Textausrichtung in Spaltenköpfen

Typ logic

HdrHeight, HdrWordBreak,

Siehe Liste, Objekte, WinPropGet(),

WinPropSet()

Durch Setzen des Eigenschaftswertes auf true wird der Text im Spaltenkopf (**Eigenschaft Caption**) vertikal dargestellt, ansonsten horizontal.

Die Höhe der Spaltenköpfe kann über die Eigenschaft HdrHeight verändert werden.

Kontakt

Objekte von HdrVerticalText

Folgende Objekte unterstützen die Eigenschaft HdrVerticalText

[HdrVerticalText](#),

[Liste der](#)

Siehe [Objekte](#), [Liste](#)

[der](#)

[Eigenschaften](#)

- [Column](#)

HdrWordBreak

Textumbruch in Spaltenköpfen

Typ logic

HdrHeight,

Siehe HdrVerticalText, Liste,

Objekte, WinPropGet(),

WinPropSet()

Durch Setzen des Eigenschaftswertes auf true wird der Text im Spaltenkopf (Eigenschaft Caption) an der Umrandung umgebrochen, sonst findet kein Umbruch statt.

Die Höhe der Spaltenköpfe kann über die Eigenschaft HdrHeight verändert werden.

Kontakt

Objekte von HdrWordBreak

Folgende Objekte unterstützen die Eigenschaft **HdrWordBreak**

HdrWordBreak,

Liste der

Siehe **Objekte**, **Liste**

der

Eigenschaften

- **Column**

HeightIvl

Höhe des Intervalls

Typ int

Liste, Objekte,

Siehe PrtPropGet(),
PrtPropSet()

Die Höhe des Intervalls in einem PrtGanttGraph-Objekt wird durch die Höhe der Zeile bestimmt. In dieser Eigenschaft kann die Höhe des Intervalls als Prozent der Höhe der Zeile angegeben werden.

Die Höhe kann zwischen 25 und 100 Prozent der Höhe der Zeile bestimmt werden.

Das Intervall wird zentriert in der Zeile gezeichnet.

Kontakt

HeightMode

Zell-Höhenmodus

Typ int

Liste,

Siehe Objekte,

PrtPropGet(),

PrtPropSet()

Die Eigenschaft bestimmt, ob und wie sich die Zelle an den Zell-Inhalt anpasst.

PrtHeightContent Inhaltabhängige Zellhöhe

PrtHeightFix Minimale feste Zellhöhe

HyphMarginTwips

Bereich ohne Silbentrennung

Typ int

Liste, Objekte,

Siehe EvtHyphenate,

WinPropGet(),

WinPropSet()

 Die Eigenschaft kann nur unter Windows-Betriebssystemen gesetzt werden, die die Datei msftedit.dll mit mindestens der Version 5.41.15.1515 verwenden. Die Version kann im Info-Dialog des Designers ermittelt werden.

In dieser Eigenschaft wird innerhalb eines RtfEdit-Objekts ein Bereich am rechten Rand definiert, in dem keine Silbentrennung durchgeführt wird. Der Bereich wird in Twips (siehe PrtUnitLog()) angegeben. Wird ein negativer Wert angegeben, wird die Silbentrennung deaktiviert.

Beginnt ein Wort innerhalb das angegebenen Bereichs, wird es nicht getrennt, sondern vollständig in die nächste Zeile geschrieben. Auf diese Weise können zu kurze Silben am Ende einer Zeile verhindert werden.

Kontakt

ID

--	--

ID des Objektes

Typ int

Liste, Objekte,

Siehe WinPropGet(),
WinPropSet()

Je nach Objekt hat die ID folgende Bedeutung:

- Button

Die ID wird benötigt um zu ermitteln, welche Schaltfläche zum Beenden eines Dialoges gedrückt wurde.

<u>WinIdNone</u>	Keine ID (0) gesetzt
<u>WinIdYes</u>	Ja-Schaltfläche
<u>WinIdNo</u>	Nein-Schaltfläche
<u>WinIdOk</u>	Ok-Schaltfläche
<u>WinIdCancel</u>	Abbrechen-Schaltfläche
<u>WinIdClose</u>	Schließen-Schaltfläche
<u>WinIdHelp</u>	Hilfe-Schaltfläche

Neben den vordefinierten IDs können auch benutzerdefinierte Werte verwendet werden.

Die ID wird beim Schließen des Dialoges als Resultat von WinDialog(), WinDialogRun() oder WinDialogResult() zurückgegeben, wenn bei der Schaltfläche die Eigenschaft TypeButton auf WinBtnClose gesetzt ist.

Die ID kann ebenfalls verwendet werden, um in der Programmierung Schaltflächen zu unterscheiden, die die gleichen EvtClicked-Funktion aufrufen.

- Radiobutton

Durch das Setzen der Eigenschaft ID, können unterschiedliche Radiobutton-Objekte und Button-Objekte mit der gleichen EvtClicked-Funktion verarbeitet werden. Die Objekte werden dann Anhand der Eigenschaft ID unterschieden.

- Interval

Der Wert der Eigenschaft wird an die Ereignisse EvtMouseItem und EvtMenuContext im letzten Parameter übergeben. Hierdurch können frei definierbare Werte mit dem Intervall verbunden werden (z. B. die Datensatz-ID des Datensatzes zum Intervall).

Kontakt

ImageSource

Quelle des Bildes

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft wird die Quelle des Bildes angegeben. In der Eigenschaft können folgende Konstanten angegeben werden:

- _WinImageSourceCaption (1) (default)

Der Name des Bildes wird aus der Eigenschaft Caption ermittelt.

- _WinImageSourceMemory (2)

Das darzustellende Bild befindet sich in einem Memory-Objekt. Der Deskriptor auf das Objekt muss in der Eigenschaft MemObjHandle angegeben werden.

Kontakt

IndexColBkg Hintergrundfarbe für untergeordnete
Objekte Typ **int**

Siehe [Liste](#), [Objekte](#), [WinPropGet\(\)](#),
[WinPropSet\(\)](#), [IndexColFg](#), [IndexData](#)

In dieser Eigenschaft können bis zu 16 Hintergrundfarben angegeben werden. Die verschiedenen Farben werden über Indizes getrennt.

Beispiel:

```
tHdlTreeView->wpIndexColBkg(0) # _WinColWhite;tHdlTreeView->wpIndexColBkg(1) # _WinColLightYellow
```

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe [Beispiel](#)).

In einem untergeordneten Objekt wird dann in der Eigenschaft [IndexData](#) dem Bereich der Hintergrundfarbe ein bestimmter Index zugewiesen:

```
tHdlTreeNode->wpIndexData(WinFlagIndexColBkg) # 1;
```

Hier erfolgt nur die Zuweisung der Hintergrundfarbe, für die Vordergrundfarbe kann ein anderer Index gewählt werden.

Kontakt

IndexColFg Vordergrundfarbe für untergeordnete

Objekte Typ **int**

Siehe [Liste](#), [Objekte](#), [WinPropGet\(\)](#),

[WinPropSet\(\)](#), [IndexColBkg](#), [IndexData](#)

In dieser Eigenschaft können bis zu 16 Vordergrundfarben angegeben werden. Die verschiedenen Farben werden über Indizes getrennt.

Beispiel:

```
tHdlTreeView->wpIndexColFg(0) # _WinColBlack; tHdlTreeView->wpIndexColFg(1) # _WinColBlue; tHdlTr
```

Alternativ zu den vordefinierten Farbkonstanten können RGB-Werte direkt verarbeitet werden (siehe [Beispiel](#)).

In einem untergeordneten Objekt wird dann in der Eigenschaft [IndexData](#) dem Bereich der Vordergrundfarbe ein bestimmter Index zugewiesen:

```
tHdlTreeNode->wpIndexData(WinFlagIndexColFg) # 1;
```

Hier erfolgt nur die Zuweisung der Vordergrundfarbe, für die Hintergrundfarbe kann ein anderer Index gewählt werden.

IndexData

Darstellungsindex des Eltern-Objektes

Typ int

Liste, Objekte, WinPropGet(),

Siehe WinPropSet(), IndexColFg,
IndexColBkg, IndexFont

Beim Eltern-Objekt werden die Eigenschaften IndexColFg, IndexColBkg und IndexFont auf verschiedene Werte gesetzt. Jede dieser Eigenschaften entspricht einem Array. Es können somit zum Beispiel in der Eigenschaft IndexColFg bis zu 16 Farbwerte eingetragen werden.

Damit das untergeordnete Objekt eine bestimmte Vordergrundfarbe verwendet, muss die Eigenschaft IndexData auf den entsprechenden Index gesetzt werden. Die Eigenschaft selbst besteht aus drei Werten:

- WinFlagIndexColFg Vordergrundfarbe
- WinFlagIndexColBkg Hintergrundfarbe
- WinFlagIndexFont Zeichensatzattribute

Jeder dieser Bereiche kann mit einem eigenen Index belegt werden.

Beispiel:

```
// Zuweisung beim Eltern-Objekt $TreeView->wpIndexColFg(0) # _WinColBlack;$TreeView->wpIndexColF
```

Ohne die Angabe einer WinFlagIndex...-Konstante kann die gepackte Information ermittelt oder gesetzt werden. Die Information setzt sich dabei aus drei Hexadezimalziffern zusammen. Die höchstwertige Ziffer gibt den Index der Vordergrundfarbe an gefolgt von der Hintergrundfarbe und dem Font.

Beispiel:

```
tTreeNode->wpIndexData # 0x102; // Vordergrundfarbe 1, Hintergrundfarbe 0, Font 2
```

Kontakt

IndexFont  Schriftattribute für untergeordnete

Objekte Typ font

Siehe [Liste](#), [Objekte](#), [WinPropGet\(\)](#),

In dieser Eigenschaft können bis zu 16 Fontausprägungen angegeben werden. Die verschiedenen Ausprägungen werden über Indizes getrennt.

Beispiel:

```
tFont:Attributes # _WinFontAttrNormal; tHdlTreeView->wpIndexFont(0) # tFont; tFont:Attributes # _
```

Die verwendete Schriftart und -größe können nicht verändert werden.

In einem untergeordneten Objekt wird dann in der Eigenschaft IndexData dem Bereich der Schriftart ein bestimmter Index zugewiesen:

```
tHdlTreeNode->wpIndexData(_WinFlagIndexFont) # 1;
```

InputCheck

Überprüfung Eingabe

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Die Eigenschaft steuert die Art und Weise, in der die Eingabe im Objekt DateEdit überprüft wird.

Es gibt folgende Überprüfungen:

WinInpCheckStrict - Prüfung während der Eingabe

WinInpCheckLeave - Prüfung bei Verlassen des Objekts

Kontakt

InputMode

Eingabemodus

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Diese Eigenschaft legt den Eingabemodus eines Objektes für den Benutzer fest.

Es gibt folgende Eingabemodi:

WinInpUserdefined Benutzerdefiniert (der Benutzer kann mit der Einfg-Taste zwischen den Modi Overwrite und Insert wechseln)

WinInpOverwrite Überschreib-Modus

WinInpInsert Einfüge-Modus

Kontakt

InstallCtxDocEdit

Installation der DocEdit-Erweiterung

Typ logic

Siehe Liste, Objekte

Diese Eigenschaft des App-Objekts muss auf true gesetzt werden, damit in dem Objekt CtxDocEdit Dokumente angezeigt werden können.

Durch das Setzen der Eigenschaft werden die Dateien tx4ole28.ocx und wspell.ocx in der Registrierungsdatenbank von Windows eingetragen. Der Einträge werden für die Darstellung und Rechtschreibprüfung der Dokumente benötigt. Die Eintragungen erfolgen in den Verzeichnissen Arbeitsplatz\HKEY_CLASSES_ROOT\TypeLib und Arbeitsplatz\HKEY_LOCAL_MACHINE\SOFTWARE\Classes\TypeLib. Die Eintragungen in die Registrierungsdatenbank müssen nur einmal erfolgen, sie kann anschließend über diese Eigenschaft abgefragt werden.

Der Erfolg der Installation kann über die Anweisung WinInfo(0, _WinErrorCode) abgefragt werden. Wird ein Wert kleiner 0 zurückgegeben, ist ein Fehler aufgetreten.

Die Suchpfade für das Auffinden der Installation können über die Eigenschaft InstallPathCommon gesetzt und ermittelt werden.



Die Installation der DocEdit-Erweiterung kann nur von einem Windows-Benutzer vorgenommen werden, der Schreibrechte in der Registrierungsdatenbank besitzt. Bei aktiver Benutzerkontensteuerung muss die Berechtigung explizit vom Anwender erlaubt werden. Die Benutzerkontensteuerung des Systems zeigt ein entsprechendes Fenster an, das der Benutzer bestätigen muss. Ist die Erweiterung installiert, kann sie von allen Windows-Benutzern verwendet werden.

Beispiel:

```
if (!_App->wpInstallCtxDocEdit)    _App->wpInstallCtxDocEdit # true;if (!_App->wpInstallCtxDocEdit)
```

Kontakt

InstallCtxOffice

Installation der Office-Erweiterung

Typ logic

Siehe Liste, Objekte

Diese Eigenschaft des App-Objekts muss auf true gesetzt werden, damit in dem Objekt CtxOffice Office-Dokumente angezeigt werden können.

Durch das Setzen der Eigenschaft wird das ActiveX-Steuerelement c16_ctxoffice.ocx in der Registrierungsdatenbank von Windows eingetragen. Der Eintrag wird für die Darstellung der Dokumente benötigt. Die Eintragung erfolgt in den Verzeichnissen Arbeitsplatz\HKEY_CLASSES_ROOT\TypeLib und Arbeitsplatz\HKEY_LOCAL_MACHINE\SOFTWARE\Classes\TypeLib. Die Eintragung in die Registrierungsdatenbank muss nur einmal erfolgen, sie kann anschließend über diese Eigenschaft abgefragt werden.

Der Erfolg der Installation kann über die Anweisung WinInfo(0, _WinErrorCode) abgefragt werden. Wird ein Wert kleiner 0 zurückgegeben, ist ein Fehler aufgetreten.



Die Installation der Erweiterung kann nur von einem Windows-Benutzer vorgenommen werden, der Schreibrechte in der Registrierungsdatenbank besitzt. Bei aktivierter Benutzerkontensteuerung muss die Berechtigung explizit vom Anwender erlaubt werden. Die Benutzerkontensteuerung des Systems zeigt ein entsprechendes Fenster an, das der Benutzer bestätigen muss. Ist die Erweiterung installiert, kann sie von allen Windows-Benutzern verwendet werden.

Beispiel:

```
if (!_App->wpInstallCtxOffice)  _App->wpInstallCtxOffice # true;if (!_App->wpInstallCtxOffice){
```

Kontakt

InstallChromium

Installation der Chromium-Komponenten

Typ [logic](#)

Siehe [Liste, Objekte](#)

Diese Eigenschaft des [App](#)-Objekts muss auf [true](#) gesetzt werden, damit das [Chromium](#)-Objekt verwendet werden kann.

Durch das Setzen der Eigenschaft werden die [Chromium](#)-Komponenten konfiguriert.

Diese müssen sich in einem der durch [InstallPathCommon](#) gesetzten Pfade befinden.

Die Komponenten wurden erfolgreich konfiguriert und geladen, wenn die Eigenschaft [true](#) zurückliefert.



Das [Chromium](#)-Objekt kann nur einmal zur Laufzeit geladen werden.

Nachfolgende Aufrufe von [InstallChromium](#) führen nicht zu einer Reinitialisierung.

Beispiel:

```
if (!App->wpInstallChromium) _App->wpInstallChromium # true;if (!App->wpInstallChromium){ //
```

Kontakt

Objekte von InstallChromium

Folgende Objekte unterstützen die Eigenschaft InstallChromium

InstallChromium,

Liste der

Siehe Objekte, Liste

der

Eigenschaften

- App

Kontakt

InstallPathCommon Suchpfade für gemeinsame
Komponenten Typ alpha

Siehe Liste, Objekte

Die Eigenschaft definiert die Pfade, die für die Installation des Chromium- und CtxDocEdit-Objektes berücksichtigt werden. Einzelne Pfade können durch ein Semicolon (;) separiert werden. Die Pfade werden in der Reihenfolge durchsucht, in der sie in der Eigenschaft angegeben sind. Relative Pfadangaben sind zulässig und beziehen sich auf das Verzeichnis, aus dem der Client gestartet wurde.

Durch lesen der Eigenschaft können die eingestellten Pfade auch ermittelt werden.

Die Eigenschaft wird standardmäßig voreingestellt:

1. Common-Pfad des aktuellen Client-Verzeichnisses

Das dem Client übergeordnete Common-Verzeichnis.

Beispiel: wurde der Client aus dem Verzeichnis c:\myInstall\Client gestartet, dann ist das Common-Verzeichnis: c:\myInstall\Common.

2. Common-Pfad des Installations-Verzeichnisses

Das Common-Verzeichnis innerhalb des Installations-Verzeichnisses (unabhängig davon, ob eine gemeinsame Komponente installiert ist). Das Verzeichnis ist nur vorhanden, wenn eine Installation mit der CONZEPT 16-Setup-Routine durchgeführt wurde.

Beispiel: Ist der Client im Verzeichnis c:\program files\concept 16\client installiert, dann ist das Common-Verzeichnis c:\program files\concept 16\common.

Im Fall der obigen Konstellation hätte die Eigenschaft folgenden Inhalt:

c:\myInstall\Common;c:\program files\concept 16\common

Die in der Eigenschaft eingetragenen Pfade werden beim Setzen von wpInstallCtxDocEdit und wpInstallChromium berücksichtigt.



Das Chromium-Objekt kann nur einmal zur Laufzeit geladen werden. Nachfolgende Aufrufe von wpInstallChromium führen nicht zu einer Reinitialisierung

Kontakt

Objekte von InstallPathCommon

Folgende Objekte unterstützen die Eigenschaft InstallPathCommon Liste der
Objekte, Liste

Siehe der
Eigenschaften

- App

Kontakt

JobCount

Druckjob-Anzahl

Typ int

Liste,

Siehe Objekte,

PrtPropGet()

Über diese Eigenschaft kann die Anzahl der Druckjobs, die sich in der Warteschlange des Druckers befinden, ermittelt werden.

Kontakt

JobPriority

Druckjob Priorität

Typ int

Liste,

Siehe Objekte,

PrtPropGet(),

PrtPropSet()

Gibt einen Wert im Bereich von 1 (niedrig) bis 99 (hoch) für die Druckjob-Priorität an.

Justify

Ausrichtung des Textes

Typ int

Liste, Objekte,

WinPropGet(),

Siehe WinPropSet(),

PrtPropGet(),

PrtPropSet()

Mit dieser Eigenschaft wird die Ausrichtung des Textes gesetzt.

Bei den Objekten HyperLink und Label gilt die Ausrichtung nur, wenn das Objekt fokussiert ist.

Ist das Objekt nicht fokussiert ist, wird die Einstellung in der Eigenschaft JustifyView verwendet.

Es gibt folgende Ausrichtungen:

WinJustLeft - Text wird linksbündig dargestellt

WinJustCenter - Text wird zentriert dargestellt

WinJustRight - Text wird rechtsbündig dargestellt

Kontakt

JustifyVert Angabe der vertikalen Text-Ausrichtung
Ausrichtung Typ int

Siehe Liste, Objekte

Über diese Eigenschaft kann die vertikale Text-Ausrichtung (oben, unten, zentriert) angegeben werden.

Es gibt folgende Text-Ausrichtungen:

- WinJustBottom - Text-Ausrichtung unten
- WinJustCenter - Text-Ausrichtung zentriert
- WinJustTop - Text-Ausrichtung oben

Kontakt

JustifyView



Ausrichtung, wenn das Objekt nicht fokussiert ist Typ int

Siehe
Liste, Objekte, WinPropGet(), WinPropSet(),
PrtPropGet(), PrtPropSet()
Mit dieser Eigenschaft wird die Ausrichtung des Textes innerhalb eines Objekts gesetzt.

Diese Ausrichtung gilt nur, wenn das Objekt nicht fokussiert ist. Bei Fokussierung auf das Objekt wird die Einstellung in der Eigenschaft **Justify** verwendet.

Es gibt folgende Ausrichtungen:

WinJustLeft - Text wird linksbündig dargestellt

WinJustCenter - Text wird zentriert dargestellt

WinJustRight - Text wird rechtsbündig dargestellt

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet(), WinPropSet(), PrtPropGet() und PrtPropSet() gelesen bzw. geändert werden.

Kontakt

Special (Objektspezifische Eigenschaften) - K bis O

<u>Eigenschaften sortiert</u>	
<u>nach Gruppen,</u>	
<u>Alphabetische Liste</u>	
Siehe <u>aller Eigenschaften,</u>	
<u>Liste aller</u>	
<u>Special-Eigenschaften</u>	
<u>LangDisplay</u>	Sprache von Standard-Dialogen einstellen
<u>LeaveAtEnd</u>	Verlassen am Ende
<u>LengthMax</u>	Eingabelänge des Objektes
<u>LineCapEnd</u>	Pfeiltyp für den Endpunkt von Linien im <u>Canvas-Objekt</u>
<u>LineCapStart</u>	Pfeiltyp für den Startpunkt von Linien im <u>Canvas-Objekt</u>
<u>LineJustify</u>	Ausrichtung von Linien im <u>Canvas-Objekt</u>
<u>LineWidth</u>	Breite des <u>CanvasGraphic-Objektes</u> Linie im <u>Canvas-Objekt</u>
<u>LinkText</u>	Aktion, die beim Anklicken auszuführen ist
<u>Locale</u>	Länderspezifische Darstellung zuweisen
<u>Location</u>	Drucker-Standort
<u>LstFlags</u>	Optionen
<u>LstStyle</u>	Darstellung der Spaltentitel
<u>MarginLeft</u>	Linker Randabstand
<u>MaxBigInt</u>	Maximaler Eingabewert des Objektes
<u>MaxDate</u>	Maximaler Eingabewert des Objektes
<u>MaxDecimal</u>	Maximaler Eingabewert des Objektes
<u>MaxFloat</u>	Maximaler Eingabewert des Objektes
<u>MaxInt</u>	Maximaler Eingabewert des Objektes
<u>MaxTime</u>	Maximaler Eingabewert des Objektes
<u>MemObjHandle</u>	Deskriptor auf ein Memory-Objekt
<u>MinBigInt</u>	Minimaler Eingabewert des Objektes
<u>MinDate</u>	Minimaler Eingabewert des Objektes
<u>MinDecimal</u>	Minimaler Eingabewert des Objektes
<u>MinFloat</u>	Minimaler Eingabewert des Objektes
<u>MinInt</u>	Minimaler Eingabewert des Objektes
<u>MinTime</u>	Minimaler Eingabewert des Objektes
<u>MobileMode</u>	Unterstützung auf mobilen Geräten
<u>ModeColor</u>	Eingabebereich des ColorEdit
<u>ModeDraw</u>	Art der Bild-Darstellung
<u>ModeEffect</u>	Drehung des Bildes oder Dokuments
<u>ModeOptimize</u>	Optimierte Anzeige von TIFF-Formaten
<u>ModeZoom</u>	Art der Darstellung
<u>ModuleUnit</u>	Breite des schmalen Elements im Barcode
<u>NameComputer</u>	Rechnername
<u>NameDriver</u>	Name des Drucker-Treibers
<u>NamePort</u>	Name des Drucker-Ports

Kontakt

<u>NamePrinter</u>	Name des Druckers
<u>NameServer</u>	Name des Servers
<u>NameShare</u>	Share-Name des Druckers
<u>NameUser</u>	Name des Benutzers ermitteln
<u>NodeDynamic</u>	Dynamischer Knoten
<u>NodeExpanded</u>	Knoten offen / geschlossen
<u>NodeStyle</u>	Knoten-Symbole
<u>NotifyParent</u>	Funktion übergeben
<u>ObjLink</u>	Verweis auf ein Objekt
<u>OffsetLeft</u>	Linke Startposition
<u>OffsetTop</u>	Obere Startposition
<u>Opacity</u>	Deckkraft des Bildes
<u>OrderPass</u>	Reihenfolge der Notizbuchseiten

Kontakt

LangDisplay Sprache von Standard-Dialogen
einstellen Typ alpha(2)

Liste, Objekte, WinPropGet(),
Siehe WinPropSet(),

WinLangDisplayEffective

Mit dieser Eigenschaft wird die Sprache folgender Standard-Dialoge eingestellt:

- Druckvorschau
- Statusanzeige bei OEMSave()
- Statusanzeige bei SelRun()
- Anzeige der Programminformation bei WinOpen(_WinC16Info, ...)
- Dialog der Benutzerpflege bei WinUrmDialog()

Das Setzen der Eigenschaft muss vor der Anzeige des entsprechenden Dialogs erfolgen. Die Eigenschaft kann folgende Werte annehmen:

'??'	Aktuell eingestellte Sprache aus der <u>c16.cfg</u> bzw. <u>c16_apgi.cfg</u> übernehmen.
'DE'	Deutsch
'EN'	U.S. Englisch
'*U'	Systemssprache des aktuellen Windows-Benutzers

Als Standard ist der Wert '??' eingetragen.

Beispiel

```
..._App->wpLangDisplay # 'EN';...tErr # tJob->PrtJobClose(_PrtJobPreview);
```

LeaveAtEnd

Verlassen am Ende

Typ logic

Siehe Liste, Objekte

Wenn diese Eigenschaft auf true gesetzt ist, wird automatisch auf das nächste Objekt gewechselt, wenn der Benutzer die maximale Zeichenanzahl erreicht hat.

Je nach Objekt gilt folgendes Verhalten:

- Edit

Das Eingabeobjekt wird verlassen, wenn die maximale Eingabelänge erreicht ist. Ist eine Maskierung definiert, wird die maximale Eingabelänge durch die Länge der Eingabemaske begrenzt. Wird keine Eingabemaske verwendet, ist die maximale Eingabelänge = LengthMax.

- TimeEdit

Das Eingabeobjekt wird verlassen, wenn die durch das Eingabeformat vorgegebene Maximalzahl von Zeichen erreicht wurde.

- DateEdit

Wie beim TimeEdit-Objekt, jedoch kann die Eigenschaft LeaveAtEnd nur dann gesetzt werden, wenn die Eigenschaft InputCheck den Wert WinInpCheckStrict enthält, da ansonsten keine Maximallänge vorhanden ist. Wird dagegen die Eigenschaft auf WinInpCheckLeave gesetzt, wird die Eigenschaft LeaveAtEnd automatisch auf false gesetzt.

- Windowbar

Bei der Windowbar kann definiert werden, ob das Objekt verlassen werden soll, wenn beim Drücken der -Taste das erste bzw. letzte Unterobjekt fokussiert war. Ist die Eigenschaft true (default), wird die Windowbar verlassen, ansonsten verbleibt der Fokus in der Windowbar.

LengthMax

Eingabelänge des Objektes

Typ int

Siehe Liste, Objekte

Über diese Eigenschaft kann die maximale Eingabelänge für das Edit- und das TextEdit-Objekt eingestellt werden.

Je nach Objekt ist folgendes zu beachten:

- Edit

Der gültige Bereich bei 1...32000.

- TextEdit

Enthält die Eigenschaft den Wert 0, ist die Länge des Textes unlimitiert.

Kontakt

LineCapEnd



Pfeiltyp für den Endpunkt von Linien im Canvas-Objekt Typ int

LineCapStart Siehe
Liste, Objekte, WinPropGet(), WinPropSet(),

Diese Eigenschaft definiert den Pfeiltyp für den Endpunkt von Linien (FormType = WinFormTypeLine) im Canvas-Objekt. Folgende Typen sind möglich:

- WinLineCapNoAnchor - Keine spezielle Darstellung für den Endpunkt (default)
- WinLineCapFlat - Der Endpunkt ist flach
- WinLineCapSquare - Der Endpunkt ist quadratisch
- WinLineCapRound - Der Endpunkt ist abgerundet
- WinLineCapTriangle - Der Endpunkt ist dreieckig
- WinLineCapSquareAnchor - Der Endpunkt wird mit einem Quadrat versehen
- WinLineCapRoundAnchor - Der Endpunkt wird mit einem Kreis versehen
- WinLineCapDiamondAnchor - Der Endpunkt wird mit einer Raute versehen
- WinLineCapArrowAnchor - Der Endpunkt wird mit einem Pfeil versehen

Kontakt

LineCapStart



Pfeiltyp für den Startpunkt von Linien im Canvas-Objekt Typ int

Siehe
Liste, Objekte, WinPropGet(), WinPropSet(),
LineCapEnd

Diese Eigenschaft definiert den Pfeiltyp für den Startpunkt von Linien (FormType = WinFormTypeLine) im Canvas-Objekt. Folgende Typen sind möglich:

- WinLineCapNoAnchor - Keine spezielle Darstellung für den Startpunkt (default)
- WinLineCapFlat - Der Startpunkt ist flach
- WinLineCapSquare - Der Startpunkt ist quadratisch
- WinLineCapRound - Der Startpunkt ist abgerundet
- WinLineCapTriangle - Der Startpunkt ist dreieckig
- WinLineCapSquareAnchor - Der Startpunkt wird mit einem Quadrat versehen
- WinLineCapRoundAnchor - Der Startpunkt wird mit einem Kreis versehen
- WinLineCapDiamondAnchor - Der Startpunkt wird mit einer Raute versehen
- WinLineCapArrowAnchor - Der Startpunkt wird mit einem Pfeil versehen

LineJustify



Ausrichtung der Linie

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Mit dieser Eigenschaft wird die Ausrichtung von Linien (FormType = WinFormTypeLine) im Canvas-Objekt gesetzt. Je nach Ausrichtung der Linie wird der Inhalt (Text oder Grafik) des CanvasGraphic-Objektes ausgerichtet.

Es gibt folgende Ausrichtungen:

WinJustLeft /

WinJustTop

- Die Linie wird je nach Eigenschaft Vertical links oder oben dargestellt. Der Inhalt des Objektes wird rechts bzw. unterhalb der Linie angezeigt.

WinJustCenter

- Die Linie wird zentriert dargestellt. Der Inhalt des Objektes wird je nach Eigenschaft Vertical links bzw. oberhalb der Linie angezeigt.

WinJustRight /

WinJustBottom

- Die Linie wird je nach Eigenschaft Vertical rechts oder unten dargestellt. Der Inhalt des Objektes wird links bzw. oberhalb der Linie angezeigt.

Kontakt

LineWidth



Breite des CanvasGraphic-Objektes Linie im Canvas-Objekt Typ int

Siehe Liste, Objekte, WinPropGet(), WinPropSet()

Diese Eigenschaft definiert die Breite der Linie von Grafikobjekten mit FormType

WinFormTypeLine im Canvas-Objekt in logischen Einheiten. Für die Umrechnung von Einheiten (z. B. Millimeter, Zentimeter) in logische Einheiten kann der Befehl PrtUnitLog() verwendet werden.

Kontakt

LinkText

Aktion, die beim Anklicken auszuführen ist

Typ alpha(8192)
Liste, Objekte, WinPropGet(),
Siehe

Wenn der Benutzer auf das Objekt klickt, wird die angegebene Aktion ausgeführt.

Als Aktionen können Programme, URLs, sowie registrierte Dateitypen angegeben werden:

Beispiele:

```
'C:\Windows\Explorer.exe' 'http://www.vectorsoft.de/' 'ftp://ftp.vectorsoft.de/' 'mailto:orga@vector
```

Kontakt

Locale

Länderspezifische Darstellung zuweisen

Typ int

Liste, Objekte, WinPropGet(),

Siehe WinPropSet(), PrtPropGet(),
PrtPropSet()

In dieser Eigenschaft kann der Deskriptor eines mit LocaleLoad() geladenen Locale-Objekts angegeben werden. Die Darstellung des Inhaltes in diesem Objekt erfolgt dann nach den Einstellungen in diesem Locale-Objekt. Bei den Objekten Axis und Statusbar-Button bezieht sich die Darstellung auf die Makrodefinitionen DATE und TIME.

Wird diese Eigenschaft auf 0 gesetzt, erfolgt die Anzeige in den länderspezifischen Einstellungen des Betriebssystems.



Da der Deskriptor erst zur Laufzeit bekannt ist, kann diese Eigenschaft nicht im Designer gesetzt werden.

Kontakt

Location

Drucker-Standort

Typ alpha(8192)

Liste,

Siehe Objekte,

PrtPropGet()

Die Eigenschaft liefert eine Zeichenkette mit einer Kurzbeschreibung des Druckerstandorts, z. B. 'Bldg. 38, Room 1164'.

Kontakt

LstFlags

Optionen

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Über diese Eigenschaft kann das Verhalten des RecList- bzw. des DataListPopup-Objekts gesteuert werden.

Es gibt folgende Einstellungen:

WinLstRecFocusTermReset Übertrag des selektierten Satzes in die Feldpuffer beim

Verlassen der Liste

WinLstRecSelectBuffer

Übertrag des selektierten Satzes in die Feldpuffer beim

EvtLstSelect

WinLstLockRefresh

Unterbinden aller Refresh Anfragen

WinLstSystemEvtSelect

Während der Verarbeitung des Ereignisses EvtLstSelect

können weitere Ereignisse verarbeitet werden.

Bei einem DataListPopup (bei Editierung in DataList, RecList und RecView) kann die Option WinLstEditClose gesetzt werden. Nach Übernahme eines Eintrags aus der DataListPopup, wird das Edit-Objekt verlassen (WinLstEdit() / WinRvwEdit()).

Die Flags können wie folgt überprüft und gesetzt werden:

```
// Flag auswerten: if ($rls->wpLstFlags & _WinLstRecFocusTermReset > 0){ ...}// Flag setzen $rls
```

LstStyle

Darstellung Spaltentitel

Typ int

Siehe Liste, Objekte

Über diese Eigenschaft wird festgelegt, ob Spaltentitel dargestellt werden sollen. Zusätzlich kann definiert werden, ob die Reihenfolge der Spalten vom Anwender geändert werden kann und ob das Anklicken eines Spaltentitels zulässig ist.

Es gibt folgende Einstellungen:

WinLstNoHeader Keine Spaltentitel

WinLstHeader Spaltentitel werden dargestellt

WinLstHeaderOrder Reihenfolge der Spalten kann geändert werden

WinLstHeaderStatic Spaltentitel kann nicht angeklickt werden

MarginLeft

Linker Randabstand

Typ **int**

Siehe [Liste, Objekte](#)

Diese Eigenschaft ist bei einigen Oberflächen- und Druck-Objekten vorhanden. Die Bedeutung der Eigenschaft ist unterschiedlich:

- **Eingabe-Objekte**

Die Eigenschaft legt den Abstand (in Pixel) des Inhaltes eines Eingabe-Objekts zum linken Rand fest.

Bei Angabe von 0 besteht kein Abstand zum linken Rand des Eingabe-Objekts. Bei Angabe von -1 wird der Abstand verwendet, der in dem gewählten [Font](#) als Standard definiert ist.

- **Druck-Objekte**



Diese Eigenschaft wurde bei den Druck-Objekten durch [AreaMarginLeft](#) abgelöst und sollte nicht mehr verwendet werden.

Die Eigenschaft definiert den linken Seitenrand in [logischen Einheiten](#).

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle [WinPropGet\(\)](#) / [PrtPropGet\(\)](#) und [WinPropSet\(\)](#) / [PrtPropSet\(\)](#) gelesen bzw. geändert werden.

Beispiele:

```
tEdit->wpMarginLeft # 12;
```

MaxBigInt

Maximaler Eingabewert des Objekts

Typ [bigint](#)

Siehe [Liste, Objekte](#)

In dieser Eigenschaft wird der maximale Wert eingetragen, der vom Benutzer eingegeben werden darf.

Beispiel:

Um eine Eingabe auf den Bereich 0 - 10 zu beschränken muss die Eigenschaft MaxBigInt auf 10\b und die Eigenschaft [MinBigInt](#) auf 0\b gesetzt werden.

Kontakt

MaxDate

Maximaler Eingabewert des Objektes

Typ date

Siehe Liste, Objekte

In dieser Eigenschaft wird der maximale Wert eingetragen, der vom Benutzer eingegeben werden darf.

Beispiel:

Um eine Eingabe auf den Bereich 1.1. - 10.1.2000 zu beschränken, muss die Eigenschaft MaxDate auf 10.1.2000 und die Eigenschaft MinDate auf 1.1.2000 gesetzt werden.

Kontakt

MaxDecimal

Maximaler Eingabewert des Objektes

Typ [decimal](#)

Siehe [Liste, Objekte](#)

In dieser Eigenschaft wird der maximale Wert eingetragen, der vom Benutzer eingegeben werden darf.

Beispiel:

Um eine Eingabe auf den Bereich 0.0 - 10.0 zu beschränken muss die Eigenschaft MaxDecimal auf 10.0|m und die Eigenschaft [MinDecimal](#) auf 0.0|m gesetzt werden.

Kontakt

MaxFloat

Maximaler Eingabewert des Objektes

Typ float
Siehe Liste, Objekte, WinPropGet(),

In dieser Eigenschaft wird der maximale Wert eingetragen, der vom Benutzer eingegeben werden darf.

Beispiel:

Um eine Eingabe auf den Bereich 0.0 - 10.0 zu beschränken muss die Eigenschaft MaxFloat auf 10.0 und die Eigenschaft MinFloat auf 0.0 gesetzt werden.

Kontakt

MaxInt



Maximaler Eingabewert des Objektes / Anzahl Seiten oder Zeichen Typ int

Liste, Objekte, WinPropGet(), WinPropSet(), PrtPropGet(),
PrtPropSet() Siehe

Je nach Objekt hat MaxInt folgende Bedeutung:

- IntEdit

In dieser Eigenschaft wird der maximale Wert eingetragen, der vom Benutzer eingegeben werden darf.

Um zum Beispiel eine Eingabe auf den Bereich 0 - 10 zu beschränken muss die Eigenschaft MaxInt auf 10 und die Eigenschaft MinInt auf 0 gesetzt werden.

- PrtJobPreview

Die Eigenschaft MaxInt liefert die Anzahl der Seiten zurück. Die Eigenschaft kann nur abgefragt werden.

- PrtRtf

Die Eigenschaft MaxInt liefert die Anzahl der Zeichen zurück, die der angegebene RTF-Text besitzt. Die Eigenschaft kann nur abgefragt werden.

Kontakt

MaxTime

Maximaler Eingabewert des Objektes

Typ time
Siehe Liste, Objekte, WinPropGet(),

In dieser Eigenschaft wird der maximale Wert eingetragen, der vom Benutzer eingegeben werden darf.

Beispiel:

Um eine Eingabe auf den Bereich 0 - 10 Uhr zu beschränken, muss die Eigenschaft MaxTime auf 10:0:0 und die Eigenschaft MinTime auf 0:0:0 gesetzt werden.

MemObjHandle

Deskriptor auf ein Memory-Objekt

Typ [handle](#)

Siehe [Liste, Objekte](#)

Befinden sich die Daten eines Bildes in einem [Memory](#)-Objekt, kann das Bild direkt angezeigt werden, indem der Deskriptor auf dieses Objekt in dieser Eigenschaft eingetragen wird. Eine Speicherung als externe Datei oder als Blob ist nicht notwendig.

Damit das Bild aus dem [Memory](#)-Objekt angezeigt wird, muss die Eigenschaft [ImageSource](#) auf [WinImageSourceMemory](#) gesetzt werden.



Auf das übergebene [Memory](#)-Objekt kann anschließend nicht mehr zugegriffen werden.
Damit entfällt die Notwendigkeit das [Memory](#)-Objekt mit [MemFree\(\)](#) freizugeben.

Kontakt

MinBigInt

Minimaler Eingabewert des Objekts

Typ [bigint](#)

Siehe [Liste, Objekte](#)

In dieser Eigenschaft wird der Minimalwert eingetragen, der vom Benutzer eingegeben werden darf (Beispiel: siehe [MaxBigInt](#)).

Kontakt

MinDate

Minimaler Eingabewert des Objekts

Typ date

Siehe Liste, Objekte

In dieser Eigenschaft wird der minimale Wert eingetragen, der vom Benutzer eingegeben werden darf.

Beispiel:

Um eine Eingabe auf den Bereich 1.1. - 10.1.2000 zu beschränken, muss die Eigenschaft MaxDate auf 1.10.2000 und die Eigenschaft MinDate auf 1.1.2000 gesetzt werden.

Kontakt

MinDecimal

Minimaler Eingabewert des Objekts

Typ decimal

Siehe Liste, Objekte

In dieser Eigenschaft wird der Minimalwert eingetragen, der vom Benutzer eingegeben werden darf (Beispiel: siehe MaxDecimal).

Kontakt

MinFloat

Minimaler Eingabewert des Objekts

Typ float

Siehe Liste, Objekte

In dieser Eigenschaft wird der Minimalwert eingetragen, der vom Benutzer eingegeben werden darf (Beispiel: siehe MaxFloat).

Kontakt

MinInt

Minimaler Eingabewert des Objekts

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft wird der Minimalwert eingetragen, der vom Benutzer eingegeben werden darf (Beispiel: siehe MaxInt).

Kontakt

MinTime

Minimaler Eingabewert des Objektes

Typ time

Siehe Liste, Objekte

In dieser Eigenschaft wird der Minimalwert eingetragen, der vom Benutzer eingegeben werden darf (Beispiel: siehe MaxTime).

Kontakt

MobileMode

Unterstützung auf mobilen Geräten

Typ [int](#)

Siehe [Liste](#), [Objekte](#)

Mit dieser Eigenschaft kann die Unterstützung von Oberflächenobjekten hinsichtlich der Verwendung auf mobilen Geräten (Smartphone oder Tablet) verbessert werden.

Folgende Flags können bei der Eigenschaft gesetzt werden:

WinMobileModeOff Die Unterstützung für mobile Geräte ist deaktiviert. Dies ist der Standardwert der Eigenschaft.

WinMobileModeGestures Diese Option verbessert das Scrolling für Listen-Objekte ([DataList](#), [RecList](#), [StoList](#)), [GanttGraph](#) und [DocView](#).

Bei der Zuweisung anderer Werte, wird der Laufzeitfehler [ErrValueInvalid](#) ausgelöst.

ModeColor

Eingabebereich des ColorEdit-Objekts

Typ int

Siehe Liste, Objekte

Diese Eigenschaft bestimmt, welche Farben eingetragen werden können.

Es gibt folgende Konstanten:

WinModeColorSimple Die 16 Standardfarben sind erlaubt

WinModeColorFull Die Standardfarben, Systemfarben und benutzerdefinierten

Farben sind erlaubt

WinModeColorAll Die Standardfarben, Systemfarben, benutzerdefinierten

Farben und _WinColUndefined und _WinColTransparent sind

erlaubt

WinModeColorPalette Die Standardfarben, benutzerdefinierten Farben und eine automatische

Farbe sind erlaubt

Die automatische Farbe wird in der Eigenschaft ColAuto gesetzt.

ModeDraw

Art der Darstellung

Typ int

Siehe Liste, Objekte

Über diese Eigenschaft wird die Art der Darstellung der Grafik festgelegt. Die Eigenschaft wird nur dann berücksichtigt, wenn die Eigenschaften ModeZoom, ZoomFactor, OffsetLeft und OffsetTop auf den Default-Werten stehen.

Es gibt folgende Einstellungen:

WinModeDrawNone Die Ausgabe des Bildes wird nicht beeinflusst (default).

WinModeDrawCenter Zentrierung des Bildes in der Ausgabefläche des Objekts.

WinModeDrawStretch Anpassen des Bildes an die Dimensionen der Ausgabefläche des Objekts.

WinModeDrawTile Kacheln des Bildes über die gesamte Ausgabefläche des Objekts.

WinModeDrawRatio Beibehaltung des Original-Seitenverhältnisses des Objekts.

Kontakt

ModeEffect

Drehung des Bildes oder Dokuments

Typ int

Siehe Liste, Objekte

Mit dieser Eigenschaft kann die Darstellung des Inhalts in einem Objekt beeinflusst werden.

Im Gegensatz zur Eigenschaft PictureMode wird nicht die Position, sondern die Drehung eingestellt. Folgende Konstanten stehen dazu zur Verfügung:

<u>_WinModeEffectNone</u> (0)	Keine Drehung	
<u>_WinModeEffectRot90L</u> (1)	Drehung um 90	Grad gegen den Uhrzeigersinn
<u>_WinModeEffectRot90R</u> (2)	Drehung um 90	Grad im Uhrzeigersinn
<u>_WinModeEffectRot180</u> (3)	Drehung um 180 Grad	
<u>_WinModeEffectRotCustom</u> (4)	Die Drehung wird in der Eigenschaft <u>Rotation</u> angegeben.	

Kontakt

ModeOptimize Optimierte Anzeige von TIFF-Formaten Typ int

Siehe Liste, Objekte

Die Anzeige von TIFF-Bildern ist dahingehend optimiert, dass möglichst viele TIFF-Formate angezeigt werden können. Dafür dauert das Laden großer TIFF-Bilder länger. Insbesondere bei monochromen und Graustufen-TIFFs kann der Unterschied beträchtlich sein. Aus diesem Grund wurde diese Eigenschaft eingeführt. Sie kann folgende Werte annehmen:

- WinModeOptimizeFormat

Dieser Wert ist standardmäßig gesetzt und beinhaltet beim Laden von TIFF-Bildern die größtmögliche Formatunterstützung. Es entsteht im Allgemeinen ein höherer Arbeitsspeicherbedarf, der dazu führt, dass bei Rechnern mit wenig Arbeitsspeicher die TIFF-Anzeige langsamer wird.

- WinModeOptimizeSpeed

Ist dieser Wert gesetzt, werden TIFF-Bilder mit kleinstmöglichem Arbeitsspeicher-Bedarf geladen, wodurch bei Rechnern mit wenig Arbeitsspeicher eine schnellere TIFF-Anzeige möglich ist.

WinModeOptimizeSpeed wirkt sich nur auf monochrome- und Graustufen-TIFFs aus (die mit 1- bzw. 8-bit Farbtiefe gespeichert sind).

Kontakt

ModeZoom

Art der Darstellung

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft kann die Skalierung des Bildes eingestellt werden. Dazu stehen folgende Konstanten zur Verfügung:

WinModeZoomNone Das Bild wird über diese Eigenschaft nicht skaliert. Die Skalierung erfolgt über die Eigenschaft ZoomFactor.

WinModeZoomCursor Der Benutzer kann die Skalierung über Maus und Tastatur vornehmen.

WinModeZoomWidth Das Bild wird auf die gleiche Breite skaliert, wie das Objekt. Je nach Seitenverhältnis wird das Bild unten abgeschnitten oder es entsteht ein freier Bereich.

WinModeZoomHeight Das Bild wird auf die gleiche Höhe skaliert, wie das Objekt. Je nach Seitenverhältnis wird das Bild rechts abgeschnitten oder es entsteht ein freier Bereich.

Kontakt

ModuleUnit Breite des schmalen Elements im

Barcode Typ int

Siehe [Liste, Objekte, PrtPropGet\(\)](#)

Ein Barcode besteht aus mehreren Elementen. Als kleinste Einheit wird das sogenannte schmale Element verwendet. Bestimmte Barcode-Scanner stellen Anforderungen bezüglich der Mindestgröße des schmalen Elements. Diese Größe kann in dieser Eigenschaft ausgelesen werden.

Die Größe des schmalen Elements ist abhängig von der Größe des [PrtBarcode](#)-Objekts und kann deshalb nur gelesen werden. Zum Zeitpunkt des Entwurf kann das Objekt solange vergrößert werden, bis die Mindestanforderung an den Scanner erfüllt sind.

Kontakt

NameComputer

Rechnername Typ

alpha(256)

Liste,

Siehe Objekte,

PrtPropGet()

In dieser Eigenschaft steht der Name des Rechners, auf dem der Druckjob erzeugt wurde.

Die Eigenschaft kann nur gelesen werden.

Beispiel:

```
switch (true){ case (aPpcObject->ppNameComputer == 'Sales') : ... case (aPpcObject->ppNameCo
```

Kontakt

NameDriver

Drucker Treiber-Name

Typ alpha(8192)

Siehe Liste, Objekte,

Mit der Eigenschaft kann der Drucker-Treibername ermittelt werden.

Bei der Druckaufbereitung durch den **CONZEPT 16-Druckertreiber** wird mit dieser Eigenschaft des PpcObjects der Name des Druckertreibers ermittelt, an den die Druckausgabe gesendet wurde.

Die Eigenschaft kann nur gelesen werden.

Kontakt

NamePort

Drucker Port-Name

Typ alpha(8192)
Liste, Objekte,
Siehe

Die Eigenschaft identifiziert den Port, der benutzt wird, um Daten mit dem Drucker auszutauschen (z. B. 'LPT1').

Bei der Druckaufbereitung durch den CONZEPT 16-Druckertreiber wird mit dieser Eigenschaft des PpcObjects der Name des Druckerport ermittelt, an den die Druckausgabe gesendet wurde. Der Druckerport kann in den Eigenschaften des Windows-Druckertreibers definiert und eingestellt werden.

Die Eigenschaft kann nur gelesen werden.

Kontakt

NamePrinter

Druckernname

Typ alpha(8192)

Liste,

Siehe Objekte,

PrtPropGet()

Die Eigenschaft liefert den Name des Druckers, über den ausgedruckt wurde.

Die Eigenschaft kann nur gelesen werden.

Kontakt

NameServer

Servername

Typ [alpha\(8192\)](#)

[Liste](#),

Siehe [Objekte](#),

[PrtPropGet\(\)](#)

Diese Eigenschaft liefert den Server, der den Drucker steuert. Wenn diese Eigenschaft eine leere Zeichenkette zurückliefert, handelt es sich um einen lokalen Drucker.

Kontakt

NameShare

Drucker Share-Name

Typ alpha(8192)

Siehe Liste, Objekte,

Liefert den Namen zurück, unter dem der Drucker freigegeben ist.

Kontakt

NameUser

Name des Benutzers ermitteln

Typ alpha

Liste, Objekte,

Siehe

In dieser Eigenschaft kann der Name des Benutzers ausgelesen werden, der den Druckjob gestartet hat.

Die Eigenschaft kann nur gelesen werden.

Kontakt

NodeDynamic

Dynamischer Knoten

Typ logic

Siehe Liste, Objekte

Bei einem Knoten wird normalerweise nur ein Pluszeichen zum Aufklappen angezeigt, wenn untergeordnete Knoten angehängt sind. Wird die Eigenschaft NodeDynamic auf true gesetzt, wird das Pluszeichen immer angezeigt und es wird automatisch das Ereignis EvtNodeExpand zum Aufklappen aufgerufen.

In diesem Ereignis können dynamisch die Einträge eingefügt werden. Beim Schließen von "dynamischen"-Knoten werden alle untergeordneten Knoten automatisch gelöscht, das kann verhindert werden, indem die Eigenschaft nach dem Anhängen auf false gesetzt wird.

Kontakt

NodeExpanded



Knoten offen / geschlossen

Typ logic

Siehe Liste, Objekte

Sofern die Eigenschaft den Wert true enthält, wird der Knoten aufgeklappt andernfalls geschlossen dargestellt.

Kontakt

NodeStyle Knoten- □

Symbole Typ int

Siehe Liste,
Objekte

Über diese Eigenschaft wird das Symbol für den Knoten festgelegt.

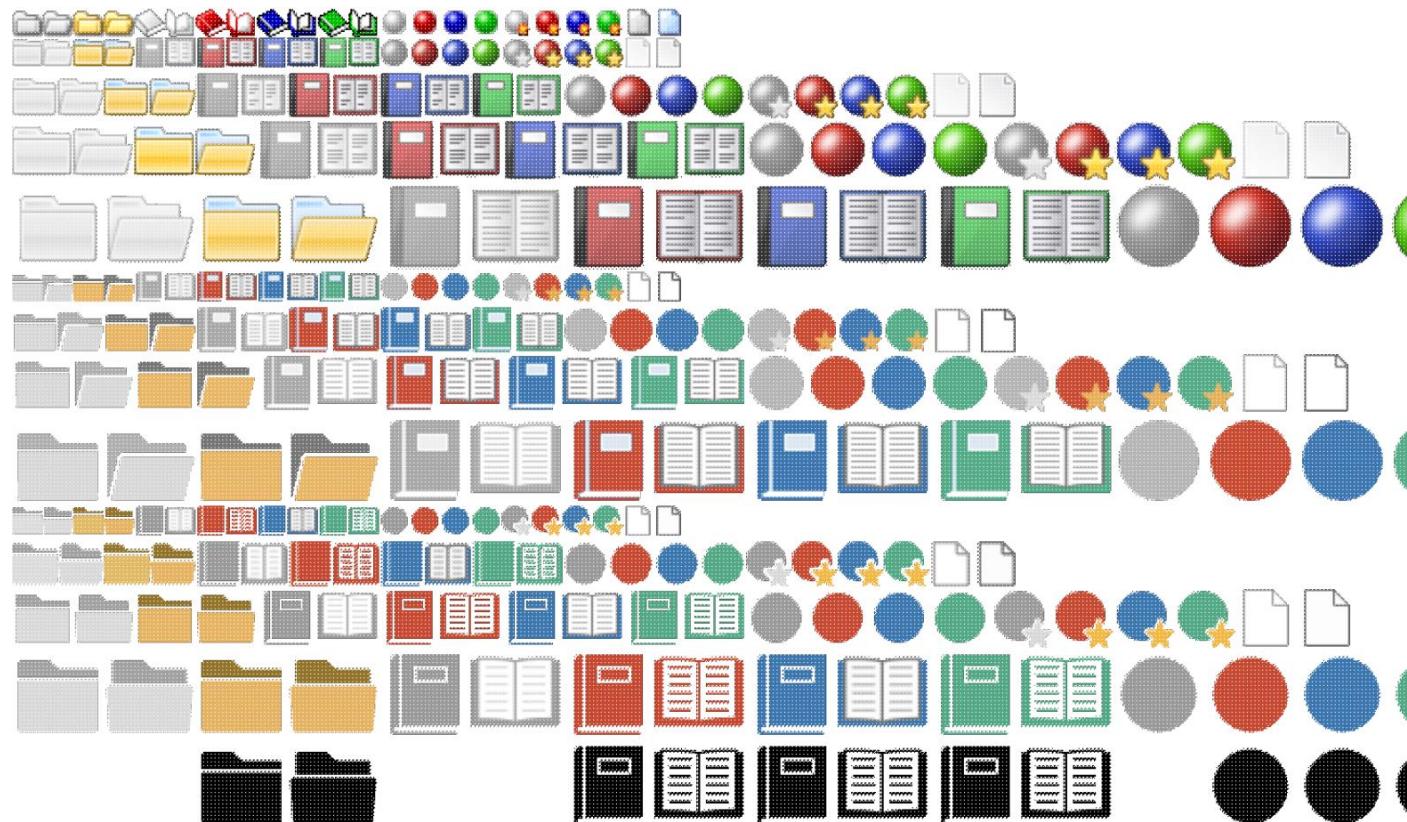
Die Bilder sind in unterschiedlichen Varianten und Abmessungen verfügbar. Die verwendete Variante wird über die Eigenschaft TileTheme und die Abmessung mit TileSize definiert.



Bilder der Variante WinTileThemeFlatMono entsprechen der Variante

WinTileThemeFlatColor mit dem Unterschied, dass nur monochrome Farben verwendet werden. Daher ist in der folgenden Liste nur ein Beispiel für die Größe 48x48 Pixel enthalten. Die Bilder werden zur Laufzeit an die Schriftfarbe der jeweiligen Objektes im ausgewählten Theme angepasst.

Die Eigenschaft NodeStyle kann folgende Werte annehmen. Die jeweils zweite Zeile zeigt die Darstellung mit NodeExpanded = true.



Konstante (Wert)

Classic Enhanced

Modern

FlatCol

16x16 24x24 32x32 48x48 16x16 24x24 32x32 48x48 16x16 2

_WinNodeDoc (7)

_WinNodeFolder (0)

_WinNodeBlueBook

Kontakt

(2)

_WinNodeRedBook

(1)

_WinNodeGreenBook

(3)

_WinNodeBlueBall

(5)

_WinNodeRedBall

(4)

_WinNodeGreenBall

(6)

Kontakt

NotifyParent Funktion

übergeben Typ logic

Siehe Liste, Objekte

Diese Eigenschaft muss gesetzt sein, wenn eine Schaltfläche in ein ToolbarRtf-Objekt eingefügt wird, das in der Eigenschaft MenuAction eine Funktion hinterlegt hat. Die Funktion der Schaltfläche wird dann an das Eltern-Objekt übermittelt.

ObjLink

Verweis auf ein Objekt

Typ alpha(8192)

Liste, Objekte,
WinPropGet(),

Siehe WinPropSet(),
PrtPropGet(),
PrtPropSet()

In dieser Eigenschaft wird der Name des Objektes angegeben, mit dem das Objekt verbunden werden soll. Je nach verwendeten Objekten werden verschiedene Funktionen zur Verfügung gestellt:

- Beim Label-Objekt wird ein Eingabe-Objekt angegeben.

Das Eingabe-Objekt bekommt den Fokus, wenn der Hotkey ( + <unterstrichener Buchstabe>) des Label-Objekts gedrückt wird. Dazu muss in der Eigenschaft Caption des Label-Objekts dem gewünschten Buchstaben das &-Zeichen vorangestellt werden.

- Beim ToolbarRtf-Objekt wird ein RtfEdit-Objekt angegeben.

Über die Werkzeugleiste kann die Formatierung des Textes in dem RtfEdit-Objekt beeinflusst werden.

- Beim RtfEdit-Objekt wird ein ToolbarRtf-Objekt angegeben.



Die Zusammenarbeit zwischen den Objekten RtfEdit und ToolbarRtf ist nur dann gewährleistet, wenn die Eigenschaft ObjLink auf das jeweils andere Objekt gesetzt ist.

Beispiel:

In der Eigenschaft Caption existiert der Eintrag "&Artikel-Nummer". und in der Eigenschaft ObjLink "edART.iNummer". Bei Verwendung der Tastenkombination  +  wird der Eingabefokus auf das Objekt "edART.iNummer" positioniert.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet(), WinPropSet(), PrtPropGet() und PrtPropSet() gelesen bzw. geändert werden.

Kontakt

OffsetLeft

Linke Startposition

Typ [int](#)

Siehe [Liste](#), [Objekte](#)

Mit dieser Eigenschaft kann eine Grafik in horizontaler Richtung verschoben werden. Das Bild wird ab dem angegebenen Koordinaten dargestellt. Alle Angaben sind in Pixel.

Ist die Anzeige der Scrollbars deaktiv, können auch negative Werte verwendet werden.

Kontakt

OffsetTop

Obere Startposition

Typ [int](#)

Siehe [Liste](#), [Objekte](#)

Mit dieser Eigenschaft kann eine Grafik in vertikaler Richtung verschoben werden. Das Bild wird ab den angegebenen Koordinaten dargestellt. Alle Angaben sind in Pixel.

Ist die Anzeige der Scrollbars deaktiv, können auch negative Werte verwendet werden.

Kontakt

OleDropEffectCtrl



Effekt beim Drag & Drop mit

Typ int

Liste, Objekte,
OleDropEffectStandard,

Siehe OleDropEffectShift,

OleDropEffectCtrlShift,
WinPropGet(), WinPropSet()

Mit dieser Eigenschaft kann die durchgeführte Drag & Drop-Operation ausgewählt werden, wenn die Operation mit der Zusatztaste durchgeführt wird.

Folgende Werte können angegeben werden:

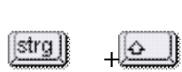
Wert	Bedeutung
<u>WinDropEffectNone</u>	Es wird keine Drag & Drop-Operation durchgeführt.
<u>WinDropEffectCopy</u>	Das gezogene Objekt wird kopiert.
<u>WinDropEffectMove</u>	Das gezogene Objekt wird verschoben.
<u>WinDropEffectLink</u>	Das gezogene Objekt wird verknüpft.

Standardmäßig ist die Eigenschaft auf den Wert WinDropEffectCopy gesetzt.

Die Eigenschaft kann über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

OleDropEffectCtrlShift



Effekt beim Drag & Drop mit
Typ int

Liste, Objekte, OleDropEffectStandard,

Siehe OleDropEffectCtrl, OleDropEffectShift,

WinPropGet(), WinPropSet()

Mit dieser Eigenschaft kann die durchgeführte Drag & Drop-Operation ausgewählt werden, wenn die Operation mit den Zusatztasten + durchgeführt wird.

Folgende Werte können angegeben werden:

Wert	Bedeutung
------	-----------

WinDropEffectNone Es wird keine Drag & Drop-Operation durchgeführt.

WinDropEffectCopy Das gezogene Objekt wird kopiert.

WinDropEffectMove Das gezogene Objekt wird verschoben.

WinDropEffectLink Das gezogene Objekt wird verknüpft.

Standardmäßig ist die Eigenschaft auf den Wert WinDropEffectLink gesetzt.

Die Eigenschaft kann über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

OleDropEffectShift



Effekt beim Drag & Drop mit 

Typ int

Liste, Objekte,
OleDropEffectStandard,

Siehe OleDropEffectCtrl,

OleDropEffectCtrlShift,
WinPropGet(), WinPropSet()

Mit dieser Eigenschaft kann die durchgeführte Drag & Drop-Operation ausgewählt werden, wenn die Operation mit der Zusatztaste  durchgeführt wird.

Folgende Werte können angegeben werden:

Wert	Bedeutung
<u>WinDropEffectNone</u>	Es wird keine Drag & Drop-Operation durchgeführt.
<u>WinDropEffectCopy</u>	Das gezogene Objekt wird kopiert.
<u>WinDropEffectMove</u>	Das gezogene Objekt wird verschoben.
<u>WinDropEffectLink</u>	Das gezogene Objekt wird verknüpft.

Standardmäßig ist die Eigenschaft auf den Wert WinDropEffectMove gesetzt.

Die Eigenschaft kann über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

OleDropEffectStandard



Effekt beim Drag & Drop ohne Zusatztaste

Typ int

Liste, Objekte, OleDropEffectCtrl,

Siehe OleDropEffectShift,

OleDropEffectCtrlShift, WinPropGet(),

WinPropSet()

Mit dieser Eigenschaft kann die durchgeführte Drag & Drop-Operation ausgewählt werden, wenn die Operation ohne Zusatztaste durchgeführt wird.

Folgende Werte können angegeben werden:

Wert	Bedeutung
------	-----------

WinDropEffectNone Es wird keine Drag & Drop-Operation durchgeführt.

WinDropEffectCopy Das gezogene Objekt wird kopiert.

WinDropEffectMove Das gezogene Objekt wird verschoben.

WinDropEffectLink Das gezogene Objekt wird verknüpft.

Standardmäßig ist die Eigenschaft auf den Wert WinDropEffectMove gesetzt.

Die Eigenschaft kann über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Opacity

Deckkraft des Bildes

Typ int

Liste, Objekte,

Siehe WinPropGet(),
WinPropSet()

Die Eigenschaft bestimmt die Deckfähigkeit des angezeigten Bildes bezogen auf den Hintergrund (Eigenschaft ColBkg). Es können Werte zwischen 0 (transparent) bis 100 (deckend) angegeben werden.

Wird in einem Picture-Objekt ein weiteres Picture-Objekt integriert, kann über diese Eigenschaft die Transparenz des untergeordneten Bildes definiert werden.

Kontakt

OrderPass Reihenfolge der Notizbuchseiten oder
Spalten Typ int

Siehe Liste, Objekte, WinPropGet(),

Über diese Eigenschaft kann die Reihenfolge der Notizbuchseiten oder Spalten einer RecList, DataList, StoList oder eines RecView, wie sie der Befehl WinInfo() ermittelt, beeinflusst werden.

Die Eigenschaft kann auf folgende Werte gesetzt werden:

- WinOrderCreate

Der Befehl WinInfo() ermittelt die Notizbuchseiten / Spalten in der Reihenfolge ihrer Erzeugung im Designer.

- WinOrderShow

Der Befehl WinInfo() ermittelt die Notizbuchseiten / Spalten in der Reihenfolge ihrer Anzeige.

Kontakt

Special (Objektspezifische Eigenschaften) - P bis S

Eigenschaften sortiert

nach Gruppen,

Alphabetische Liste

Siehe aller Eigenschaften,

Liste aller

Special-Eigenschaften

PageCount

Anzahl der Seiten

PageNumClm

Anzahl der Spalten für Seitenanzeige

PageFrom

Startseite der Druckausgabe

PageTo

Endeseite der Druckausgabe

PageZoom

Seitenzoom

PathName

Verzeichnis

PdfAuthor

Autor des PDF-Dokumentes

PdfCompression

Kompression des PDF-Dokuments

PdfCreator

Ersteller des PDF-Dokumentes

PdfEncryptKeyLen

Verschlüsselung des PDF-Dokuments

PdfFileName

Dateiname des PDF-Dokumentes

PdfImageResolution

Auflösung der Grafiken

PdfJpegQuality

Qualität der JPEG-Komprimierung

PdfMode

PDF/A-Dokument erstellen

PdfPasswdOpen

Password zum Öffnen des PDF-Dokuments

PdfPasswdOwner

Password des Eigentümers des PDF-Dokuments

PdfRestriction

Einschränkungen der Funktionen im PDF-Dokument

PdfTitle

Titel des PDF-Dokumentes

PicBPP

Farbauflösung des Bildes

PicDpiDefaultX

Horizontale DPI-Standardauflösung von Bildern

PicDpiDefaultY

Vertikale DPI-Standardauflösung von Bildern

PicDpiExifX

Horizontale DPI-Auflösung des Bildes

PicDpiExifY

Vertikale DPI-Auflösung des Bildes

PicDpiHeadX

Horizontale DPI-Auflösung des Bildes

PicDpiHeadY

Vertikale DPI-Auflösung des Bildes

PicDpiX

Horizontale DPI-Auflösung des Bildes

PicDpiY

Vertikale DPI-Auflösung des Bildes

PicHeight

Höhe des Bildes

PicHeightLog

Höhe des Bildes in logischen Einheiten

PictureMode

Art der Darstellung

PictureName

Hintergrundbild

PicWidth

Breite des Bildes

PicWidthLog

Breite des Bildes in logischen Einheiten

PopupFlags

Anzeige-Optionen der Font-Liste

PopupLineHeight

Zeilenhöhe der Drop-Down-Liste

PopupWidth

Breite der Drop-Down-Liste

Kontakt

<u>PreventOverlap</u>	Überschneidung von Intervallen
<u>PrinterDefault</u>	Standard-Drucker
<u>PrinterList</u>	Druckerliste
<u>PrintMode</u>	Druckmodus des Textes
<u>PrintOptions</u>	Druckoptionen
<u>PrintToFile</u>	In eine Datei drucken
<u>Priority</u>	Drucker-Priorität
<u>ProgressMax</u>	Maximaler Wert des Fortschrittsbalkens
<u>ProgressPos</u>	Position des Fortschrittsbalkens
<u>PrtDevice</u>	Drucker-Ausgabeeinheit
<u>PrtRtfFlags</u>	Optionen der Verarbeitung
<u>QualityX</u>	Horizontale Druckauflösung
<u>QualityY</u>	Vertikale Druckauflösung
<u>Radius</u>	Radius der abgerundeten Ecken eines Grafikobjektes im <u>Canvas</u> -Objekt
<u>Range</u>	markierter Bereich
<u>Repeat</u>	Abspielen des Films wiederholen
<u>Rotation</u>	Drehung des Bildes
<u>RtfEditFlags</u>	Optionen der Darstellung
<u>RtfMixMarker</u>	Markierungszeichen für Platzhalter
<u>Ruler</u>	Linealanzeige
<u>SBarStyle</u>	Position der Scrollbar
<u>ScalaLabelCount</u>	Definiert die Anzahl der Beschriftungen auf der Achsenkala
<u>ScalaLabels</u>	Definiert die Beschriftung der Achsenkala
<u>ScalaTextJustify</u>	Ausrichtung der Beschriftung für die Achsenkala
<u>ScalaTextOrigin</u>	Anordnung der Skalen
<u>SearchEnabled</u>	Suche im <u>TreeView</u> aktivieren
<u>SearchFlags</u>	Optionen für die Suche im <u>TreeNode</u>
<u>SearchPattern</u>	Suchtext zur Suche im <u>TreeView</u>
<u>SelData</u>	Deskriptor für <u>SelectionData</u> -Objekt
<u>SelectedVisible</u>	Selektierte Zeile im sichtbaren Bereich
<u>SelectPrt</u>	Seitenauswahl
<u>SelectPvw</u>	Seitenauswahl
<u>ShapeType</u>	Ausrichtung
<u>ShowArrow</u>	Darstellung eines Arrow-Buttons
<u>ShowBorder</u>	Anzeige eines Randes
<u>ShowFocus</u>	Zeichnen des Fokusrechtecks
<u>ShowGrip</u>	Darstellung des Ziehpunktes einer Toolbar
<u>ShowText</u>	Textausgabe in den Intervallen
<u>SignColor</u>	Farbe für Zeichnungsmodus
<u>SignMode</u>	Zeichnungsmodus aktivieren

Kontakt

<u>SignWidth</u>	Liniendicke für Zeichnungsmodus
<u>Silent</u>	Anzeige von Meldungen unterdrücken
<u>Size</u>	Größe / Breite des Objektes
<u>SizeScala</u>	Größe der Skala
<u>SkipPrint</u>	Bedingungen zum Drucken
<u>SoftLineCount</u>	Anzahl der Zeilen
<u>SpellIgnoreAllCapsWords</u>	Wörter, die nur aus Großbuchstaben bestehen, ignorieren
<u>SpellSplitWords</u>	Zusammengesetzte Wörter für Rechtschreibprüfung trennen
<u>SplitFlags</u>	Optionen des GroupSplit-Objekts
<u>SplitStyle</u>	Position des Trenn-Balkens (Split bar)
<u>Status</u>	Druckerstatus
<u>StoClmType</u>	Typ der Spalte
<u>StreamSource</u>	Art der Quelle
<u>StyleButton</u>	Darstellung der Schaltfläche
<u>StyleCloseBox</u>	Aktivität der Schließen-Schaltfläche
<u>StyleConsole</u>	Funktionsumfang des Objektes
<u>StyleCorner</u>	Eckenform der Markierung
<u>StyleDisplay</u>	Darstellung des Kalenders
<u>StyleFrame</u>	Fensterstil
<u>StyleIvl</u>	Darstellung der Intervalle
<u>StylePen</u>	Linientyp der Markierung
<u>StyleRow</u>	Darstellung der Registerreiter-Zeile
<u>StyleShade</u>	Darstellungsart
<u>StyleTab</u>	Darstellung der Registerreiter
<u>SubItemStretch</u>	<u>ClmStretch auch bei SubItems berücksichtigen</u>

PageCount

Anzahl der Seiten

Typ int

Liste, Objekte,

Siehe PrtPropGet(),

WinPropGet(),

WinPropSet()

Diese Eigenschaft gibt beim Picture- und PrtPicture-Objekt die Anzahl der enthaltenen Seiten bzw. Bilder eines MultiPage-TIFFs oder einer ICO-Datei an. Beim ppcObject enthält sie die Anzahl der Seiten des Druckjobs. Die Eigenschaft kann bei diesen Objekten nur gelesen werden.

Beim Objekt PrtJobPreview kann definiert werden, wie viele Seiten ab der in der Eigenschaft CurrentInt angegebenen Startseite maximal angezeigt werden sollen. Diese Eigenschaft kann maximal auf 10 gesetzt werden. Bei diesem Objekt kann die Eigenschaft gelesen und geschrieben werden.

Kontakt

PageNumClm



Anzahl der Spalten für Seitenanzeige

Typ int

Siehe Liste, Objekte

Die Eigenschaft definiert, wie viele Seiten maximal nebeneinander angeordnet werden, wenn ein PrintDoc-Objekt mehrere Seiten besitzt, oder in einem PrtJobPreview- oder DocView-Objekt mehrere Seiten angezeigt werden.

Kontakt

PageFrom



Startseite der Druckausgabe

Typ [int](#)

Siehe [Liste](#), [Objekte](#)

Über die Eigenschaft **PageFrom** wird bei dem Systemdialog [Drucken](#) bestimmt, mit welcher Seite der Ausdruck beginnt.

Kontakt

PageScrollTop Seite beim Seitenwechsel nach oben
scrollen Typ logic

Siehe Liste, Objekte

Wird die Eigenschaft in der Druckvorschau gesetzt, wird beim Seitenwechsel nach oben gescrollt.

Beispiel:

```
// Druckvorschau ermittelntPrtPreview # tPrtJob->PrtInfo(_PrtFrame); // Bei Seitenwechsel oben pos
```

Kontakt

PageTo



Endeseite der Druckausgabe

Typ int

Siehe Liste, Objekte

Über die Eigenschaft PageTo wird bei dem Systemdialog Drucken bestimmt, bis zu welcher Seite der Ausdruck erfolgen soll.

PageZoom

Seitenzoom

Typ int
Liste,
Siehe Objekte

Diese Eigenschaft definiert die Art des Seitenzooms. Für Druck- und für Oberflächen-Objekte stehen unterschiedliche Konstanten zur Verfügung:

- _WinPageZoomUser / _PrtPageZoomUser (0)

Ist diese Konstante bei der Eigenschaft gesetzt, wird das Objekt mit dem Zoomfaktor, der in der Eigenschaft ZoomFactor definiert ist, dargestellt.

- _WinPageZoomPageWidth / _PrtPageZoomPageWidth (1)

Ist diese Konstante bei der Eigenschaft gesetzt, wird der Inhalt des Objekts so gezoomt, dass eine volle Seitenbreite sichtbar ist.

Bei Objekten mit der Eigenschaft PageNumClm wird ein horizontaler Scrollbar dargestellt.

- _WinPageZoomPage / _PrtPageZoomPage (2)

Ist diese Konstante bei der Eigenschaft gesetzt, wird der Inhalt des Objekts so gezoomt, dass genau eine Seite vollständig sichtbar ist.

- _WinPageZoomPageAll / _PrtPageZoomPageAll (3)

Ist diese Konstante bei der Eigenschaft gesetzt, wird so gezoomt, dass alle Seiten des Objekts sichtbar sind.

- _WinPageZoomPageWidthAll / _PrtPageZoomPageWidthAll (4)

Ist diese Konstante bei der Eigenschaft gesetzt, wird so gezoomt, dass alle Seiten im Objekt über die gesamte Breite aller Seiten sichtbar sind.

Bei Objekten mit der Eigenschaft PageNumClm werden die Seiten so gezoomt, dass die dort angegebenen Seiten in den Anzeigebereich passen.

- _WinPageZoomCursor (5)

Diese Ausprägung steht nur bei Oberflächen-Objekten zur Verfügung. Ist diese Konstante bei der Eigenschaft gesetzt, kann die Anzeige über die Tastatur (/) oder die Maus bzw. das Mausrad vom Anwender gezoomt werden.

Kontakt

PathName

Verzeichnis

Typ alpha(8192)

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Über diese Eigenschaft wird das zu verwendende Verzeichnis vorgegeben bzw. das ausgewählte Verzeichnis ermittelt.

Kontakt

PdfAuthor

Verfasser des PDF-Dokumentes

Typ [alpha\(8192\)](#)

Siehe [Liste](#), [Objekte](#), [Beispiel](#)

Über diese Eigenschaft wird bei der Erzeugung eines PDF-Dokuments der Verfasser des Dokumentes gesetzt oder ermittelt (Dokumentzusammenfassung "Verfasser").

PdfCompression

Kompression des PDF-Dokuments

Typ int
Siehe PdfFileName, PrtJobPdf,

Beispiel

Diese Eigenschaft wird nur ausgewertet, wenn aus CONZEPT 16 heraus mit der Anweisung PrtJobClose() ein PDF erzeugt wird.

In dieser Eigenschaft wird die Kompressionsstufe des PDF-Dokuments eingestellt. Es stehen folgende Kompressionsstufen zur Verfügung:

<u>PdfCompressionNone</u>	keine Komprimierung
<u>PdfCompressionZipDefault</u>	Komprimierung mit ZIP-Methoden
<u>PdfCompressionZipFast</u>	Geschwindigkeitsoptimiert
<u>PdfCompressionZipMax</u>	Komprimierungsoptimiert
<u>PdfCompressionJpgDefault</u>	Komprimierung mit JPEG-Mitteln
<u>PdfCompressionJpgFast</u>	Geschwindigkeitsoptimiert
<u>PdfCompressionJpgMax</u>	Komprimierungsoptimiert

Wird eine Komprimierung mit JPEG-Mitteln angegeben, kann die Qualität bei der Komprimierung in der Eigenschaft PdfJpegQuality angegeben werden.

Die Eigenschaft entspricht beim Acrobat PDFWriter der Einstellung der Komprimierung von Text/Vektorgrafik, Farb-/Graustufenbildern und Schwarzweißbildern.

Kontakt

PdfCreator

Ersteller des PDF-Dokumentes

Typ [alpha\(8192\)](#)

Siehe [Liste](#), [Objekte](#), [Beispiel](#)

Über diese Eigenschaft wird der Ersteller des Dokumentes

(Dokumentzusammenfassung "Erstellt mit") gesetzt oder ermittelt.

Kontakt

PdfEncryptKeyLen

Verschlüsselung des PDF-Dokuments

Typ int

PdfPasswdOpen,

Siehe PdfPasswdOwner, PdfFileName,

PrtJobPdf, Beispiel

Diese Eigenschaft wird nur ausgewertet, wenn aus CONZEPT 16 heraus mit der Anweisung PrtJobClose() ein PDF erzeugt wird.

Ein PDF-Dokument kann mit verschiedenen Schlüssellängen verschlüsselt abgespeichert werden. Der verwendete Schlüssel bildet sich dabei aus den in den Eigenschaften PdfPasswdOpen und PdfPasswdOwner angegebenen Passwörtern. Je nachdem mit welcher Version des Acrobat Readers oder Writers das Dokument wieder geöffnet werden soll, stehen unterschiedliche Verschlüsselungen zur Verfügung:

PdfEncryptAcrobat4 40 Bit Verschlüsselung, zu öffnen mit Versionen 4, 5 und 6

PdfEncryptAcrobat5 128 Bit Verschlüsselung, zu öffnen mit Versionen 5 und 6

PdfEncryptAcrobat6 128 Bit Verschlüsselung, zu öffnen mit Version 6

Wird in den Eigenschaften PdfPasswdOpen und / oder PdfPasswdOwner ein Passwort angegeben, erfolgt automatisch eine Verschlüsselung nach dem hier angegebenen Verfahren.

Kontakt

PdfFileName

Dateiname des PDF-Dokumentes

Typ alpha(8192)

Siehe Liste, Objekte, PrtJobClose().

Über diese Eigenschaft wird bei der Ausgabe auf den Acrobat PDF-Writer der Name des zu erzeugenden PDF-Dokumentes definiert. Die Eigenschaft muss ebenfalls angegeben werden, wenn von CONZEPT 16 aus direkt eine PDF-Datei erzeugt werden soll.

Beispiel

```
tHdlPrintJob->ppPdfFileName # _Sys->spPathMyDocuments + '\Rechnung080412.pdf';tErr # tHdlPrintJob
```

Kontakt

PdfImageResolution

Auflösung der Grafiken

Typ int

Siehe [PdfFileName](#),

Siehe [PrtJobPdf](#), [Beispiel](#)

Diese Eigenschaft wird nur ausgewertet, wenn aus CONZEPT 16 heraus mit der Anweisung [PrtJobClose\(\)](#) ein PDF erzeugt wird.

In dieser Eigenschaft wird die Grafikauflösung innerhalb des PDF-Dokuments angegeben. Es kann eine Auflösung zwischen 75 und 2400 dpi angegeben werden. Eine Auflösung von 75 dpi eignet sich nur als Darstellung auf einem Monitor. Der Standardwert beträgt 150 dpi.

PdfJpegQuality

Qualität der JPEG-Komprimierung

Typ int

PdfCompression,

Siehe PdfFileName, PrtJobPdf,

Beispiel

Diese Eigenschaft wird nur ausgewertet, wenn aus CONZEPT 16 heraus mit der Anweisung PrtJobClose() ein PDF erzeugt wird.

Wird in der Eigenschaft PdfCompression eine Methode zur Komprimierung mit JPEG-Mitteln angegeben, kann in dieser Eigenschaft die Qualität der Komprimierung angegeben werden. Die Qualität wird mit einem Wert zwischen 0 und 100 angegeben, was der Bildqualität gegenüber dem Original in Prozent entspricht. Der Standardwert ist 75.

Wird keine JPEG-Komprimierung angegeben, wird die Eigenschaft nicht ausgewertet.

Kontakt

PdfMode

PDF/A-Dokument erstellen

Typ int

Liste, Objekte,

Siehe PrtJobClose(),

PrtPropSet(), Beispiel

Über diese Eigenschaft wird bei der Ausgabe als PDF bestimmt, welche Art von PDF-Dokument erzeugt werden soll. Folgende PDF-Dokumente stehen zur Verfügung:

- PdfModePdfNormal - Normales PDF-Dokument (Default)
- PdfModePdfA - PDF/A-1b:2005 konformes PDF-Dokument für die Langzeit-Archivierung
- PdfModePdfANormal - PDF/A-1b:2005 konformes PDF-Dokument für die Langzeit-Archivierung. Wenn dies nicht gelingt wird bei PrtJobClose() ein normales PDF-Dokument erstellt.
- PdfModePdfA2b - PDF/A-2b:2011 konformes PDF-Dokument für die Langzeit-Archivierung
- PdfModePdfA2bNormal - PDF/A-2b:2011 konformes PDF-Dokument für die Langzeit-Archivierung. Wenn dies nicht gelingt wird bei PrtJobClose() ein normales PDF-Dokument erstellt.
- PdfModePdfA3b - PDF/A-3b:2012 konformes PDF-Dokument für die Langzeit-Archivierung
- PdfModePdfA3bNormal - PDF/A-3b:2012 konformes PDF-Dokument für die Langzeit-Archivierung. Wenn dies nicht gelingt wird bei PrtJobClose() ein normales PDF-Dokument erstellt.

Die Eigenschaft muss ebenfalls angegeben werden, wenn von CONZEPT 16 aus direkt eine PDF-Datei erzeugt werden soll (PrtJobClose(PrtJobPdf)).

Beispiel

```
tHdlPrintJob->ppPdfFileName # _Sys->spPathMyDocuments + '\Rechnung080412.pdf';tHdlPrintJob->ppPdf
```

Kontakt

PdfPasswdOpen Password
zum Öffnen des PDF-
Dokuments Typ alpha

Siehe PdfPasswdOwner, PdfFileName,
PrtJobPdf, Beispiel

Diese Eigenschaft wird nur ausgewertet, wenn aus CONZEPT 16 heraus mit der Anweisung PrtJobClose() ein PDF erzeugt wird.

In dieser Eigenschaft kann ein Passwort, das beim Öffnen des PDF-Dokuments angegeben werden muss, eingetragen werden. Ist ein Passwort vorhanden, wird das Dokument auch verschlüsselt gespeichert. Die Art der Verschlüsselung wird in der Eigenschaft PdfEncryptKeyLen angegeben.

Kontakt

PdfPasswdOwner Password des Eigentümers des PDF-Dokuments Typ alpha

PdfPasswdOpen, PdfFileName, PrtJobPdf, Siehe
Beispiel

Diese Eigenschaft wird nur ausgewertet, wenn aus CONZEPT 16 heraus mit der Anweisung PrtJobClose() ein PDF erzeugt wird.

In dieser Eigenschaft wird das Passwort des Besitzes des PDF-Dokuments angegeben. Das Passwort des Besitzers muss angegeben werden, wenn im Acrobat PDFWriter die Sicherheitseinstellungen des Dokuments verändert werden sollen. Die Verwendung dieses Passwortes ist sinnvoll, wenn innerhalb des Dokuments bestimmte Funktionen (Selektieren, Drucken usw.) unterbunden werden sollen. Die Sicherheitseinstellungen werden in der Eigenschaft PdfRestriction angegeben.

Ist ein Passwort vorhanden, wird das Dokument auch verschlüsselt gespeichert. Die Art der Verschlüsselung wird in der Eigenschaft PdfEncryptKeyLen angegeben.

Kontakt

PdfPrintOrder



Reihenfolge von PrtPdf-Objekten gegenüber anderen Druckobjekten Typ int

Siehe Liste, Objekte, Beispiel

Diese Eigenschaft wird nur ausgewertet, wenn aus CONZEPT 16 heraus mit der Anweisung PrtJobClose() ein PDF erzeugt wird.

Die Eigenschaft definiert, wann die in einer PrintForm enthaltenen PrtPdf-Objekte zum Druckjob hinzugefügt werden. Sie kann folgende Werte annehmen:

- PrtPdfPrintOrderFirst

Zuerst werden die PrtPdf-Objekte ausgegeben. Anschließend erfolgt die Ausgabe der restlichen Druckobjekte (unter Berücksichtigung der Eigenschaft ZOrder).

- PrtPdfPrintOrderLast

Die PrtPdf-Objekte werden zuletzt ausgegeben. Alle anderen Druckobjekte werden zuerst ausgegeben. Dies ist der Standardwert der Eigenschaft.

Kontakt

PdfRestriction Einschränkungen der Funktionen im PDF-Dokument Typ int

Siehe
PdfPasswdOwner, PdfFileName, PrtJobPdf, Beispiel

Diese Eigenschaft wird nur ausgewertet, wenn aus CONZEPT 16 heraus mit der Anweisung PrtJobClose() ein PDF erzeugt wird.

In dieser Eigenschaft werden Beschränkungen in den Funktionen für das PDF-Dokument festgelegt. Damit diese Sicherheitseinstellungen nicht einfach wieder durch einen beliebigen Benutzer entfernt werden können, muss ein Besitzerpasswort (PdfPasswdOwner) definiert werden.

Folgende Beschränkungen können angegeben werden:

<u>PdfDenyNone</u>	keine Beschränkung
<u>PdfDenyAll</u>	alle Beschränkungen
<u>PdfDenyPrint</u>	kein Drucken
<u>PdfDenyModify</u>	keine Änderungen
<u>PdfDenyCopy</u>	kein Kopieren von Text oder Grafiken
<u>PdfDenyComment</u>	keine Anmerkungen
<u>PdfDenyFillForm</u> *	Ausfüllen von Formularfelder unterbinden
<u>PdfDenyExtractText</u> *	kein Text aus PDF extrahieren
<u>PdfDenyAssemble</u> *	Dokument darf nicht mit anderen Dokumenten zusammengestellt werden
<u>PdfDenyPrintHigh</u> *	ausdrucken in hoher Qualität unterbinden
<u>PdfDenyMetaEncrypt</u>	Meta-Daten nicht verschlüsseln

*

Die Konstanten können alle miteinander kombiniert werden. Die Konstanten mit einem Stern (*) werden nur dann ausgewertet, wenn eine 128 Bit-Verschlüsselung des PDF-Dokuments verwendet wird. Diese Verschlüsselung wird durch setzen der Eigenschaft PdfEncryptKeyLen auf PdfEncryptAcrobat5 oder PdfEncryptAcrobat6 erreicht.

Beispiel:

```
// Verhindern von Änderungen und dass das Dokument gedruckt wirdtPrtJob->ppPdfRestriction # _PdfD
```

Kontakt

PdfTitle

Titel des PDF-Dokumentes

Typ alpha(8192)

Siehe Liste, Objekte, Beispiel

Über diese Eigenschaft kann der Titel eines PDF-Dokuments
(Dokumentzusammenfassung "Titel") gesetzt oder abgefragt werden.

Kontakt

PicBPP

Farbauflösung des Bildes

Typ int
Siehe Liste, Objekte,

Über diese Eigenschaft kann die Farbauflösung des Bildes ermittelt werden.

Die Eigenschaft kann nur ausgelesen werden.

Kontakt

PicDpiDefaultX Horizontale DPI-Standardauflösung von
Bildern Typ int

Siehe [Liste](#), [Objekte](#), [WinPropGet\(\)](#), [WinPropSet\(\)](#)

Über diese Eigenschaft kann die horizontale DPI-Standardauflösung von Bildern ermittelt und gesetzt werden. Diese Auflösung wird für Bilder verwendet, die selbst keine DPI-Informationen enthalten.

Der Wert kann im Bereich von 1 bis 65535 liegen.

Kontakt

PicDpiDefaultY **Vertikale DPI-Standardauflösung**
von Bildern Typ int

Siehe [Liste](#), [Objekte](#), [WinPropGet\(\)](#),

Über diese Eigenschaft kann die vertikale DPI-Standardauflösung von Bildern ermittelt und gesetzt werden. Diese Auflösung wird für Bilder verwendet, die selbst keine DPI-Informationen enthalten. Der Wert kann im Bereich von 1 bis 65535 liegen.

Kontakt

PicDpiExifX

Horizontale DPI-Auflösung des Bildes

Typ [int](#)

Siehe [Liste](#), [Objekte](#), [WinPropGet\(\)](#)

Über diese Eigenschaft kann die horizontale DPI-Auflösung des Bildes ermittelt werden. Die Informationen stammen aus den Exif-Informationen der Bild-Datei (nur JPEG).

Die Eigenschaft kann nur ausgelesen werden.

PicDpiExifY

Vertikale DPI-Auflösung des Bildes

Typ [int](#)

Siehe [Liste](#), [Objekte](#), [WinPropGet\(\)](#)

Über diese Eigenschaft kann die vertikale DPI-Auflösung des Bildes ermittelt werden.

Die Informationen stammen aus den Exif-Informationen der Bild-Datei (nur JPEG).

Die Eigenschaft kann nur ausgelesen werden.

Kontakt

PicDpiHeadX

Horizontale DPI-Auflösung des Bildes

Typ [int](#)

Siehe [Liste](#), [Objekte](#), [WinPropGet\(\)](#)

Über diese Eigenschaft kann die horizontale DPI-Auflösung des Bildes ermittelt werden. Die Informationen stammen aus dem Header der Bild-Datei (nur JPEG).

Die Eigenschaft kann nur ausgelesen werden.

PicDpiHeadY

Vertikale DPI-Auflösung des Bildes

Typ [int](#)

Siehe [Liste](#), [Objekte](#), [WinPropGet\(\)](#)

Über diese Eigenschaft kann die vertikale DPI-Auflösung des Bildes ermittelt werden.

Die Informationen stammen aus dem Header der Bild-Datei (nur JPEG).

Die Eigenschaft kann nur ausgelesen werden.

PicDpiX

Horizontale DPI-Auflösung des Bildes

Typ int

Siehe Liste, Objekte, PrtPropGet(),



Diese Eigenschaft dient dazu fehlerhafte Informationen innerhalb des Bildes zu korrigieren.

Das Bild selbst wird dabei nicht verändert. Um das Bild in einer anderen Größe zu Drucken, muss die Eigenschaft AutoSize verwendet werden.

Über diese Eigenschaft kann die horizontale DPI-Auflösung des Bildes ermittelt und gesetzt werden. Die Darstellung des Bildes ist nur von dieser und der Eigenschaft PicDpiY abhängig. Änderungen in dieser Eigenschaft ändert die Eigenschaft PicDpiY auf den gleichen Wert. Existieren für das Bild DPI-Informationen im Header der Datei oder im Exif-Dateiformat, werden diese in der Eigenschaft PicDpiHeadX bzw. PicDpiExifX wiedergegeben.

Die Eigenschaft wird automatisch beim Laden eines Bildes gesetzt. Dabei wird nach der unten angegebenen Reihenfolge vorgegangen. Ist eine Information nicht gesetzt, wird die nächst folgende verwendet.

- Information im Exif-Dateiformat
- Information im Datei-Header
- Default-Einstellungen (PicDpiDefaultX)
- 72 DPI

Beispiel:

Ein JPEG-Bild verfügt über keine Informationen über die Auflösung des Bildes, weder im Header noch in den Exif-Informationen. In diesem Fall wird das Bild mit den Default-Einstellungen beim Application-Objekt geladen. Sind dort keine Angaben gemacht, werden 72 DPI angenommen. Die Auflösung kann nach dem Laden des Bildes durch Setzen der Eigenschaft an die tatsächliche Auflösung des Bildes angepasst werden.

PicDpiY

Vertikale DPI-Auflösung des Bildes

Typ int

Siehe Liste, Objekte, PrtPropSet(),



Diese Eigenschaft dient dazu fehlerhafte Informationen innerhalb des Bildes zu korrigieren.

Das Bild selbst wird dabei nicht verändert. Um das Bild in einer anderen Größe zu Drucken, muss die Eigenschaft AutoSize verwendet werden.

Über diese Eigenschaft kann die vertikale DPI-Auflösung des Bildes ermittelt und gesetzt werden. Die Darstellung des Bildes ist nur von dieser und der Eigenschaft PicDpiX abhängig. Änderungen in dieser Eigenschaft ändert die Eigenschaft PicDpiX auf den gleichen Wert. Existieren für das Bild DPI-Informationen im Header der Datei oder im Exif-Dateiformat, werden diese in der Eigenschaft PicDpiHeadY bzw. PicDpiExifY wiedergegeben.

Die Eigenschaft wird automatisch beim Laden eines Bildes gesetzt. Dabei wird nach der unten angegebenen Reihenfolge vorgegangen. Ist eine Information nicht gesetzt, wird die nächst folgende verwendet.

- Information im Exif-Dateiformat
- Information im Datei-Header
- Default-Einstellungen (PicDpiDefaultY)
- 72 DPI

Beispiel:

Ein JPEG-Bild verfügt über keine Informationen über die Auflösung des Bildes, weder im Header noch in den Exif-Informationen. In diesem Fall wird das Bild mit den Default-Einstellungen beim Application-Objekt geladen. Sind dort keine Angaben gemacht, werden 72 DPI angenommen. Die Auflösung kann nach dem Laden des Bildes durch Setzen der Eigenschaft an die tatsächliche Auflösung des Bildes angepasst werden.

Kontakt

PicHeight Höhe

des Bildes Typ int

Siehe Liste, Objekte,

In dieser Eigenschaft kann die Höhe des Bildes abgefragt werden.

Die Eigenschaft kann nur ausgelesen werden.

Kontakt

PicHeightLog

Höhe des Bildes in logischen Einheiten

Typ int

Siehe Liste, Objekte, PicWidthLog

In dieser Eigenschaft wird die Höhe des dargestellten Dokumentes oder Bildes in logischen Einheiten angegeben. Die Eigenschaft kann nur gelesen werden.

Bei Bildern ist darauf zu achten, dass der Wert aufgrund der im Bild gespeicherten Information (DPI) berechnet wird. Ist diese Information nicht vorhanden wird die Eigenschaft PicDpiDefaultY für die Berechnung herangezogen.

Kontakt

PictureMode

Art der Darstellung

Typ **int**

Siehe **Liste, Objekte**

Über diese Eigenschaft wird die Art der Darstellung der Grafik festgelegt.

Es gibt folgende Einstellungen:

WinPictLeftTop Positionierung des Bildes in der linken oberen Ecke der Ausgabefläche des Objektes.

WinPictCenter Zentrierung des Bildes in der Ausgabefläche des Objektes (default).

WinPictStretch Anpassen des Bildes an die Dimensionen der Ausgabefläche des Objektes.

WinPictTile Kacheln des Bildes über die gesamte Ausgabefläche des Objektes.

WinPictRatio Beibehaltung des Original-Seitenverhältnisses des Objektes.

Kontakt

PictureName

Hintergrundbild

Typ alpha(8192)

Liste,

Siehe Objekte

Über diese Eigenschaft wird einem Fenster ein Hintergrundbild zugeordnet. Die anzuzeigende Grafik kann extern oder in der Datenbank vorliegen und muss vom Typ BMP, JPG, oder TIFF sein. Die Angabe einer externen Datei kann mit einem absoluten oder relativen Pfad erfolgen. Bei der Angabe eines absoluten Pfades wird ein '*', bei einer relativen Pfade (relativ zum Startverzeichnis des CONZEPT 16-Clients) wird ein '!' vorangestellt.

Bei Angabe des Namens ohne Präfix ('!' und '*') wird eine Bilddatei in der Datenbank referenziert. Über die Ressourcenverwaltung lassen sich Grafiken vom Format BMP, JPG und TIFF in die Datenbank einlesen.

Beispiele:

```
'!Picture\Application.tif'           // externe Datei relativ angegeben'*C:\C16\Picture\Application.
```

Mit Hilfe der Eigenschaft PictureMode können spezielle Einstellungen für die Anzeige des Hintergrundbildes vorgenommen werden.

Ist bei einem Frame-Objekt ein Hintergrundbild gesetzt, erscheinen Unterobjekte, die ansonsten die Hintergrundfarbe des Frame-Objekts ausgeben würden (Eigenschaft ColBkg = WinColParent) transparent. Die Transparenz wird von folgenden Objekten unterstützt:

- Checkbox, Divider, Groupbox, Icon, Label

Der Hintergrund wird transparent dargestellt, wenn die Eigenschaft ColBkg auf WinColParent gesetzt ist.

- Notebook

Der Bereich in dem die Registerreiter dargestellt werden, wird transparent dargestellt, wenn die Eigenschaften ColBkgTabArea und ColBkg auf WinColParent gesetzt sind.



Die Transparenz wird nur dann sichtbar, wenn auch alle übergeordneten Objekte bis hin zum Frame-Objekt die Eigenschaft ColBkg auf WinColParent gesetzt haben.

Kontakt

PicWidth

Breite des Bildes

Typ int

Siehe Liste, Objekte,

In dieser Eigenschaft kann die Breite des Bildes abgefragt werden.

Die Eigenschaft kann nur ausgelesen werden.

Kontakt

PicWidthLog Breite des Bildes in logischen Einheiten Typ int

Siehe Liste, Objekte, PicHeightLog

In dieser Eigenschaft wird die Breite des dargestellten Dokumentes oder Bildes in logischen Einheiten angegeben. Die Eigenschaft kann nur gelesen werden.

Bei Bildern ist darauf zu achten, dass der Wert aufgrund der im Bild gespeicherten Information (DPI) berechnet wird. Ist diese Information nicht vorhanden wird die Eigenschaft PicDpiDefaultX für die Berechnung herangezogen.

PopupFlags

Anzeige-Optionen der Font-Liste

Typ int
Siehe Liste, Objekte, WinPropGet(),

Mit dieser Eigenschaft wird bestimmt, welche Fonts im FontNameEdit-Objekt angezeigt werden. Folgende Ausprägungen stehen zur Verfügung:

- WinPopupFontCharsetAnsi

In der Liste werden nur ANSI-Zeichensätze angezeigt (default-Einstellung).

- WinPopupDrawFont

Der Name des Fonts wird mit dem entsprechenden Font geschrieben.

- WinPopupDrawFontType

Vor dem Font-Namen wird durch ein Symbol der Font-Typ angezeigt:

Symbol Bedeutung

Raster- oder Vector-Font

 OpenType-Font

 TrueType-Font

 Adobe Type-1-Font

PopupLineHeight

Zeilenhöhe der Drop-Down-Liste

Typ int
Siehe Liste, Objekte, WinPropGet(),

In dieser Eigenschaft wird die Zeilenhöhe der Drop-Down-Liste der Objekte FontNameEdit und FontSizeEdit angegeben. Die Angabe erfolgt in Pixeln. Ist die Eigenschaft auf 0 gesetzt, wird die Höhe der eingestellten Schriftart verwendet.

Das Setzen dieser Eigenschaft ist insbesondere dann sinnvoll, wenn die Option WinPopupDrawFont der Eigenschaft PopupFlags angegeben ist.

PopupWidth

Breite der Drop-Down-Liste

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In dieser Eigenschaft wird die Breite der Drop-Down-Liste der Objekte FontNameEdit und FontSizeEdit angegeben. Die Angabe erfolgt in Pixeln. Ist die Eigenschaft auf 0 gesetzt, entspricht die Breite der Liste der Breite des Eingabe-Objekts.

PreventOverlap

Überschneidung von Intervallen

Typ logic

Siehe Liste, Objekte

Wenn diese Eigenschaft auf true gesetzt ist, können sich Interval-Objekte innerhalb des GanttGraph-Objekts nicht überlappen.

Kontakt

PrinterDefault

Standard-Drucker

Typ alpha

Siehe Liste, Objekte

Die Eigenschaft legt den Standard-Drucker fest.

Kontakt

PrinterList

Druckerliste

Typ int
Liste,

Siehe Objekte,
PrtPropGet(),
Blog

Die Eigenschaft liefert den Deskriptor auf das Objekt **PrinterList**. Bei dem Objekt handelt es sich um eine Liste der verfügbaren System-Drucker. Die Eigenschaft kann nur gelesen werden.

Die Systemdrucker werden beim Start des CONZEPT 16-Clients gelesen. Sollten in der Zwischenzeit weitere Drucker installiert worden sein, kann mit dem Parameter PrtListRefresh die Liste aktualisiert werden.

Beispiele:

```
// Anzahl der Drucker ermitteln tPrinterList # _App->ppPrinterList(_PrtListRefresh);tPrinterList
```

Kontakt

PrintMode

Druckmodus des Textes

Typ int
Liste, Objekte,
Siehe PrtPropGet(),
PrtPropSet()

Diese Eigenschaft bestimmt den Druckmodus des Textes:

- _PrtPrintModeText (default)

Text im PDF wird auch als Text an den Drucker ausgegeben. Bei Druck über einen PDF-Drucker enthält das resultierende PDF markierbaren Text, sofern zuvor vorhanden.

- _PrtPrintModeTextAsCurves

Text im PDF wird in Bogensegmente (Kurven) umgewandelt. Bei Druck über einen PDF-Drucker enthält das resultierende PDF keinen markierbaren Text und keine Schriftarten mehr.

Beispiel:

```
// Text in Bogensegmente wandelt PrtPdf->ppPrintMode # _PrtPrintModeTextAsCurves;
```

Kontakt

PrintOptions

Druckoptionen

Typ int

Siehe Liste,

Siehe Objekte

Über diese Eigenschaft werden Druckoptionen festgelegt. Folgende Optionen können gesetzt werden:

_PrtPrintOptionsDefault Keine Optionen gesetzt.

_PrtPrintOptionsClearRtfCache Cache für RTF-Objekte nach jeder gedruckten Seite leeren.

Die Eigenschaft murr vor dem Aufruf von PrtJobClose() gesetzt werden.

Beispiel

```
tJob # PrtJobOpen('', 'myjob.prt', _PrtJobOpenRead);if (tJob > 0) { ... tJob->ppPrintOptions # _
```

Kontakt

PrintToFile

In eine Datei drucken

Typ [alpha\(8192\)](#)

Siehe [Liste, Objekte](#)

Über diese Eigenschaft kann die Druckausgabe über den angegebenen Druckertreiber direkt in eine Datei geschrieben werden, z. B. kann beim Microsoft Office Document Image Writer ein MDI (Microsoft Document Imaging-Format) oder TIFF Dokument erzeugt werden.

In der Eigenschaft wird der Pfad- und Dateiname angegeben. Das Dokument wird dann in diese Datei gedruckt.

Ein PDF-Dokument kann auf andere Weise erstellt werden (siehe [PrtJobPdf](#)).

Kontakt

Priority

Drucker-Priorität

Typ int

Liste,

Siehe Objekte,

PrtPropGet()

Ermitteln der Druckerpriorität.

Die Eigenschaft kann über den Befehl PrtPropGet() gelesen werden.

Kontakt

ProgressMax



Maximaler Wert des Fortschrittbalkens

Typ [int](#)

Siehe [Liste](#), [Objekte](#)

In dieser Eigenschaft wird der maximale Wert eingetragen, den die Eigenschaft [ProgressPos](#) erreichen kann.

Kontakt

ProgressPos



Position des Fortschrittbalkens

Typ int

Siehe Liste, ProgressMax

In dieser Eigenschaft wird die Position des Fortschrittbalkens angegeben, wobei in der Eigenschaft ProgressMax der Maximalwert einzugeben ist.

Kontakt

PrtDevice

Drucker-Ausgabeeinheit

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft wird der Deskriptor eines PrintDevice-Objekts angegeben. Die Besonderheiten des entsprechenden Windows-Druckertreibers können dann bei der Druckvorschau und beim Drucken berücksichtigt werden.

Das Setzen der Eigenschaft sollte vor dem Setzen der Eigenschaft Caption eines PrtJobPreview-Objekts erfolgen. Wird zuerst die Caption gesetzt, erfolgt die Darstellung des Druckjobs mit den Informationen des Standard-Druckers bzw. des Druckers, der bei PrtJobClose() angegeben wurde. Wird anschließend die Eigenschaft PrtDevice gesetzt, erfolgt eine erneute Anzeige mit den neuen Informationen. Dies kann die Ausgabe unnötig verzögern. Die Reihenfolge kann beibehalten werden, wenn die Eigenschaft AutoUpdate vor den Zuweisungen auf false und anschließend wieder auf true gesetzt wird, da dann erst nach beiden Zuweisungen der Druckjob angezeigt wird.

Die Eigenschaft kann über die Befehle WinPropGet() bzw. PrtPropGet() und WinPropSet() bzw. PrtPropSet() gelesen bzw. geändert werden.

Kontakt

PrtRtfFlags



Optionen der Verarbeitung

Typ [int](#)

Siehe [Liste](#), [Objekte](#)

In dieser Eigenschaft kann die Verarbeitung des Objekts geändert werden. Für die Eigenschaft steht die Ausprägung [PrtRtfMix](#) zur Verfügung. Ist diese Option aktiviert wird beim Setzen des Textes der Text mit den Daten gemischt. Weitere Informationen befinden sich im Abschnitt [Text und Daten mischen](#).

QualityX

Horizontale Druckauflösung

Typ int

Liste, Objekte,

Siehe PrtPropGet(),

PrtPropSet()

Die Eigenschaft liefert oder setzt die horizontale Druckauflösung in DPI. Einige Druckertreiber erlauben nicht, dass die vertikale und horizontale Druckauflösung unterschiedlich gesetzt werden. In diesem Fall kann PrtFlagSquare angegeben werden, um beide Druckauflösungen zu setzen.

Die Eigenschaft kann beim Objekt PpcObject nur gelesen werden.

Beispiel:

```
gPrintDevice->ppQualityX(_PrtFlagSquare) # 300;
```

QualityY

Vertikale Druckauflösung

Typ int

Liste, Objekte,

Siehe PrtPropGet(),

PrtPropSet()

Die Eigenschaft liefert oder setzt die vertikale Druckauflösung in DPI. Einige Druckertreiber erlauben nicht, dass die vertikale und horizontale Druckauflösung unterschiedlich gesetzt werden. In diesem Fall kann PrtFlagSquare angegeben werden, um beide Druckauflösungen zu setzen.

Die Eigenschaft kann beim Objekt PpcObject nur gelesen werden.

Beispiel:

```
gPrintDevice->ppQualityY(_PrtFlagSquare) # 300;
```

Kontakt

Radius



Radius der abgerundeten Ecken eines Grafikobjektes im Canvas-Objekt Typ int

Siehe Liste, Objekte, WinPropGet(), WinPropSet()

Diese Eigenschaft definiert den Radius der abgerundeten Ecken von Grafikobjekten mit FormType WinFormTypeRectangle bzw. WinFormTypeSquare im Canvas-Objekt in logischen Einheiten. Für die Umrechnung von Einheiten (z. B. Millimeter, Zentimeter) in logische Einheiten kann der Befehl PrtUnitLog() verwendet werden.

Range

Markierter Bereich

Typ range

Liste, Objekte,

WinPropGet(),

Siehe WinPropSet(),

PrtPropGet(),

PrtPropSet()

Die Komponenten der range definieren den Anfang und das Ende eines Bereiches. Der Beginn des Bereiches liegt hinter dem Zeichen an der Position min und endet hinter dem Zeichen an der Position max.

Die Eigenschaft kann auch verwendet werden, um in einem Eingabe-Objekt einen Bereich zu markieren oder um die Einfügemarkierung an eine bestimmte Textstelle zu setzen. Im letzten Fall wird für min und max der gleiche Wert angegeben. Der gesamte Inhalt kann durch Setzen der Range auf den Wert (0, -1) (zum Beispiel mit RangeMake(0, -1)) markiert werden.

Diese Eigenschaft steht für folgende Objekte zur Verfügung:

- Eingabe-Objekte (Edit, TextEdit, IntEdit usw.)

Ist die Eigenschaft Range gesetzt, kann in der entsprechenden Caption-Eigenschaft des Objekts der markierte Inhalt abgefragt oder gesetzt werden.

- RtfEdit

Die Eigenschaft bestimmt den markierten Bereich im RtfEdit-Objekt. Der markierte Bereich kann in der Eigenschaft Caption ausgelesen oder über entsprechende Eigenschaften formatiert werden.

- PrtRtf

Die Eigenschaft bestimmt, welcher Bereich des RTF-Textes in dem Objekt gedruckt wird. Beim Setzen des Bereiches kann mit der Option RtfRangeUpdate der dargestellte Bereich berechnet werden.

Die Eigenschaft kann beim RtfEdit-Objekt nur über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Beispiele:

```
// Cursor an den Anfang stellentObj->wpRange # RangeMake(0, 0); // Cursor an das Ende stellentObj-
```

Kontakt

Repeat

Abspielen des Films wiederholen

Typ logic
Siehe Liste, Objekte, WinPropGet(),

Wenn diese Eigenschaft auf true gesetzt ist, wird das Abspielen des Films unendlich oft wiederholt.

Rotation

Drehung des Bildes

Typ int

Liste, Objekte,

WinPropGet(),

Siehe WinPropSet(),

PrtPropGet(),

PrtPropSet()

Diese Eigenschaft wird nur dann berücksichtigt, wenn in der Eigenschaft ModeEffect der Wert WinModeEffectRotCustom gesetzt ist. Hier wird der Winkel in Grad angegeben, mit dem der Inhalt gedreht werden soll. Bei negativen Zahlen erfolgt die Drehung gegen den Uhrzeigersinn, bei positiven Zahlen im Uhrzeigersinn.

Bei den Objekten DocView und PrtText können nur Vielfache von 90 Grad angegeben werden. Beim DocView wirkt sich die Einstellung auf alle Seiten im Objekt aus.

Untergeordnete Objekte (Overlays) werden nicht mit gedreht.

RtfEditFlags

Optionen der Darstellung

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

In dieser Eigenschaft können folgende Ausprägungen gesetzt werden:

WinRtfEditLineal

Das Zeilenlineal des Objekts wird aktiviert.

WinRtfEditTabSingle

Das Drücken der -Taste führt nicht mehr zu einem Fokuswechsel, sondern zum Eintragen eines TabStops in das RTF-Dokument. Der Fokuswechsel kann mit  +  durchgeführt werden.

WinRtfEditPlainText

In dem RtfEdit-Objekt kann nur noch eine Formatierung angegeben werden. Tabstops und Text-Ausrichtungen werden nicht unterstützt.

Dateien aus dem Explorer oder vom Desktop werden als Bilder eingefügt. Dateien, die keine Bilder sind, werden weiterhin als Datei-Link hinzugefügt. Mit dieser Option werden zu Windows 7 kompatible Bilder generiert.

WinRtfEditAddPictAsMetafile Bilder aus der Zwischenablage im Windows 7-kompatiblen Format einfügen.

RtfMixMarker

Markierungszeichen für Platzhalter

Typ alpha(1)
Liste, Objekte, WinPropGet(),
Siehe

In dieser Eigenschaft wird das Markierungszeichen für das Text und Daten mischen bei RtfEdit-Objekten definiert. Als Markierungszeichen sollte ein Zeichen gewählt werden, das ausschließlich zur Markierung dient und sonst nicht im Text vorkommt.

Das Markierungszeichen muss in seiner Form auch im RTF-Text (also im "Source-Code" des Textes) vorkommen. Einige Sonderzeichen werden in ihre Tabellenposition im Zeichensatz übersetzt (z. B. '^' in '\b0' oder '\' in '\\'). Solche Zeichen können nicht als Markierungszeichen verwendet werden.

Ist kein Markierungszeichen definiert, wird die Tilde '~' als Markierungszeichen verwendet.

Die Eigenschaft kann über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Ruler

Lineal Anzeige

Typ int

Liste,

Siehe Objekte,

PrtPropGet(),

PrtPropSet()

Die Eigenschaft legt fest, ob und wo im Formular ein Lineal angezeigt wird. Über die Eigenschaft Unit kann die Maßeinheit für das Lineal bestimmt werden.

PrtRulerNone Keine Linealanzeige

PrtRulerLeft Linealanzeige links

PrtRulerTop Linealanzeige oben

SBarStyle

Position der Scrollbar

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Über diese Eigenschaft wird festgelegt, ob und an welcher Stelle die Scrollbar dargestellt wird.

Es gibt folgende Einstellungen:

WinSBarNone Keine Scrollbar

WinSBarHorz horizontale Scrollbar

WinSBarVert vertikale Scrollbar

WinSBarBoth horizontale und vertikale Scrollbar



Beim WebNavigator-Objekt können die Einstellungen WinSBarHorz und WinSBarVert nicht verwendet werden.

Kontakt

ScalaLabelCount | Anzahl der Beschriftung auf der Achsenkala Typ int

Siehe
Liste, Objekte, ScalaLabels, WinPropGet(),
WinPropSet(), PrtPropGet(), PrtPropSet()
In dieser Eigenschaft wird die maximale Anzahl der ScalaLabels angegeben. Die Eigenschaft muss gesetzt werden, wenn die einzelnen Labels über einen Index erreichbar sein sollen. Es können maximal 30.000 Labels definiert werden.

Wird ein größerer Index als in dieser Eigenschaft angegeben, wird ein Laufzeitfehler generiert.

Kontakt



ScalaLabels

Definiert die Beschriftung der Achseneskala.

Typ alpha(8192)

Liste, Objekte, ScalaLabelCount,

Siehe WinPropGet(), WinPropSet(),

PrtPropGet(), PrtPropSet()

Die Eigenschaft kann wie folgt gesetzt werden:

- Kommaseparierte Liste

label1, label2, label3, ...

Jede Einheit i (i >= 0) der Achse bekommt den Text von label i + 1 zugeordnet.

- Indizes

Das Setzen der Labels über Indizes ist nur prozedural möglich.

Beispiel:

```
$axsSample->wpScalaLabels(1) # 'Sample1' // Setzen des 1. LabelstLabel # $axsSample->wpScal
```

Auf diese Weise können bis zu 30.000 Skalenbeschriftungen pro Achse erzeugt werden. Die maximal mögliche Anzahl von Skalenbeschriftungen der Achse muss zuvor mit der Eigenschaft ScalaLabelCount gesetzt werden:

```
$axsSample->wpScalaLabelCount # 1000;
```

Durch diese Anweisung ist die Zuweisung der Indizes 0...1000 an die Eigenschaft ScalaLabels erlaubt. Indizes > 1000 oder < 0 führen in diesem Fall zu einem Laufzeitfehler.

Der Index 0 hat die gleiche Bedeutung wie die Verwendung der Eigenschaft ohne Index.

Beim Ermitteln der Eigenschaft ScalaLabels ohne Index oder mit Index 0 ist zu beachten, dass diese maximal 8192 Zeichen zurückliefert.

Wird eine Skalenbeschriftung über den Index (> 0) gesetzt, darf diese kein Komma enthalten. Die Anweisung \$axsSample->wpScalaLabels(1) # 'A,B' erzeugt einen Laufzeitfehler.

Pro Achse ist außerdem nur ein einziges Makro (z. B. \$(DATE...)) möglich.

Deshalb geschieht die Zuweisung von Makros ausschließlich über die Eigenschaft ScalaLabels ohne Index oder mit Index 0. Die Anweisung \$axsSample->wpScalaLabels(1) # '\$(DATE,1.1.2000,31.12.2000,1)' erzeugt kein Makro.

- Makro

a) \$(DATE, Start-Datum, Ende-Datum, Inc-Tage, [Format])

b) \$(TIME, Start-Zeit, Ende-Zeit, Inc-Zeit, [Format])

c) \$(INT, Start-Wert, End-Wert, Inc-Wert)

Kontakt

zu a)

Die Skala wird in Datumswerte unterteilt. Dabei wird mit Start-Datum begonnen. Der letzte Datums-Wert ist Ende-Datum. Für jede Einheit der Skala wird das Start-Datum um Inc-Tage Anzahl Tage erhöht.

Es werden immer nur die Anzahl der Tage erhöht und nicht etwa der Monat. Das erste Beispiel gibt nur den Monat des zugrundeliegenden Datums aus. Bei einem Inkrement von 31 Tagen verschiebt sich das tatsächliche Datum, da einige Monate weniger als 31 Tage besitzen (1.1.2000, 1.2.2000, 3.3.2000, 3.4.2000, 4.5.2000 usw.). Bei einer großen Anzahl von Spalten kann es durch diese Verschiebung (in diesem Beispiel sechs bis sieben Tage pro Jahr) zu einer falschen Darstellung kommen.

Durch Format wird das Datumsformat festgelegt.

Beispiele:

`$(DATE,1.1.2000,31.12.2000,31,MMMM)`

Erzeugt eine Achse, die vom 1.1.2000 - 31.12.2000 die Monatsnamen anzeigt.

`$(DATE,1.1.2000,31.12.2000,31,MM/yy)`

Erzeugt eine Achse, die Monat/Jahr anzeigt : 01/00 02/00 03/00 ... 12/00.

`$(DATE,1.1.2000,7.1.2000,1,ddd)`

Erzeugt eine Achse, die für die erste Woche des Jahres 2000 die Wochentage anzeigt.

`$(DATE,1.1.2000,7.1.2000,1,ddd., dd.MM.)`

Erzeugt eine Achse, die die Wochentage gefolgt von Tag des Monats und Monat anzeigt: Sa, 01.01. So, 02.01. Mo, 03.01. ... Fr, 07.01.

`$(DATE,1.1.2000,7.1.2000,1,LS)`

Entnimmt Datumsformat aus Ländereinstellung (kurzes Format).

`$(DATE,1.1.2000,7.1.2000,1,LL)`

Entnimmt Datumsformat aus Ländereinstellung (langes Format).

Folgende Formatspezifizierer gibt es:

d Tag des Monats (1..31)

dd Tag des Monats (01..31)

**ddd Tag des Monats (Sa, So, ...) dddd Tag des
Monats ausgeschrieben**

M Monat (1..12) MM

Monat (01..12)

Kontakt

MMM Monat (Jan, Feb, Mrz, ...)

MMMM Monat (Januar, Februar, März, ...)

wKalenderwoche (1...53)

wwKalenderwoche (01...53)

yJahr einstellig

yy Jahr zweistellig mit führenden Nullen yyyy Jahr

vierstellig

LS Ländereinstellung kurz

LL Ländereinstellung lang

zu b)

Analog zu a), jedoch wird eine Zeitbeschriftung erstellt:

Beispiele:

`$(TIME,10:00, 17:00, 01:00, hh:mm)`

Achsenbeschriftung : 10:00 11:00 12:00 01:00 02:00 03:00 04:00 05:00

(Stundenintervalle)

`$(TIME,10:00, 17:00, 01:00, HH:mm)`

Achsenbeschriftung : 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00

(Stundenintervalle)

`$(TIME,10:00, 17:00, 00:30, hh:mm)`

Achsenbeschriftung : 10:00 10:30 11:00 11:30 ... (Halbstundenbeschriftung)

Folgende Formatspezifizierer gibt es:

h Stunde ohne führende Null.

hh Stunde mit führender Null (00..12) HH

Stunden mit führender Null (00..23) m Minute

ohne führende Null.

mm Minute mit führende Null.

s Sekunde ohne führende Null. ss

Sekunde mit führender Null.

t Zeitmarkierung (A/P) (12/24 Uhr) (Ländereinstellung Zeitformat) tt

Zeitmarkierung (AM/PM) (Ländereinstellung Zeitformat)

LS Ländereinstellung

Fehlt "Format" wird das Zeitformat der Ländereinstellung ("LS") übernommen.



Innerhalb der Format-Zeichenkette kann normaler Text angegeben werden. Text in einfachen Hochkommata wird nicht durch das Makro ausgewertet.

`$(DATE,1.1.2000,31.1.2000,1,dddd', der 'dd.mm.yyyy)`

zu c)

Kontakt

Analog zu a), jedoch wird eine numerische Achsenbeschriftung erstellt:

Beispiele:

```
$ (INT, 1, 100, 5)
```

Achsenbeschriftung : 1 6 11 16 21 ...

```
$ (INT, 100, 1, -5)
```

Achsenbeschriftung : 100, 95, 90, 85, ...

Normalerweise ist die Ausgabe der Beschriftung zyklisch, sofern es sich nicht um ein Makro (z. B. \$DATE) handelt. So erzeugt beispielsweise die Zuweisung tObj->wpScalaLabels # 'A,B,C' die Beschriftung "A,B,C,A,B,C,A,B,C,..." über die gesamte Achse des Gantt-Objekts.

Soll dies verhindert werden, so kann dies per tObj->wpScalaLabels # 'A,B,C~' geschehen.

Dies erzeugt nur die Beschriftung "A,B,C" ohne Wiederholung.

Bei der Verwendung mit Indizes wird die Tilde aus einer vorherigen Zuweisung der Eigenschaft ScalaLabels übernommen:

```
// Zyklische Labels unterbinden$axxSample->wpScalaLabels # '~';// Setzen einzelner Indizes
```

tLabel würde im Beispiel den Wert 'A,B~' enthalten.

Kontakt

ScalaTextJustify



Ausrichtung der Beschriftung für die Achsenkala

Typ int

Siehe

Liste, Objekte, WinPropGet(), WinPropSet(),
PrtPropGet(), PrtPropSet()

Mit dieser Eigenschaft wird die Ausrichtung der Beschriftung für die Achsenkala festgelegt.

Folgende Ausrichtungen sind möglich:

WinTextJustLeft linksbündige Ausrichtung

WinTextJustCenter zentrierte Ausrichtung

WinTextJustRight rechtsbündige Ausrichtung

Kontakt

ScalaTextOrigin



Anordnung der Skalen

Typ int

Liste, Objekte,

Siehe PrtPropGet(),

PrtPropSet()

Enthält die Eigenschaft den Wert PrtTextOriginVisible, wird der Skalentext in Abhängigkeit von der Eigenschaft Justify und vom sichtbaren Teil der Skala ausgerichtet. Dadurch wird erreicht, dass der Skalentext bei der Druckausgabe immer sichtbar ist (sofern ein Teil der Skala sichtbar ist) und nicht aus dem Druckbereich herausgescrollt wird (wie dies normalerweise der Fall wäre; Konstante PrtTextOriginNone).

Kontakt

SearchEnabled



Suche im TreeView aktivieren

Typ logic

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Ist der Wert auf true gesetzt, kann der Benutzer die Suche im TreeView verwenden. Dafür werden die Tastenkombinationen, wie sie in den Eigenschaften SearchKeyStart, SearchKeyNext und SearchKeyPrev definiert sind, verwendet. Der Standardwert dieser Eigenschaft ist false.

Kontakt

SearchFlags



Optionen für die Suche im TreeNode

Typ int

Siehe Liste, Objekte, WinPropGet(),

Mit dieser Eigenschaft wird das Standardverhalten der TreeView-Suche definiert. Hier können folgende Suchoptionen angegeben werden:

- WinTreeNodeSearchCaption

Vergleicht den Suchtext mit Caption-Eigenschaft des Knotens.

- WinTreeNodeSearchCustom

Vergleicht den Suchtext mit Custom-Eigenschaft des Knotens.

- WinTreeNodeSearchCI

Die Groß-/Kleinschreibung wird bei der Suche ignoriert.

- WinTreeNodeSearchLike

Führt einen Wildcard-Vergleich durch.

- WinTreeNodeSearchLikeAuto

Führt einen Wildcard-Vergleich durch. Der Suchtext wird automatisch mit *<Suchtext>* maskiert. Dies entspricht dem Verhalten im Designer.

- WinTreeNodeSearchToken

Führt eine begriffsorientierte Suche durch.

- WinTreeNodeSearchRegEx Vergleich mittels regulären Ausdrücken.

- WinTreeNodeSearchChildrenOnly

Nur untergeordnete TreeNode-Objekte werden durchsucht. Als Elternobjekt für die Suche wird dann das TreeNode-Objekt verwendet, bei dem die Tastenkombination aus SearchKeyStart gedrückt wurde. Als Referenzobjekt gilt das aktuelle TreeNode.

Standardmäßig sind die Optionen WinTreeNodeSearchCaption, WinTreeNodeSearchCI und WinTreeNodeSearchLikeAuto gesetzt.



Die Optionen WinTreeNodeSearchLike, WinTreeNodeSearchLikeAuto, WinTreeNodeSearchToken und WinTreeNodeSearchRegEx können nicht miteinander kombiniert werden.

Ist weder WinTreeNodeSearchCaption noch WinTreeNodeSearchCustom gesetzt wird trotzdem das Ereignis EvtNodeSearch aufgerufen.



Um die Suche im TreeView zu verwenden, muss die Eigenschaft SearchEnabled auf true gesetzt sein.

SearchPattern

Suchtext zur Suche im TreeView

Typ alpha(8192)

Siehe Liste, Objekte, WinPropGet(),

Diese Eigenschaft enthält den Text nach dem im TreeView gesucht werden soll. Bei jeder neuen Suche überschreibt der, vom Benutzer angegebene, Suchtext den Wert dieser Eigenschaft.

Beim Starten der Suche, mittels der in SearchKeyStart definierten Tastenkombination, wird der Wert dieser Eigenschaft in dem Sucheingabefeld angezeigt. Die Eigenschaft kann also auch als Suchvorschlag für den Benutzer verwendet werden.



Um die Suche im TreeView zu verwenden, muss die Eigenschaft SearchEnabled auf true gesetzt sein.

SelData

Deskriptor für SelectionData-Objekt

Typ int

Siehe Liste, Objekte, MultiSelect,

In dieser Eigenschaft befindet sich der Deskriptor des SelectionData-Objekts. Der Deskriptor und das Objekt werden erst dann erzeugt, wenn bei dem betreffenden Objekt die Eigenschaft MultiSelect auf true gesetzt wird. Über das SelectionData-Objekt können die selektierten Objekte ermittelt werden.

Die Eigenschaft kann nur gelesen werden.

Kontakt

SelectedVisible

Selektierte Zeile im sichtbaren Bereich

Typ [logic](#)

Siehe [Liste, Objekte](#)

Über die Eigenschaft kann ermittelt werden, ob die aktuell selektierte Zeile innerhalb des sichtbaren Bereichs der Anzeige liegt oder außerhalb. Sind mehrere Ansichten im Objekt vorhanden, wird die aktive Ansicht verwendet.



Wird die Zeile am unteren Ende der Liste abgeschnitten, gilt sie als nicht sichtbar.

Kontakt

SelectorItem



Item eines RecView-Objektes auswählen oder Auswahl abfragen.

Typ int

Siehe Liste, Objekte, WinPropGet(), WinPropSet(), SelectorSubItem

Mit dem SelectorItem kann man ein Item eines RecViews auswählen oder die aktuelle Auswahl abfragen.

Um Werte in EvtLstGroupInit anzupassen muss diese Eigenschaft auf die Nummer des entsprechenden Items gesetzt werden.

Die Reihenfolge der Items ist wie in der Definition im RecView-Editor.

Die Eigenschaft kann über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

SelectorSubItem



Subitem eines RecView-Objektes auswählen oder Auswahl abfragen.

Typ int

Siehe Liste, Objekte, WinPropGet(), WinPropSet(), SelectorItem

Mit dem SelectorSubItem kann man ein SubItem eines RecViews auswählen oder die aktuelle Auswahl abfragen.

Um Werte in EvtLstGroupInit anzupassen muss diese Eigenschaft auf die Nummer des entsprechenden SubItems gesetzt werden. Vorher muss die Eigenschaft SelectorItem auf das entsprechend übergeordnete Item gesetzt werden.

Die Reihenfolge der SubItems ist wie in der Definition im RecView-Editor.

Die Eigenschaft kann über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

SelectPrt



Seitenauswahl

Typ alpha

Liste,

Siehe Objekte,

PrtPropGet(),

PrtPropSet()

Die Eigenschaft definiert die Auswahl der Seiten, die bei der Verarbeitung durch die PpcPrint-Befehle berücksichtigt werden. Der Eigenschaft kann die Anweisung 'range', 'odd' oder 'even' zugewiesen werden.

- range()

Dieser Eintrag definiert einen Bereich oder eine Aufzählung von Seiten die gedruckt werden soll. In Klammern wird die Aufzählung der Seitennummern, ein Bereich von Seiten oder eine Aufzählung von Seitenbereichen angegeben. Bereichsüberlappungen sind nicht erlaubt und führen zu einem Laufzeitfehler.

```
ppcObject->ppSelectPrt # 'range()' ;           // Alle SeitenppcObject->ppSelectPrt # 'range()
```

- odd()

Dieser Eintrag definiert einen Bereich ungerader Seiten die gedruckt werden sollen.

```
ppcObject->ppSelectPrt # 'odd()' ;           // Alle ungeraden SeitenppcObject->ppSelectPrt # 'od
```

- even()

Dieser Eintrag definiert einen Bereich gerader Seiten die gedruckt werden sollen.

```
ppcObject->ppSelectPrt # 'even()' ;           // Alle geraden SeitenppcObject->ppSelectPrt # 'eve
```

Standardmäßig hat die Eigenschaft den Wert 'range()'. Es werden also alle Seiten verarbeitet.

Kontakt

SelectPvw
Seitenauswahl
Typ alpha



Liste,

Siehe Objekte,

PrtPropGet(),

PrtPropSet()

Die Eigenschaft definiert die Auswahl der Seiten, die bei der Verarbeitung durch die PpcMakePreview-Befehle berücksichtigt werden. Der Eigenschaft kann die Anweisung 'range', 'odd' oder 'even' zugewiesen werden.

- range()

Dieser Eintrag definiert einen Bereich oder eine Aufzählung von Seiten für die eine Vorschau erstellt werden soll. In Klammern wird die Aufzählung der Seitennummern, ein Bereich von Seiten oder eine Aufzählung von Seitenbereichen angegeben. Bereichsüberlappungen sind nicht erlaubt und führen zu einem Laufzeitfehler.

```
ppcObject->ppSelectPvw # 'range()' ;
```

```
// Alle Seiten
```

- odd()

Dieser Eintrag definiert einen Bereich ungerader Seiten für die eine Vorschau erstellt werden soll.

```
ppcObject->ppSelectPvw # 'odd()' ;
```

```
// Alle ungeraden Seiten
```

- even()

Dieser Eintrag definiert einen Bereich gerader Seiten für die eine Vorschau erstellt werden soll.

```
ppcObject->ppSelectPvw # 'even()' ;
```

```
// Alle geraden Seiten
```

Standardmäßig hat die Eigenschaft den Wert 'range()'. Es wird also für alle Seiten eine Vorschau generiert.

Kontakt

ShapeType

Ausrichtung

Typ int

[Liste, Objekte,](#)

Siehe [WinPropGet\(\)](#),

[WinPropSet\(\)](#)

Diese Eigenschaft bestimmt die Ausrichtung und die Form des Dividers.

Es gibt folgende Ausrichtungen:

[WinShapeLine3dHor](#) - Der Divider bildet eine horizontale Trennlinie mit 3D-Effekt

[WinShapeLine3dVer](#) - Der Divider bildet eine vertikale Trennlinie mit 3D-Effekt

[WinShapeLineHor](#) - Der Divider bildet eine horizontale Trennlinie. Die Linie entspricht der Linie des GroupBox-Objekts.

[WinShapeLineVer](#) - Der Divider bildet eine vertikale Trennlinie. Die Linie entspricht der Linie des GroupBox-Objekts.

ShowArrow

Darstellung eines Arrow-Buttons

Typ logic
Siehe Liste, Objekte, WinPropGet(),

Sofern die Eigenschaft den Wert true enthält, wird ein Arrow-Button als letzter darzustellender Button angezeigt, falls nicht alle Buttons einer Toolbar gezeichnet werden können. Ein Klick auf diesen Button öffnet ein Popup-Menü, das die restlichen Buttons darstellt.



Es werden nur die Buttons in dem Popup-Menü dargestellt, deren Caption-Eigenschaft gesetzt ist.

Kontakt

ShowBorder

Anzeige eines Randes

Typ logic

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Mit dieser Eigenschaft wird bestimmt, ob um das NotebookPage-Objekt ein Rahmen gezeichnet wird. Der Rahmen wird gezeichnet, wenn diese Eigenschaft gesetzt ist.

Ist die Eigenschaft StyleTab auf den Wert WinStyleTabTheme gesetzt, wird die Einstellung in ShowBorder ignoriert.

ShowFocus

Zeichnen des Fokusrechtecks

Typ logic

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Sofern die Eigenschaft den Wert false enthält, wird kein Fokusrechteck gezeichnet, andernfalls wird ein Fokusrechteck gezeichnet, wenn das Eingabe-Objekt, das Notebook, der Button oder das Slider-Objekt den Eingabefokus besitzt.

Ist bei einem Eingabe-Objekt mit Populiste die Eigenschaft ReadOnly gesetzt wird standardmäßig ein Fokus-Rechteck gezeichnet. Wird die Eigenschaft ShowFocus auf false gesetzt, wird kein Fokus-Rechteck gezeichnet.



Beim Slider-Objekt wirkt sich diese Eigenschaft nur im Modern Theme Style aus.

In der systemabhängigen Darstellung wird immer ein Fokusrechteck gezeichnet.

Standardmäßig ist die Eigenschaft auf true gesetzt.

Kontakt

ShowGrip



Darstellung eines Ziehpunktes

Typ logic

Liste, Objekte,

Siehe WinPropGet(),
WinPropSet()

Ist die Eigenschaft gesetzt, wird der Ziehpunkt  zum Ziehen des Toolbar-Objektes dargestellt, ansonsten nicht.

Kontakt

ShowText

Textanzeige

Typ logic

Liste,

Objekte,

Siehe WinPropGet(),

WinPropSet(),

PrtPropGet(),

PrtPropSet()

Wenn diese Eigenschaft den Wert false enthält, wird die Anzeige des Textes unterdrückt.

Bei dem Objekt GanttGraph bezieht sich die Eigenschaft auf die Intervall-Objekte.

SignColor

Farbe für Zeichnungsmodus

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet(),

SignMode, SignWidth

Mit dieser Eigenschaft wird die Farbe für den Zeichnungsmodus gesetzt. Der Zeichnungsmodus muss zusätzlich in der Eigenschaft SignMode aktiviert werden.

SignMode

Zeichnungsmodus aktivieren

Typ logic

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet(), SignColor,

SignWidth

Die Eigenschaft aktiviert den Zeichnungsmodus, in dem Unterschriften und Skizzen erstellt werden können.

Ist der Zeichnungsmodus aktiviert, wird bei gedrückter linker Maustaste ein Pfad erstellt und mithilfe der Werte der Eigenschaften SignWidth und SignColor gezeichnet wird.

Die Zeichnung kann mit der Funktion WinPicSaveImage() gespeichert werden.

SignWidth

Liniendicke für Zeichnungsmodus

Typ int

Liste, Objekte, WinPropGet(),

Siehe WinPropSet(), SignMode,

SignColor

Mit dieser Eigenschaft wird die Liniendicke für den Zeichnungsmodus gesetzt. Der Zeichnungsmodus muss zusätzlich in der Eigenschaft SignMode aktiviert werden.

Kontakt

Silent



Anzeige von Meldungsfenstern unterdrücken

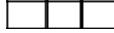
Typ logic

Siehe Liste, Objekte

Über diese Eigenschaft kann gesteuert werden, ob Meldungsfenster im WebNavigator unterdrückt werden sollen (true) oder nicht (false). Standardmäßig ist diese Eigenschaft auf true gesetzt. Ist die Eigenschaft auf true gesetzt, werden beispielsweise JavaScript-Fehler unterdrückt.

Kontakt

Size



Größe / Breite des Objektes

Typ int

Siehe Liste, Objekte

Je nach Objekt hat diese Eigenschaft folgende Bedeutung:

- Axis

Mit dieser Eigenschaft wird die horizontale bzw. vertikale Ausdehnung der Achse (Titel + Skala) festgelegt.

- PrtGanttAxis

Mit dieser Eigenschaft wird die horizontale bzw. vertikale Ausdehnung der Achse (Titel + Skala) festgelegt.

- Toolbar und Statusbar

Höhe (bei Position am oberen oder unteren Rand) bzw. die Breite (bei Position am linken oder rechten Rand) des Toolbar- und Statusbar-Objekts.

Die Höhe bzw. die Breite der Toolbar/Statusbar verändert sich erst, wenn die durch die Schaltflächen vorgegebene Mindesthöhe überschritten wird. Wird ein Wert kleiner 0 angegeben, entspricht die Höhe bzw. die Breite immer dem Absolutwert der Eigenschaft. Die Schaltflächen werden innerhalb der Toolbar bzw. Statusbar zentriert dargestellt, d. h. der obere und untere Rand bzw. der linke und rechte Rand werden gleichmäßig abgeschnitten.

- Toolbar-Button und Statusbar-Button Breite der Schaltfläche.

- Memory

Dies ist die Größe des Speicherbereichs in Bytes, wie sie in MemAllocate() angegeben wurde. Bei einigen Befehlen kann bei der Datenlänge die Konstante MemObjSize übergeben werden, wodurch dann dieser Wert verwendet wird. Die Eigenschaft kann bei diesem Objekt nur gelesen werden.

SizeScala

Größe der Skala

Typ int

Liste, Objekte,

WinPropGet(),

Siehe WinPropSet(),

PrtPropGet(),

PrtPropSet()

Mit dieser Eigenschaft wird die Größe (Höhe bzw. Breite) der Skala eingestellt:

- Wert der Eigenschaft < 0

Die Gesamthöhe der Achse ergibt sich wie bisher. Es wird allein durch die Eigenschaft Size bestimmt. Titel und Skala teilen sich den Platz jeweils zur Hälfte.

- Wert der Eigenschaft >= 0

Der Wert der Eigenschaft Size bestimmt die Titelhöhe und der Wert der Eigenschaft SizeScala die Höhe der Skala.

SkipPrint

Bedingungen zum Drucken

Typ int

Liste, Objekte,

Siehe PrtPropGet(),

PrtPropSet()

In dieser Eigenschaft können folgende Konstanten bzw. Kombinationen davon gesetzt werden:

- PrtSkipPrintEmptyCaption

Ist diese Ausprägung gesetzt, werden alle Objekte innerhalb des PrtLine-Objekts überprüft. Die Zeile wird nicht gedruckt, wenn bei allen PrtText- und PrtBarcode-Objekten die Eigenschaft Caption leer ist und bei allen PrtPicture- und PrtMetaPicture-Objekten entweder die Caption leer ist oder das referenzierte Bild nicht geladen werden konnte.

Ist bei diesen Objekten die Eigenschaft DbFieldName gesetzt, wird der Inhalt des Datenbankfeldes automatisch in die Caption des Objekts übertragen.

Befinden sich weitere Objekte innerhalb der Zeile, wird die Zeile gedruckt. Die Ausnahme bildet dabei das PrtRtf-Objekt. Der Inhalt dieses Objekts wird nicht zur Entscheidung, ob die Zeile gedruckt wird oder nicht, herangezogen.

- PrtSkipPrintEmptyFldName

Ist diese Ausprägung gesetzt, entscheidet der Inhalt der Eigenschaften Caption und DbFieldName darüber, ob eine Zeile gedruckt wird oder nicht. Auch hier werden alle Objekte innerhalb der Zeile untersucht. Die Zeile wird gedruckt, wenn in der Zeile mindestens ein PrtText- oder ein PrtBarcode-Objekt enthalten ist, bei dem beide Eigenschaften gesetzt sind.

Besteht eine Zeile zum Beispiel aus zwei PrtText-Objekten, wobei dem ersten Objekt kein Datenbankfeld zugeordnet, aber in der Caption ein fester Wert eingetragen ist und dem zweiten ein Feld zugeordnet ist, das keinen Inhalt besitzt (die Eigenschaft Caption ist leer), wird die Zeile nicht gedruckt.

Befindet sich innerhalb der Zeile ein PrtRtf-Objekt, werden hier die Eigenschaften DbFieldName, StreamSource und Caption überprüft. Die Zeile wird dann gedruckt, wenn dem Objekt ein Datenbankfeld zugeordnet und die Caption nicht leer ist. Zusätzlich muss die Eigenschaft StreamSource auf WinStreamCaption gesetzt sein, damit der Inhalt des Datenbankfeldes auch in dem Objekt dargestellt wird.

- PrtSkipPrintInvisible

Alle untergeordnete Objekte des PrtLine-Objekts bei denen in der Eigenschaft VisiblePrint das Flag PrtVisiblePrintLine gesetzt ist, werden gedruckt. Alle anderen Objekte werden nicht ausgedruckt.

- PrtSkipPrintAll

Ist diese Konstante gesetzt, wird das PrtLine-Objekt nicht gedruckt.

- PrtSkipPrintNone

Kontakt

Diese Konstante dient zum Zurücksetzen aller Ausprägungen der Eigenschaft SkipPrint.

Ein PrtLine-Objekt, das keine der Konstanten gesetzt hat bzw. auf PrtSkipPrintNone gesetzt wurde, wird immer gedruckt, es sei denn seine Höhe ist 0. Das kann sein, wenn das PrtLine-Objekt keine Unterobjekte besitzt.

Kontakt

SoftLineCount

Anzahl der Zeilen

Typ int

Siehe Liste, Objekte,

Über diese Eigenschaft kann die Anzahl der Zeilen eines Textes (einschließlich der umgebrochenen Zeilen) in einem RtfEdit-Objekt ermittelt werden. Die Eigenschaft kann nur gelesen werden.

Kontakt

SpellIgnoreAllCapsWords



Wörter, die nur aus Großbuchstaben bestehen, ignorieren Typ logic

Siehe

Liste, Objekte, Dictionaries, SpellSplitWords,
WinPropGet(), WinPropSet()

Ist diese Eigenschaft gesetzt, werden Wörter, die nur aus Großbuchstaben bestehen, bei der Rechtschreibprüfung ignoriert.

Kontakt

SpellSplitWords Zusammengesetzte Wörter für Rechtschreibprüfung
trennen Typ logic

Siehe
Liste, Objekte, Dictionaries, SpellIgnoreAllCapsWords,
WinPropGet(), WinPropSet()
Ist diese Eigenschaft gesetzt, werden zusammengesetzte Wörter in die Wordbestandteile zerlegt
und mit den aktiven Wörterbüchern (siehe Dictionaries) verglichen.

Kontakt

SplitFlags



Optionen des GroupSplit-Objekts

Typ int

Siehe Liste, Objekte, WinPropGet(),
WinPropSet()

In dieser Eigenschaft können folgende Optionen gesetzt werden:

- WinSplitCanCloseAll

Diese Option ist standardmäßig gesetzt. Durch Rücksetzen dieses Flags kann verhindert werden, dass alle GroupTile-Objekte vom Benutzer geschlossen werden können. Das letzte GroupTile-Objekt kann nicht vom Benutzer geschlossen werden.

- WinSplitTabSkipChildren

Wird dieses Flag gesetzt, kann durch die GroupTile-Objekte mittels <Tab>-Taste navigiert werden. Untergeordnete Objekte werden ignoriert. Ist diese Flag nicht gesetzt kann durch die GroupTile-Objekte und deren untergeordneten Objekte mittels <Tab>-Taste navigiert werden.

- WinSplitCanSizeOverMax

Sind nur zwei GroupTile-Objekte in dem GroupSplit-Objekt enthalten, wird in der Eigenschaft diese Konstante ausgewertet. Ist die Höhe bzw. Breite eines GroupTile-Objektes durch die Eigenschaft AreaHeightMax (bei Eigenschaft Vertical = true) bzw. AreaWidthMax (bei Eigenschaft Vertical = false) limitiert, kann die Höhe bzw. Breite des GroupSplit-Objekts nicht weiter vergrößert werden, wenn die Maximalhöhe bzw. -breite des GroupTile-Objekts erreicht ist. Durch Setzen der Konstante WinSplitCanSizeOverMax wird diese Limitation aufgehoben.

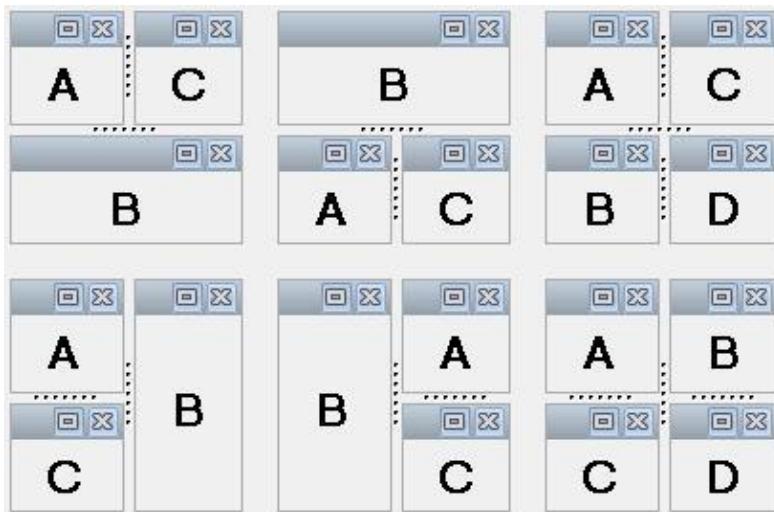
- WinSplitNoCrossOverMove

Das Verschieben von GroupTile-Objekten durch den Benutzer hat zur Folge, dass sich die Größe des verschobenen GroupTiles verändern kann, da dieses seinen Platz mit dem Ziel-GroupTile tauscht.

Um zu erreichen, dass die Größe aller GroupTile-Objekte nicht verändert wird, kann diese Konstante gesetzt werden. Bei zwei GroupTile-Objekten unterliegt die Verschiebeoperation keine Limitation. Bei mehr als zwei GroupTile-Objekten kann eine Vertauschung nur innerhalb eines GroupSplit-Bereiches erfolgen. Es können also nur dann GroupTile-Objekte verschoben werden, wenn die Möglichkeit bestehen bleibt, die Größe beizubehalten.

Beispiele:

Kontakt



Ist für die hier dargestellten GroupSplit-Objekte das Flag gesetzt, können nur die GroupTile-Objekte A mit C und (falls vorhanden) B mit D vertauscht werden.

SplitStyle

Position des Trenn-Balkens (Split bar)

Typ int

Siehe Liste, Objekte, WinPropGet(),

Über diese Eigenschaft wird festgelegt, ob und an welcher Position der Trenn-Balken dargestellt wird.

Es gibt folgende Einstellungen:

WinSplitNone Kein Trenn-Balken

WinSplitHorz horizontaler Trenn-Balken

WinSplitVert vertikaler Trenn-Balken

WinSplitBoth horizontaler und vertikaler Trenn-Balken

Kontakt

Status



Druckerstatus

Typ int

Liste, Objekte,

Siehe PrtDeviceRefresh(),
PrtPropGet()

Die Eigenschaft liefert Informationen zum aktuellen Druckerstatus. Es sind Kombinationen der folgenden Konstanten möglich:

<u>PrtStatusBusy</u>	Drucker ist beschäftigt
<u>PrtStatusDoorOpen</u>	Druckerklappe geöffnet
<u>PrtStatusError</u>	Fehler beim Drucken
<u>PrtStatusInitializing</u>	Drucker-Initialisierung
<u>PrtStatusIoActive</u>	Drucker-Datentransfer aktiv
<u>PrtStatusManualFeed</u>	Manueller Seitenvorschub
<u>PrtStatusNotAvailable</u>	Drucker nicht verfügbar
<u>PrtStatusNotConnected</u>	Drucker nicht verbunden
<u>PrtStatusNoToner</u>	Toner leer
<u>PrtStatusOffline</u>	Drucker Offline
<u>PrtStatusOutOfMemory</u>	Druckerspeicher belegt
<u>PrtStatusOutputBinFull</u>	Ausgabeschacht voll
<u>PrtStatusPagePunt</u>	Seite kann nicht gedruckt werden
<u>PrtStatusPaperJam</u>	Papierstau
<u>PrtStatusPaperOut</u>	Papier leer
<u>PrtStatusPaperProblem</u>	Papierproblem
<u>PrtStatusPause</u>	Drucker angehalten
<u>PrtStatusPendingDelete</u>	Druckjob wird gelöscht
<u>PrtStatusPowerSave</u>	Drucker im Energiesparmodus
<u>PrtStatusPrinting</u>	Drucker druckt
<u>PrtStatusProcessing</u>	Druckjob in Bearbeitung
<u>PrtStatusServerUnknown</u>	Druckserver unbekannt
<u>PrtStatusTonerLow</u>	Toner fast aufgebraucht
<u>PrtStatusUserAction</u>	Druckerfehler aufgetreten
<u>PrtStatusWaiting</u>	Drucker im Wartezustand
<u>PrtStatusWarmingUp</u>	Drucker in der Aufwärmphase

Der Status wird ebenfalls von dem Befehl PrtDeviceRefresh() zurückgegeben.

Kontakt

StoClmType

Typ der Spalte

Typ int,
Liste,

Siehe Objekte

In dieser Eigenschaft wird der Spaltentyp einer Spalte eines StoList-Objekts abgelegt. Der Type wird vom System vorgegeben und kann nicht verändert werden. Jede Spalte besitzt einen eigenen Typ:

StoClmType	Datentyp	Beschreibung
_WinStoClmIcon	<u>int</u>	Symbol des Spaltentyps
_WinStoClmName	<u>alpha</u>	Name des Objekts
_WinStoClmUnicode	<u>logic</u>	Anzeige der Spalte in Unicode
_WinStoClmSize	<u>bigint</u>	Größe des Objekts in Bytes
_WinStoClmLines	<u>int</u>	Anzahl der Zeilen (bei Prozeduren)
_WinStoClmDateModified	<u>date</u>	Datum der letzten Änderung
_WinStoClmTimeModified	<u>time</u>	Uhrzeit der letzten Änderung
_WinStoClmType	<u>alpha</u>	Typ des Objekts
_WinStoClmUserCreated	<u>alpha</u>	Benutzer, der das Objekt erstellt hat
_WinStoClmUserModified	<u>alpha</u>	Benutzer, der das Objekt zuletzt verändert hat
_WinStoClmPerm	<u>bigint</u>	Effektiven Rechte des angemeldeten Benutzers an diesem Objekt

Die Anzahl der Zeilen und der Typ eines Objekts werden nur bei bestimmten Objekten angezeigt.

Kontakt

StreamSource [] |

Art der Quelle

Typ int

Liste,

Objekte,

Siehe WinPropGet(),

WinPropSet(),

PrtPropGet(),

PrtPropSet()

In dieser Eigenschaft wird die Quelle des anzuzeigenden Textes angegeben. Folgende symbolischen Konstanten können in die Eigenschaft eingetragen werden:

- WinStreamNone

Es ist keine Textquelle angegeben. Die Eigenschaft wird auf diesen Wert gesetzt, wenn das entsprechende Objekt zu einer Gruppe gehört. Nur das erste Objekt der Gruppe benötigt die Angabe einer Textquelle.

- WinStreamCaption

Der Text wird aus der Eigenschaft Caption entnommen.

- WinStreamBuffField

Der Text wird einem Datenbankfeld entnommen. Das Feld muss in der Eigenschaft DbFieldName spezifiziert werden.

- WinStreamNameText

Der Text steht in einem internen Text. Der Name des Textes wird in der Eigenschaft FileName abgelegt.

- WinStreamNameFile

Der Text steht in einer externen Datei. Der Pfad und der Dateiname wird in der Eigenschaft FileName abgelegt.

- WinStreamNameBin

Der Text ist als binäres Objekt in der Datenbank gespeichert. Der Pfad und der Dateiname wird in der Eigenschaft FileName abgelegt.

Kontakt

StyleButton



Darstellung der Schaltfläche

Typ [int](#)

Siehe [Liste](#), [Objekte](#)

Zwischen folgenden Darstellungsformen kann gewählt werden:

[WinStyleButtonNormal](#) Die Schaltfläche wird mit Rand dargestellt.

[WinStyleButtonTBar](#) Die Schaltfläche wird ohne Rand dargestellt.

 Ist diese Eigenschaft auf [WinStyleButtonTBar](#) gesetzt, kann die Eigenschaft [Default](#) nicht gesetzt werden. In diesem Fall wird der Button im Windows-Theme gezeichnet, in dem die Default-Darstellung nicht unterstützt wird.

StyleCloseBox

Aktivität der Schließen-Schaltfläche

Typ logic
Siehe Liste, Objekte, WinPropGet(),

Ist die Eigenschaft auf false gesetzt, ist die Schließen-Schaltfläche in der Titelleiste des Fensters und die Menüoption "Schließen Alt+F4" des Systemmenüs deaktiv. Ein Beenden des Fensters ist bei dieser Einstellung nicht möglich. Bei true sind beide aktiv.



Bei der Programmierung ist darauf zu achten, dass nach der Deaktivierung der Schließen-Schaltfläche diese auch wieder aktiviert wird. Erfolgt keine Aktivierung, kann das Fenster nicht mehr geschlossen werden.

StyleConsole

Funktionsumfang des Objektes

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Folgende Funktionsumfänge können eingestellt werden:

WinStyleConRead Funktionen zum Betrachten von Datensätzen

WinStyleConWrite Funktionen zum Verändern von Datensätzen

WinStyleConAll WinStyleConRead und WinStyleConWrite

StyleCorner

Eckenform der Markierung

Typ int

Liste, Objekte,

Siehe PrtPropGet(),

PrtPropSet()

In dieser Eigenschaft wird die Darstellung der Ecken der Markierung definiert. Es können folgende symbolische Konstanten angegeben werden:

PrtStyleCornerRect Darstellung als Rechteck

PrtStyleCornerRound Darstellung als Rechteck mit abgerundeten Ecken

StyleDisplay

Darstellung des Kalenders

Typ int

Liste, Objekte,

Siehe WinPropGet(),
WinPropSet()

Diese Eigenschaft bestimmt das Aussehen des Kalender-Objektes.

Es gibt folgende Einstellung:

WinDisplayWeeknumber Der Kalender wird mit Kalenderwochen angezeigt

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Um die Eigenschaft prozedural im Calendar eines DateEdit-Objekts zu setzen, wird diese beim DateEdit selbst eingetragen. Beim Öffnen des Popups durch den Anwender wird der Wert automatisch an das Calendar-Objekt übergeben.

Kontakt

StyleFrame 

Fensterstil

Typ int
Liste,

Siehe Objekte

Diese Eigenschaft bestimmt die Darstellung des Fenster.

Es gibt folgende Darstellungen:

[WinWndStandard](#)

Normales Fenster

[WinWndDialog](#)

Fenster mit fester Größe

[WinWndFrameHelp](#)

Fenster mit ?-Schaltfläche

[WinWndDialogHelp](#)

Fenster mit fester Größe und ?-Schaltfläche

[WinWndDialogMax](#)

Ein maximierbares, jedoch nicht in der Größe veränderbares

Fenster

[WinWndDialogMin](#)

Ein minimierbares, jedoch nicht in der Größe veränderbares

Fenster

[WinWndDialogMinMax](#)

Ein minimierbares / maximierbares jedoch nicht in der Größe

veränderbares Fenster

[WinWndDialogSize](#)

Ein in der Größe veränderbares Fenster, das nur über eine

Schließen-Schaltfläche verfügt

[WinWndFrameSplash](#)

Fenster ohne Rahmen und Titel (Splash-Screen Darstellung)

[WinWndFrameShape](#)

transparentes Fenster ohne Rahmen und Titel

(Splash-Screen Darstellung)

[WinWndNone](#)

Dieser Stil wird nur gesetzt, wenn die Eigenschaft

StyleBorder geändert wird, da für die Fälle WinBorSunken
und WinBorRaised keiner der vordefinierten Stile zutrifft.

Kontakt

StyleIvl



Darstellung der Objekte
im GanttGraph

Typ int
Liste, Objekte, WinPropGet(),
Siehe WinPropSet(), PrtPropGet(),
PrtPropSet()

In der Eigenschaft wird die Darstellung der Objekte definiert, die innerhalb eines GanttGraph- oder PrtGanttGraph-Objekts angezeigt werden.

Abhängig vom Objekt können folgende Werte in der Eigenschaft gesetzt werden:

- GanttGraph, PrtGanttGraph oder Interval

Über diese Eigenschaft wird festgelegt, ob die Intervallgrenzen mit Pfeilen gekennzeichnet werden.

Es gibt folgende Einstellungen:

WinStyleIvlStandard Darstellung mit Pfeilen

WinStyleIvlNoArrows Darstellung ohne Pfeile

Das Interval-Objekt verfügt zusätzlich über folgende Einstellungen:

WinStyleIvlGantt Übernimmt den Stil, der beim GanttGraph eingestellt ist.

WinStyleIvlOpenStart Zeichnet ein nach rechts offenes Intervall

WinStyleIvlOpenEnd Zeichnet ein nach links offenes Intervall

WinStyleIvlRound Die Enden des Intervalls sind abgerundet

WinStyleIvlHexagon Die Enden des Intervalls sind spitz

- Box

Durch das Setzen der Eigenschaft bei einem Box-Objekt kann statt einer Fläche eine Linie gezeichnet werden. Die Linie wird in der Hintergrundfarbe des Objekts gezeichnet. In der Eigenschaft können folgende Werte eingetragen werden:

0 Es wird eine Fläche gezeichnet (default)

WinStyleIvlBoxHorz Zeichnen einer horizontalen Linie.

WinStyleIvlBoxVert Zeichnen einer vertikalen Linie.

Die Line wird am oberen oder linken Rand des Box-Objekts auf das Grid gezeichnet. Standardmäßig wird das Grid nach der Box gezeichnet und überdeckt somit die Linie (siehe auch WinGanttGridDrawFirst).

StylePen

Linientyp der Markierung

Typ int

Liste, Objekte,

Siehe PrtPropGet(),

PrtPropSet()

In dieser Eigenschaft wird die Linienart der Markierung definiert. Folgende Symbolische Konstanten können angegeben werden:

- PrtStylePenNone Die Markierung wird nur in der Hintergrundfarbe gezeichnet.
- PrtStylePenSolid durchgezogenen Linie

- PrtStylePenBroken unterbrochene Linie

- PrtStylePenDotted ausgefüllte Quadrate

- PrtStylePenCircles ausgefüllte Kreise


Bei den Ausprägungen PrtStylePenBroken, PrtStylePenDotted und PrtStylePenCircles werden die in den Eigenschaften ColFillOdd und ColFillEven eingetragenen Farben zum Füllen verwendet.

Die Breite der Linie ergibt sich aus der Eigenschaft WidthPen.

Kontakt

StyleRow



Darstellung der Registerreiter-Zeile

Typ int

Siehe Liste, Objekte, WinPropGet(),

WinPropSet()

Diese Eigenschaft bestimmt die Darstellung der Zeile der Registerreiter.

Es gibt folgende Darstellungen:

WinStyleRowScroll - Schaltfläche zum Scrollen

WinStyleRowSingle - Einzeilige Darstellung

WinStyleRowMultiLine - Bei Umbruch mehrzeilige Darstellung

WinStyleRowInvisible - Registerreiter verstecken

Kontakt

StyleShade



Darstellungsart

Typ int

Liste, Objekte,

WinPropGet(),

Siehe WinPropSet(),

Betriebssystemdarstellung

(Blog)

Der Fortschrittsbalken kann unterschiedlich dargestellt werden.

Folgende Darstellungsarten gibt es:

WinStyleShadeNone Standarddarstellung

WinStyleShadeFlat Schattierte Darstellung

WinStyleShade3D 3D-Darstellung

WinStyleShadeTheme betriebssystemabhängige Darstellung

StyleTab



Darstellung der Registerreiter

Typ int

Liste, Objekte,

WinPropGet(),

Siehe WinPropSet(),

Betriebssystemdarstellung

(Blog)

Diese Eigenschaft bestimmt die Darstellung des Registereiters.

Es gibt folgende Darstellungen:

WinStyleTabRectangle Rechteckige Registerreittertitel

WinStyleTabSharkFin Haifischflossenähnliche Registerreittertitel

WinStyleTabTheme Registerreittertitel in der betriebssystemabhängigen

Darstellung

WinStyleTabTrapezoid Trapezförmige Registerreittertitel

WinStyleTabRectFlat Flache, rechteckige Registerreittertitel

 Einige Systeme können in der betriebssystemabhängigen Darstellung die Registerkartentitel nur an der oberen Kante ausrichten (Eigenschaft AlignTab mit dem Wert WinAlignInpTop).

SubDivisions

Skalenunterteilung der Achse

Typ alpha(8192)

Liste, Objekte,

SubDivisionsCount,

Siehe WinPropGet(),

WinPropSet(), PrtPropGet(),

PrtPropSet()

Mit dieser Eigenschaft wird die Skalenunterteilung der Achse festgelegt. Ein Wert von 5 bedeutet zum Beispiel, dass fünf Zellen zu einem Achsenabschnitt gehören. Wenn insgesamt 30 Zellen dargestellt werden, werden also sechs Achsenabschnitte angezeigt.

Die letzte untergeordnete Achse besitzt keine Angabe in der Eigenschaft SubDivisions. Die darüber liegenden Achsen ergeben sich somit aus der Angabe in SubDivisions automatisch.

Wenn zum Beispiel zwei horizontale Achsen vorhanden sind und CellSizeHorz auf 32 gesetzt ist, so hat die untergeordnete Achse 32 Pixel pro Zelle (in horizontaler Richtung). Wenn nun die Eigenschaft SubDivisions der Hauptachse einen Wert von 3 enthält, ist jede Zelle der Hauptachse in genau 3 Zellen der Unterachse (also 96 Pixel) unterteilt.

Die Eigenschaft kann wie folgt gesetzt werden:

- Kommaseparierte Liste

Unterschiedlich lange Unterteilungen können in Form einer kommaseparierten Liste angegeben werden. So könnte zum Beispiel die Darstellung der Monate durch folgende Angabe realisiert werden:

Beispiel:

```
$axs->wpSubDivisions = '31,28,31,30,31,30,31,31,30,31,30,31'>
```

- Indizes

Das Setzen der Skalenunterteilungen über Indizes ist nur prozedural möglich.

Um die Unterteilungen über einen Index anzusprechen, muss zuvor die Eigenschaft SubDivisionsCount auf den entsprechenden Maximalwert gesetzt werden. Wird die Eigenschaft SubDivisions ohne Index gesetzt, wird automatisch die Eigenschaft SubDivisionsCount angepasst.

Beispiel:

```
$axs->wpSubDivisionsCount # 10;$axs->wpSubDivisions(1) # '2';$axs->wpSubDivisions(5) # '5'
```

Durch die erste Anweisung ist die Zuweisung der Indizes 0 bis 10 an die Eigenschaft SubDivisions erlaubt. Indizes > 10 oder < 0 führen in diesem Fall zu einem Laufzeitfehler.

Kontakt

Der Index 0 hat die gleiche Bedeutung wie die Verwendung der Eigenschaft ohne Index.

Wird eine Skalenunterteilung über den Index (> 0) gesetzt, darf diese kein Komma enthalten. Die Anweisung \$axsSample->wpSubDivisions(1) # '2,3' erzeugt einen Laufzeitfehler.

Über den Index können maximal 30.000 Einträge definiert werden, ohne Index oder mit Index 0 kann nur eine Zeichenkette mit 8192 Zeichen angegeben werden. Beim Auslesen wird die Zeichenkette bei 8192 Zeichen abgeschnitten.



Weiterhin wird die Zeichenkette bei 8192 Zeichen abgeschnitten, wenn der Dialog mit WinSave() gespeichert, oder die Eigenschaft bei Achsen des PrtGanttGraph-Objektes gesetzt wird.

Kontakt

SubDivisionsCount Anzahl der Skalenunterteilungen der Achse Typ int

Liste, Objekte, SubDivisions,
Siehe WinPropGet(), WinPropSet(),
PrtPropGet(), PrtPropSet()

Mit dieser Eigenschaft wird die maximale Anzahl der Skalenunterteilungen der Achse festgelegt. Die Eigenschaft muss gesetzt werden, wenn die einzelnen Unterteilungen über einen Index erreichbar sein sollen. Es können maximal 30.000 Skalenunterteilungen definiert werden.

Beispiel:

```
$axs->wpSubDivisionsCount # 10;$axs->wpSubDivisions(1) # '2';$axs->wpSubDivisions(5) # '5';// Bew
```

Die Eigenschaft SubDivisionsCount wird automatisch gesetzt, wenn die Eigenschaft SubDivisions ohne Index gesetzt wird.

Wird ein größerer Index als in dieser Eigenschaft angegeben, wird ein Laufzeitfehler generiert.

Kontakt

SubItemStretch ClmStretch auch bei SubItems
berücksichtigen Typ logic

Siehe Liste, Objekte

Diese Eigenschaft bestimmt, ob die Eigenschaft ClmStretch bei SubItems berücksichtigt werden soll (SubItemStretch = true) oder nicht (SubItemStretch = false).

Ein so definiertes SubItem passt sich von der Breite an die Breite des übergeordneten Items an. Die Position des SubItems bleibt dabei aber unverändert. Dadurch kann auf einfache Weise ein automatischer Zeilenumbruch für SubItems realisiert werden.

Kontakt

Special (Objektspezifische Eigenschaften) - T bis Z

Siehe

Eigenschaften sortiert

nach Gruppen,

Alphabetische Liste

aller Eigenschaften,

Liste aller

Special-Eigenschaften

TaskbarButton Anzeige des Fensters in der Taskleiste des Betriebssystems

ThemeBaseName Name des verwendeten Basis-Themes

ThemeBkg Darstellung des Hintergrundes der Werkzeugeiste

Themed Darstellung des Hintergrundes der Titelzeile von GroupSplit- und GroupTile-Objekten

TifFileName Dateiname des TIF-Dokuments

TickAlignment Ausrichtung der Markierungen des Objekts

TickInterval Ausrichtung der Markierungen des Objekts

TickLineSize Anzahl der Positionen, um die der Schieberegler mit den Pfeiltasten verschoben wird

TickPageSize Anzahl der Positionen, um die der Schieberegler mit den Tasten verschoben wird

TifMode Farb-Modus bei Drucken als TIFF-Datei

TileBorder Rahmen des GroupSplit-Objekts

TileTheme Darstellung der ImageTile und NodeStyle

TimeStart Startzeit der Drucker-Verfügbarkeit

TimeUntil Endzeit der Drucker-Verfügbarkeit

TitleJustify Ausrichtung der Beschriftung des Achsentitels

TitleText Titel des Objektes

Toolbars Anzeige von Toolbars

TrayFrame TrayFrame-Applikation zuweisen

TreeFlags Darstellung des TreeView-Objektes

TypeBarcode Typ des zu druckenden Barcodes

TypeButton Aktion der Schaltfläche

TypeFile Filmtyp

Unit Maßeinheit für Lineal

Url Aktuell angezeigte Adresse des WebNavigators

Version Version des Druckertreibers

Vertical Vertikale Darstellung des Balkens

VerticalText Vertikale Darstellung

ViewId Nummer des Views zur Anzeige des Overlays

ViewMode Anzeigemodus

ViewType Bedingungen zur Anzeige des Overlays

WaitCursor Mauszeiger als Sanduhr darstellen

WidthPen Strichstärke

WordBreak Automatischer Umbruch bei zu langem Text

Kontakt

XmlFileName

Dateiname des XML-Dokumentes

ZoomEffective

Vergrößerungsfaktor

ZoomFactor

Benutzerdefinierter Vergrößerungsfaktor

ZOrder

Darstellungsreihenfolge

Kontakt

TaskbarButton



Anzeige des Fensters in der Taskleiste des Betriebssystems Typ int

Siehe Liste, Objekte, WinPropGet(), WinPropSet() Frame-Objekte, die ein Parent-Objekt besitzen, werden von Windows nicht in der Taskleiste dargestellt. Durch setzen der Eigenschaft TaskbarButton auf die Konstante WinTaskbarVisible kann dies verhindert werden, d. h. es wird eine Schaltfläche in der Windows Taskleiste angezeigt.

Durch Setzen der Konstante WinTaskbarDefault wird das Standard-Verhalten wiederhergestellt.

Ein Fenster, dass unsichtbar ist hat auch weiterhin keine Schaltfläche in der Taskbar.

Für MdiFrame- und TrayFrame-Objekte existiert die Eigenschaft nicht.

Kontakt

ThemeBaseName Name zur Identifikation des
Basis-Themes Typ alpha(60)

Siehe Liste, Objekte, WinPropGet(),

Diese Eigenschaft definiert in einem Theme-Objekt den Name des vordefinierten Themes, dass bei der Erstellung des Themes angegeben wurde.

Wenn in zukünftigen Versionen neue Eigenschaften eines bereits existierenden Theme-Elementes hinzukommen, werden diese mit den Werten des vordefinierten Themes initialisiert.

Die Eigenschaft wird automatisch gesetzt:

- Bei der Erstellung über WinThemeOpen()
- Bei der Erstellung im Theme-Editor

Die Eigenschaft kann nur gelesen werden.

Kontakt

ThemeBkg Darstellung des Hintergrundes der Werkzeugleiste Typ logic

Siehe
[Liste](#), [Objekte](#), [WinPropGet\(\)](#), [WinPropSet\(\)](#),
[Betriebssystemdarstellung \(Blog\)](#)
Wird diese Eigenschaft auf true gesetzt, wird der Hintergrund der Werkzeugleiste im entsprechenden Stil des Betriebssystems gezeichnet. Wird die Eigenschaft zurückgesetzt, wird der Hintergrund im CONZEPT 16-Stil gezeichnet.

Zum Hintergrund gehört auch der "Griff" der Werkzeugleiste, sowie die Pfeilschaltfläche, die angezeigt wird, wenn nicht alle Schaltflächen der Werkzeugleiste sichtbar sind.

Die Eigenschaft kann unter allen Betriebssystemen gesetzt werden, hat aber nur Auswirkungen, wenn das Betriebssystem über spezifische Darstellungen verfügt. Unter anderen Betriebssystemen hat das Setzen der Eigenschaft keinen Einfluß auf die Darstellungsweise der Werkzeugleiste.

Kontakt

Themed



Darstellung des Hintergrundes der Titelzeile von GroupSplit- und GroupTile-Objekten

Typ logic

Siehe Liste, Objekte, WinPropGet(), WinPropSet()

Wird diese Eigenschaft auf true gesetzt, dann wird die optional vorhandene Titelleiste des GroupSplit-Objektes und der untergeordneten GroupTile-Objekte im Stil des aktuell verwendeten Betriebssystems dargestellt. Bei false wird wieder die durch CONZEPT 16 vorgegebene Titelleiste gezeichnet.

Die Eigenschaft kann unter allen Betriebssystemen gesetzt werden, hat aber nur Auswirkungen, wenn die folgenden Bedingungen erfüllt sind:

- Das Betriebssystem ist mindestens Windows Vista
- Die Farbtiefe des Bildschirms ist mindestens 256 Farben
- Visuelle Themes des Betriebssystems müssen aktiviert sein
- Unter Windows Vista / Windows 7 muss die Desktop Composition aktiviert sein

ThemeDescription

Beschreibung des ThemeSets

Typ alpha(100)

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Mit dieser Eigenschaft kann eine Beschreibung für ThemeSets, hinterlegt werden.

Die Eigenschaft kann im Theme-Editor mit Doppelklick auf den Spaltenkopf eines ThemeSets oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Beispiele:

```
// ThemeSet öffnetThemeSet # tTheme->WinThemeSetOpen(1);if (tThemeSet > 0){ // Beschreibung set
```

ThemeSetId

Nummer des ThemeSets

Typ int

Liste, Objekte,

StyleTheme,

Siehe ThemeName,

WinPropGet(),

WinPropSet()

Diese Eigenschaft enthält die Nummer des ThemeSets, das aktiviert werden soll.

Das zu verwendende Theme wird in Abhängigkeit der Eigenschaften ThemeName und StyleTheme ermittelt:

- StyleTheme = WinStyleThemeSystem Die

Eigenschaft ThemeSetId wird ignoriert.

- StyleTheme = WinStyleThemeParent

Es wird das Theme des Elternobjektes verwendet. Das verwendete ThemeSet hängt von der Eigenschaft ThemeSetId des Objektes ab.

> 0

Es wird das ThemeSet verwendet, dass durch die Eigenschaft ThemeSetId definiert ist.

WinThemeSetParent Es wird das ThemeSet des Elternobjektes verwendet, dass durch die Eigenschaft ThemeSetId des Elternobjektes definiert ist.

WinThemeSetNone

Es wird das Theme ohne ThemeSet des Elternobjektes verwendet.

- StyleTheme = WinStyleThemeModern

Besitzt das Objekt die Eigenschaft ThemeName, definiert diese das für die Darstellung zu verwendende Theme. Der Wert der Eigenschaft ThemeSetId definiert das gewünschte ThemeSet.

Besitzt das Objekt keine ThemeName-Eigenschaft, wird das Theme über die Eigenschaft ThemeName des App-Objektes definiert. Der Wert der Eigenschaft ThemeSetId referenziert das ThemeSet aus diesem Theme. Enthält die Eigenschaft ThemeSetId den Wert WinThemeSetParent oder WinThemeSetNone, wird kein ThemeSet, sondern das Theme selber verwendet.



Existiert kein ThemeSet mit der angegebenen Nummer in dem eingesetzten Theme, werden die Eigenschaften des Themes verwendet. Der Wert der Eigenschaft ThemeSetId behält den gesetzten Wert.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

TickAlignment Ausrichtung der Markierungen des
Objekts Typ int

Siehe Liste, Objekte

In dieser Eigenschaft wird die Ausrichtung der Markierungen festgelegt.

Folgende Werte können angegeben werden:

WinTickAlignNone

Keine Markierungen anzeigen.

WinTickAlignRightBottom

Die Markierungen werden in Abhängigkeit von der
Eigenschaft Vertical rechts bzw. unterhalb des Kanals
angezeigt.

WinTickAlignLeftTop

Die Markierungen werden in Abhängigkeit von der
Eigenschaft Vertical links bzw. oberhalb des Kanals
angezeigt.

WinTickAlignBoth

Die Markierungen werden auf beiden Seiten des Kanals
angezeigt.

Kontakt

TickInterval



Abstand zwischen zwei Markierungen

Typ [int](#)

Siehe [Liste](#), [Objekte](#)

In dieser Eigenschaft wird der Werteanstand zwischen zwei Markierungen festgelegt. Wird dieser ein Wert von 2 zugewiesen, erhöht sich [CurrentInt](#) entsprechend um 2, wenn der Schieberegler eine Position nach rechts bzw. unten verschoben wird

Kontakt

TickLineSize 

Anzahl der Positionen, um die der Schieberegler mit den Pfeiltasten verschoben wird

Typ **int**

Siehe [Liste](#), [Objekte](#)

In dieser Eigenschaft wird die Anzahl der Positionen, um die der Schieberegler mit den Pfeiltasten ( /  /  / 

Kontakt



TickPageSize

Anzahl der Positionen, um die der Schieberegler mit den Tasten verschoben wird

/

Typ int

Siehe [Liste](#), [Objekte](#)

In dieser Eigenschaft wird die Anzahl der Positionen, um die der Schieberegler mit den Tasten / verschoben wird, festgelegt.

Kontakt

TifFileName

Dateiname des TIF-Dokuments

Typ alpha(8192)

Siehe Liste, Objekte,

Siehe PrtJobClose(), PrtPropSet()

Über diese Eigenschaft wird bei der Erzeugung eines TIFF-Dokuments der Name der zu erzeugenden Datei definiert. Die Eigenschaft muss angegeben werden, wenn von CONZEPT 16 aus eine TIFF-Datei erzeugt werden soll. Besteht der Druckjob aus mehreren Seiten, wird ein Multipage-TIFF erzeugt.

Über die Eigenschaft TifMode kann bestimmt werden, ob ein monochromes TIFF erzeugt wird.

Beispiel

```
tHdlPrintJob->ppTiffFileName # _Sys->spPathMyDocuments + '\Rechnung.tif';tErr # tHdlPrintJob->PrtJ
```

Kontakt

TifMode



Farb-Modus bei Drucken als TIFF-Datei

Typ int
Siehe Liste, Objekte, PrtJobClose(),
 PrtPropSet()

Über diese Eigenschaft wird bei der Ausgabe des Druckjobs als TIFF-Datei bestimmt, ob eine monochrome oder farbige Darstellung verwendet werden soll. Die Eigenschaft kann folgende Werte annehmen:

TifModeColor farbige Darstellung

TifModeMonochrome monochrome Darstellung

Wird die Eigenschaft nicht gesetzt, erfolgt ein farbiger Ausdruck.

Beispiel

```
tHdlPrintJob->ppTifFileName # _Sys->spPathMyDocuments + '\letterBW.tif';tHdlPrintJob->ppTifMode #
```

TileBorder

Rahmen des GroupSplit-Objekts

Typ int

Siehe Liste, Objekte, ColBoundary

Die Eigenschaft beim GroupSplit-Objekt gibt die Größe eines Rahmens an, der um jedes vorhandene GroupTile-Objekt gezeichnet wird.

Die Farbe des Rahmens ist individuell für jedes GroupTile-Objekt über die Eigenschaft ColBoundary definierbar. Hat ColBoundary den Wert WinColParent wird die Hintergrundfarbe des übergeordneten Objekts ermittelt. Hat ColBoundary den Wert WinColUndefined wird die Hintergrundfarbe des GroupTile-Objekts (Eigenschaft ColBkg) verwendet.

Die Breite des Rahmens wird in Pixel angegeben. Der Rahmen kann maximal acht Pixel breit werden.

Beispiel

```
$GroupSplit->wpTileBorder # 4;
```

Kontakt

TileTheme



Darstellung der Bilder von ImageTile, HdrImageTile und NodeStyle Typ int

Siehe Liste, Objekte, WinPropGet(), WinPropSet(), TileSize

Über diese Eigenschaft wird die Darstellung der Bilder von ImageTile, HdrImageTile und NodeStyle gesteuert.

Es gibt folgende Einstellungen:

WinTileThemeClassic Die Bilder werden in der klassischen Variante dargestellt.

WinTileThemeEnhanced Die Bilder werden in der erweiterten Variante dargestellt.

WinTileThemeModern Die Bilder werden in einer an Windows 8/8.1/10

angelehnten (flachen) Darstellung gezeichnet.

WinTileThemeFlatColor Die Bilder werden in einer an Office 2016 und Windows

8/8.1/10 angelehnten (flachen) Darstellung gezeichnet.

WinTileThemeFlatMono Die Bilder werden in einer an Office 2016 und Windows 8/8.1/10
angelehnten flachen und einfarbigen Darstellung gezeichnet.



Ist die Eigenschaft ColTheme des App-Objektes auf WinColorTileLow oder
WinColorTileC16Low gesetzt, wird die Eigenschaft TileTheme nicht ausgewertet.

Die Größe wird über die Eigenschaft TileSize festgelegt. Die Größe wirkt sich nicht in der klassischen Variante aus.

Kontakt

TimeStart

Startzeit der Drucker-Verfügbarkeit

Typ time

Siehe Liste, Objekte, PrtPropGet()

Die Eigenschaft liefert die Uhrzeit, ab wann der Drucker verfügbar ist.

Die Eigenschaft kann über den Befehl PrtPropGet() gelesen werden.

Kontakt

TimeUntil

Endzeit der Drucker-Verfügbarkeit

Typ time

Siehe Liste, Objekte, PrtPropGet()

Die Eigenschaft liefert die Uhrzeit bis wann der Drucker verfügbar ist.

Die Eigenschaft kann über den Befehl PrtPropGet() gelesen werden.

Kontakt

TitleJustify



Ausrichtung der Beschriftung des Achsenstittels

Typ int

Siehe

Liste, Objekte, WinPropGet(), WinPropSet(),
PrtPropGet(), PrtPropSet()

Mit dieser Eigenschaft wird die Ausrichtung der Beschriftung für den Achsentitel festgelegt.

Folgende Ausrichtungen sind möglich:

WinTextJustLeft linksbündige Ausrichtung

WinTextJustCenter zentrierte Ausrichtung

WinTextJustRight rechtsbündige Ausrichtung

Kontakt

TitleText

Titel des Objektes

Typ alpha(8192)

Liste, Objekte,
WinPropGet(),

Siehe WinPropSet(),
PrtPropGet(),
PrtPropSet()

Mit dieser Eigenschaft wird der Titel des Objektes festgelegt:

- Axis und PrtGanttAxis

Der in TitleText angegebene Text wird als Achsenbeschriftung verwendet.

- WinOpen(WinComPath)

Der Titeltext zwischen Titel und Treeview-Objekt. Je nach Einstellung und Betriebssystem kann der Text aus zwei oder drei Zeilen bestehen.

Toolbars

Anzeige von Toolbars

Typ int

Liste, Objekte,

Siehe WinPropGet(),
WinPropSet()

Über diese Eigenschaft kann gesteuert werden, welche Toolbars im CtxDocEdit-Objekt angezeigt werden sollen.

Es gibt folgende Ausprägungen:

WinToolbarRulerH Horizontales Lineal anzeigen

WinToolbarRulerV Vertikales Lineal anzeigen

WinToolbarButtons Toolbar mit Formatoptionen anzeigen

Die Ausprägungen können wie folgt überprüft und gesetzt werden:

```
// Ausprägung auswerten: if ($ctxDocEdit->wpToolbars & _WinToolbarButtons > 0){ ...}// Ausprägung
```



Die Toolbars können nur mit WinInfo() ermittelt werden, wenn diese in der Eigenschaft Toolbars eingeschaltet sind.

Kontakt

TrayFrame (Eigenschaft)

TrayFrame-Applikation zuweisen

Typ int
Siehe Liste, Objekte, WinPropGet(),

Diese Eigenschaft wird benötigt, wenn eine TrayFrame-Applikation zusammen mit einem Applikationsfenster laufen soll.

Der Deskriptor des TrayFrame-Objekts wird dieser Eigenschaft zugewiesen.

Beispiel:

```
// TrayFrame laden und zu Applikation zuweisen _App->wpTrayFrame # WinOpen('TrayStart', _WinOpenDi
```

Besteht die Applikation aus dem TrayFrame, muss die Eigenschaft nicht gesetzt werden.

Kontakt

TreeFlags



Darstellung des TreeView-Objektes

Typ int
Liste, Objekte, WinPropGet(),
Siehe WinPropSet()

Über diese Eigenschaft wird die Darstellung festgelegt.

Es gibt folgende Einstellungen:

WinTreeBoxes

Vor jedem Knoten wird ein Plus- bzw. Minus-Zeichen
dargestellt

WinTreeClickFocusSet

Fokus wird auf das TreeView-Objekt gesetzt

WinTreeImageExpand

Darstellung eines Bildes bei aufgeklappten Knoten

WinTreeImageGray

Alle Knoten-Symbole werden grau dargestellt

WinTreeImages

Vor den Knoten wird ein Bildchen angezeigt

WinTreeLines

Die Knoten sind mit Linien verbunden

WinTreeLinesAtRoot

In der ersten Ebene werden Linien zu den Wurzelknoten
gezeichnet

WinTreeNoHorzScroll

Horizontale Scrollbar wird ausgeblendet

WinTreeNoSelectOnFold Ereignis EvtNodeSelect nur beim Selektieren eines

Knotens auslösen

WinTreeShowSelected

Selektierter Knoten ist markiert, auch wenn der Baum den

Fokus nicht hat

Die Flags können wie folgt überprüft und gesetzt werden:

```
// Flag auswerten if ($TreeView->wpTreeFlags & _WinTreeImageExpand > 0) { ... } // Flag setzen $TreeView
```

Kontakt

TypeBarcode



Typ des Barcodes

Typ int

Liste, Objekte,
Caption,

Siehe Erzeugen von

Barcodes (Blog)
QR-Codes (Blog)

In dieser Eigenschaft wird der Typ des Barcodes angegeben. Das Objekt unterstützt unterschiedliche Barcodes. Der Wert in der Eigenschaft Caption wird in dem hier angegebenen Barcode ausgegeben. Folgende Barcode stehen zur Verfügung:

- _WinBarcodeNone

Es ist kein Barcode vorgegeben. Das System bestimmt aufgrund der Daten in der Eigenschaft Caption, welcher Barcode verwendet wird. Dort kann entweder der Barcode durch ein Präfix angegeben sein, oder er ergibt sich aus der Länge und der integrierten Prüfziffer des darzustellenden Wertes.

- _WinBarcodeI25 und _WinBarcodeI25R2

Präfix 'I25' bzw. 'I25R2' Abgeleitet von dem Barcode Industrial 2 of 5 besteht jedes Zeichen aus dem Alphabet des Barcodes Interleaved 2 of 5 aus fünf Strichen, von denen zwei breiter sind. In diesem Barcode werden immer zwei Zeichen zusammen kodiert, in dem die Abstände der Striche des ersten Zeichens, den Breiten der Striche des zweiten Zeichens entsprechen.



Auf diese Weise entsteht ein kompakterer Code. Es können allerdings nur Informationen mit einer geraden Anzahl von Ziffern kodiert werden. Wird eine ungerade Zahl von Ziffern angegeben, wird der Barcode mit einer führenden Null ergänzt. Als Start- und Ende-Zeichen werden zwei schmale bzw. ein breiter und ein schmaler Strich verwendet. Der Code wird hauptsächlich im Transportwesen und in der Warenwirtschaft verwendet.

Das Verhältnis des dünnen Strichs zum dicken Strich (bzw. der dünnen Lücke zur dicken Lücke) ist in dem Code mit 1:3 angegeben. Soll eine Ratio von 1:2 verwendet werden, muss der Typ _WinBarcodeI25R2 angegeben werden.

- _WinBarcode39 und _WinBarcode39CS

Präfix: 'Code39N' bzw. 'Code39C' Der "Code 39" oder "Code 3 from 9" verfügt über einen wesentlich größeren Zeichenvorrat. Mit diesem Code können Großbuchstaben, Ziffern und die Sonderzeichen -, ., <Space>, \$, /, + und %

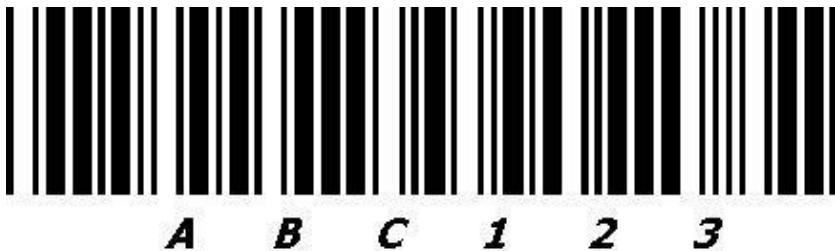
Kontakt

dargestellt werden. Dazu verfügt sein Alphabet noch über ein Start- und ein Ende-Zeichen. Jedes der Zeichen wird aus einer Gruppe aus fünf Balken und fünf Zwischenräumen dargestellt, wobei zwei der Balken und einer der Zwischenräume breiter sind. Ein Zeichen endet mit einem schmalen Zwischenraum. Damit besteht ein Zeichen aus 9 Strichen und Zwischenräumen, von denen 3 breiter sind (daher der Name).

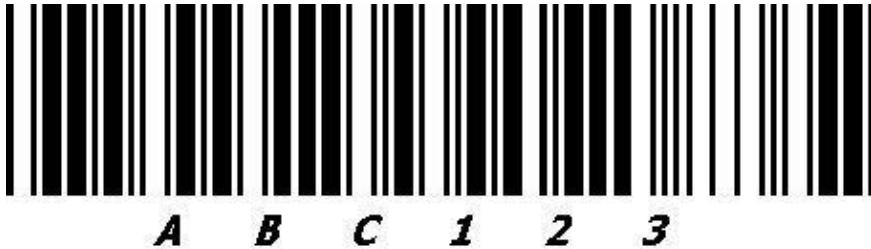
Das Zeichen * existiert nicht im Alphabet, kann aber zum Setzen des Startzeichens verwendet werden. In CONZEPT 16 wird das Start- und Ende-Zeichen automatisch gesetzt.

Der Code 39 kann optional mit einer Prüfsumme versehen werden (_WinBarcodeCode39CS). Die Prüfsumme ergibt sich aus der Summe der Zeichenwerte modulo 43 (Anzahl der Zeichen im Alphabet). Das Prüfsummenzeichen wird dem Code vor dem Ende-Zeichen angefügt.

Code ohne Prüfzeichen:



Code mit Prüfzeichen:



Das Start- und Ende-Zeichen sowie das eventuell vorhandene Prüfsummenzeichen müssen nicht mit angegeben werden.

Für den Code 39 existiert noch eine Variante, die alle Zeichen der ASCII-Tabelle bis Zeichen 127 (DEL) darstellt. Der "Extended Code 39" realisiert fast alle Zeichen, indem er für ein Zeichen zwei Barcode-Zeichen verwendet. Zum Beispiel besteht der Barcode für "Extended Code 39" der Zeichenfolge "Abc(123)" aus der Zeichenfolge "A+B+C/H123/I" in "Code 39".

- _WinBarcode128, _WinBarcode128b und _WinBarcode128c

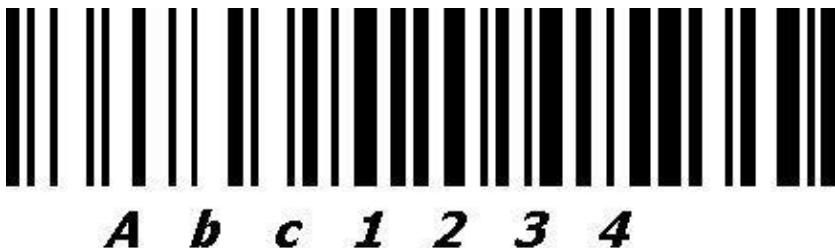
Präfix 'Code128X', 'Code128B', 'Code128C' Der "Code 128" beherrscht die ersten 128 Zeichen der ASCII-Tabelle bereits von Hause aus. Der Code besteht aus lediglich drei Strichen und drei Zwischenräumen. Es werden allerdings vier verschiedene Breiten unterschieden. Jedes Zeichen besteht aus 11 Einheiten,

Kontakt

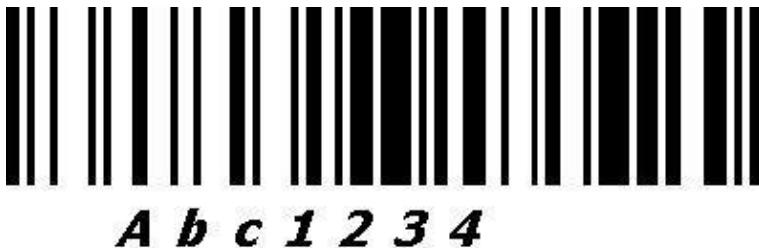
wobei dickere Stiche mehrere Einheiten einnehmen. Ein Zeichen beginnt mit einem Strich und endet mit einem Zwischenraum.

Der "Code 128" ist in drei Varianten (A, B und C) definiert. Die Variante A und B sind in den ersten 64 Zeichen identisch. In Variante A folgen danach die Steuerzeichen, während in Variante B die Kleinbuchstaben folgen. Die Variante C dient lediglich zur komprimierten Darstellung von Zahlenpaaren. Das Besondere an diesem Code ist, dass durch bestimmte Zeichen eine Umschaltung zwischen den Varianten erfolgen kann. Besteht also die Information aus Buchstaben und einer größeren (gerade) Anzahl von Ziffern, kann der Text im "Code 128 B" und die Ziffern in "Code 128 C" dargestellt werden.

Darstellung in Code 128 B:



Darstellung mit automatischer Umschaltung von Code 128 B und Code 128 C:



Neben den Zeichen und Sonderzeichen befinden sich noch vier sogenannte Funktionscodes im Alphabet. Diesen Funktionscodes sind keine Zeichen zugeordnet. Die entsprechende Applikation kann diese Codes für eigene Zwecke interpretieren.

Das ASCII-Zeichen 0 und die Funktionscodes werden wie folgt angegeben:

ASCII 0 -> StrCnv(StrChar(128), _StrFromAns)

Funktionscode 1 -> StrCnv(StrChar(193), _StrFromAns)

Funktionscode 2 -> StrCnv(StrChar(194), _StrFromAns)

Funktionscode 3 -> StrCnv(StrChar(195), _StrFromAns)

Funktionscode 4 -> StrCnv(StrChar(196), _StrFromAns)

Alle Varianten des Codes werden mit einem Prüfzeichen vor dem Ende-Zeichen versehen.

Dabei wird der Wert des Zeichens mit seiner Stelle in der Zeichenkette multipliziert und die Summe darüber gebildet. Das Prüfzeichen ist das Zeichen mit dem Wert der Summe modulo 103.

Kontakt

In dem Objekt PrtBarcode können die Varianten B bzw. C durch die Auswahl von _WinBarcodeCode128b bzw. _WinBarcodeCode128c erzeugt werden. Das automatische Wechseln zwischen den Codes ermöglicht die Angabe von _WinBarcodeCode128.

- _WinBarcodeGS1_128

Der Barcode GS1-128 (oder auch UCC-128, EAN-128) entspricht dem Barcode Code 128 C. Um die Barcodes zu unterscheiden wird beim GS1-128 das Funktionszeichen 1 (FNC1) als erstes Zeichen angegeben. Im Barcode sind einheitlich definierte Datenbezeichner festgelegt, die die Bedeutung der nachfolgenden Nummern ergeben. Ist das Funktionszeichen nicht als erstes Zeichen angegeben, wird dieses automatisch hinzugefügt.

- _WinBarcodeEAN

Präfix: 'EAN'. Der EAN-Code (European Article Number) ist in zwei Varianten definiert: EAN-13 (zwölf Nutzziffern, eine Prüfziffer) und EAN-8 (sieben Nutzziffern, eine Prüfziffer). Der am häufigsten zur Anwendung kommende Barcode ist der EAN-13. Er ist auf jedem Konsumartikel zu finden. Der Barcode wird nicht nur in Europa, sondern Weltweit verwendet. In Japan wird er als JAN (Japan Article Number) verwendet. In den USA und in Kanada hat er seinen Ursprung als UPC-A.

Mit dem EAN-Code können nur Ziffern dargestellt werden. Das Alphabet umfasst zusätzlich ein Start- und Ende-Zeichen und einen sogenannten "Center Guard". Der "Center Guard" teilt die Nummer nach sieben bzw. vier Ziffern in zwei Bereiche. Beim EAN-13 bestimmen die ersten zwei bis drei Ziffern das Herkunftsland oder die Anwendung des Codes. Im folgenden eine Auswahl der definierten Codes:

Code	Land/Anwendung
00..13	USA, Kanada
20..29	Lagerverwaltung
40..44	Deutschland
45, 49	Japan
76	Schweiz
746	Dominikanische Republik
977	International Standard Serial Number (ISSN)
978	International Standard Book Number (ISBN)
979	International Standard Music Number (ISMN)
980	Rückerstattungen
981, 982	Gemeinsame Währungscoupons
99	Coupons

Die folgenden vier oder fünf Ziffern (der Rest des ersten Blockes) enthält die Herstellerkennung. Die Nummern bzw. Nummernbereiche werden vom "Uniform Code Council" (UCC) bzw. von der "Centrale für Coorganisation" (CCG) vergeben und sind weltweit eindeutig. Die ersten fünf Ziffern des zweiten Blocks bilden den Produktcode. Dieser wird von der Herstellerfirma frei bestimmt. Die letzte Ziffer ist die Prüfsumme. Die Prüfsumme berechnet sich

Kontakt

aus der Summe der einzelnen Ziffern, wobei alle geraden Positionen zuvor mit drei multipliziert werden. Die Prüfsumme ist die Differenz von der entstandenen Summe zur nächsten durch zehn teilbaren Zahl.

EAN-13 Code:



Im PrtBarcode-Objekt kann der EAN-13-Code durch die Angabe von
_WinBarcodeEAN ausgewählt werden. Die Prüfziffer wird vom System berechnet und muss nicht mit angegeben werden.

- _WinBarcodeUPC

Der UPC-A-Barcode entspricht dem EAN-13 mit einer vorangestellten 0.

- _WinBarcodeISBN

Präfix: 'ISBN' Der ISBN-Code entspricht einem EAN-13-Code. Die ersten drei Ziffern sind mit 978 vorbelegt und müssen nicht mit angegeben werden. Die Prüfsumme wird durch CONZEPT 16 berechnet.

Neben diesen ein-dimensionalen Barcodes können auch zwei-dimensionale Barcodes angegeben werden. Bei diesen Barcodes hat die Eigenschaft ShowText keine Auswirkung.

- _WinBarcode2dCode16k

Der Barcode besteht aus maximal 16 Zeilen. Die Zeilen sind im Code 128 codiert.

- _WinBarcode2dCode49

Der Barcode Code 49 besteht aus zwei bis 8 Zeilen. Jede Zeile kann bis zu 16 Zeichenpaare beinhalten. Der Code ist mit einer Prüfsumme versehen, die nicht mit angegeben werden muss.

- _WinBarcode2dPDF417

In einem Barcode PDF417 können mehr als 1 Kilobyte an Daten angezeigt werden. Er wird in mehreren Zeilen aufgebaut und ist stabil gegenüber Lesefehlern.

- _WinBarcode2dMicroPDF417

Der Barcode Micro PDF417 ist vom Barcode PDF 417 abgeleitet. Er ist dichter, kann aber nicht so viele Informationen aufnehmen.

- _WinBarcode2dAztec

Kontakt

Der Barcode ist um einen "Center Guard" herum aufgebaut und kann in jede Richtung gelesen werden. Die Eigenschaft Vertical hat keine Auswirkung auf die Darstellung des Barcodes.

- _WinBarcode2dDataMatrix

Der Barcode wird ausgehend von einer waagerechten und einer senkrechten Linie aufgebaut. Er kann quadratisch oder rechteckig sein. Ist die AreaHeight des Objektes kleiner als zwei Drittel der AreaWidth, wird der Barcode rechteckig dargestellt.

- _WinBarcode2dGS1_DataMatrix

Der Barcode GS1 Datamatrix entspricht dem Barcode Datamatrix. Um die Barcodes zu unterscheiden wird beim GS1 Datamatrix der Funktionscode 1 (FNC1) als erstes Zeichen angegeben. Der Funktionscode wird bei Verwendung dieses Barcodes automatisch hinzugefügt. Im Barcode sind einheitlich definierte Datenbezeichner festgelegt, die die Bedeutung der nachfolgenden Nummern ergeben. Als Trennzeichen zwischen Zeichenketten variabler Länge kann das "FNC1 Codewort" oder der "Group Separator" verwendet werden. Diese werden in CONZEPT 16 wie folgt erzeugt:

FNC1 Codewort -> StrCnv(StrChar(129), _StrFromAnsi)

Group Separator -> StrCnv(StrChar(29), _StrFromAnsi)

Ist die AreaHeight des Objektes kleiner als zwei Drittel der AreaWidth, wird der Barcode rechteckig dargestellt.

- _WinBarcode2dMaxicode

Der MaxiCode besteht aus einem "Center Guard" um den sechseckige Waben angeordnet werden. Es können maximal 93 ASCII-Zeichen oder 138 Ziffern dargestellt werden. Der Code sollte immer in einer Größe von 25,4 x 25,4 mm dargestellt werden. Die Eigenschaft Vertical hat keine Auswirkung auf die Darstellung des Barcodes.

- _WinBarcode2dDataBar

Der Barcode GS1 Databar oder auch RSS (Reduced Space Symbology) besteht aus einer oder mehreren Zeilen, wobei einzelne Zeilen nicht durch Striche voneinander getrennt sind.

- _WinBarcode2dCodablockF

Der Barcode Codablock F basiert auf der Struktur des Code 128. Er kann zwischen 2 und 44 Zeilen mit je 4 bis 62 Zeichen beinhalten.

- _WinBarcode2dQRCode

Der Barcode ist immer quadratisch. In drei seiner Ecken befinden sich Positionierungsbereiche. Der Code ist zwischen 21 x 21 und 177 x 177 Elemente (ausgefüllte Quadrate) groß. Mit der Eigenschaft CorrectionLevel kann zwischen unterschiedlichen Korrekturstufen gewählt werden. Standardmäßig können die Daten auch dann noch wieder hergestellt werden, wenn bis zu 7 % und mit dem größten Level bis zu 30 % der Codewörter nicht gelesen werden konnten. Die Eigenschaft Vertical hat keine Auswirkung auf die Darstellung des Barcodes. Über die Eigenschaft Utf8 kann der Zeichensatz

Kontakt

zwischen UTF8 und ISO8859-1 gewechselt werden.

- WinBarcode2dMicroQRCode

Der Micro-QR-Code hat nur eine Größe von 11 x 11 bis 17 x 17 Elementen und kann daher nur weniger Daten beinhalten. Er ist aber von seinen Ausmaßen geringer. Die Eigenschaft Vertical hat keine Auswirkung auf die Darstellung des Barcodes.

Kontakt

TypeButton 

Aktion der Schaltfläche

Typ int
Liste, Objekte,
Siehe WinPropGet(),
WinPropSet()

In dieser Eigenschaft kann die Aktion bestimmt werden, die beim Drücken der Schaltfläche auszuführen ist.

Folgende Aktionen stehen zur Verfügung:

<u>WinBtnNone</u>	Keine Aktion soll stattfinden
<u>WinBtnClose</u>	Schließen des Fensters
<u>WinBtnHelp</u>	Aufruf der Hilfe
<u>WinBtnUserBreak</u>	Aktion durch Benutzer abgebrochen

Kontakt

TypeFile

Filmtyp

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Es können verschiedene Animationen des Betriebssystems oder eine Animation, die als AVI-Datei vorliegt, angezeigt werden.

Folgende Animationen des Betriebssystems können angezeigt werden:

WinAniExternFile

Der in FileName eingetragene externe Film wird verwendet

WinAniScanning

Suchende Taschenlampe

WinAniSearchComputer Lupe vor Computer

WinAniSearchSheet

Lupe vor Datei

WinAniFileMove

Seiten werden von einem Ordner in einen anderen

verschoben

eine Seite wird in einen anderen Ordner kopiert

verschieben einer Seite in den Papierkorb

eine Seite aus dem Papierkorb wird gelöscht

WinAniFileCopy

WinAniFileTrash

WinAniFileDelete

Kontakt

Unit

Lineal Maßeinheit

Typ int

Siehe Liste, Objekte

Die Eigenschaft definiert die Maßeinheit für die Lineal-Anzeige im Formular-Objekt.

PrtUnitCentimetres Anzeige erfolgt in Zentimeter

PrtUnitInches Anzeige erfolgt in Zoll

Kontakt

Url



Aktuell angezeigte Adresse des WebNavigators

Typ alpha

Siehe Liste, Objekte

- WebNavigator-Objekt

In dieser Eigenschaft steht die aktuell angezeigte Adresse des WebNavigator-Objektes.

Diese Eigenschaft verändert sich, wenn sich die Caption des WebNavigator-Objektes verändert, oder eine andere Seite, beispielsweise per Klick auf einen Link, aufgerufen wird. Die Eigenschaft Url enthält somit den aktuellen Pfad des angezeigten Inhalts. Weiterhin wird beim Ändern der Eigenschaft Url das EvtChanged ausgelöst.

Die Eigenschaft kann nur gelesen werden.

- Chromium-Objekt

Die Eigenschaft enthält die finale URL des angezeigten Inhaltes. Die

Eigenschaft kann nur gelesen werden.

Kontakt

Utf8

Barcode im UTF8-Zeichensatz erzeugen

Typ [logic](#)

Siehe [Liste, Objekte](#),

[WinBarcode2dQRCode](#)

In dieser Eigenschaft kann für QR-Codes ([TypeBarcode](#) = [WinBarcode2dQRCode](#) definiert werden, ob der Barcode im UTF8-Zeichensatz ([true](#)) oder im ISO8859-1 Zeichensatz erzeugt wird. Standardmäßig ist der Wert auf [true](#) gesetzt.

Kontakt

Version Version des Druckertreibers bzw. Druckjobs

Typ int

Siehe [Liste](#), [Objekte](#), [PrtPropGet\(\)](#)

Die Eigenschaft liefert die Version des Druckertreibers ([PrintDevice](#)) bzw. des Druckjobs ([PrintJob](#)).

Die Eigenschaft kann über den Befehl [PrtPropGet\(\)](#) gelesen werden.

Kontakt

Vertical



Vertikale Darstellung des Objekts

Typ logic
 Liste, Objekte,
 WinPropGet(),
Siehe WinPropSet(),
 PrtPropGet(),
 PrtPropSet()

Wird diese Eigenschaft auf true gesetzt, wird das Objekt um 90 Grad gegen den Uhrzeigersinn gekippt.

Das Setzen dieser Eigenschaft setzt ebenfalls die Eigenschaften ModeEffect und Rotation, falls diese bei dem Objekt vorhanden sind.

Beim GroupSplit-Objekt hat diese Eigenschaft eine andere Bedeutung. Dort korrespondiert sie mit der Eigenschaft StyleGroup und bestimmt die Anordnung der untergeordneten GroupTile-Objekte. Bei Angabe von true werden die GroupTile-Objekte übereinander, bei false nebeneinander dargestellt.

Kontakt

VerticalText Vertikale 

Darstellung Typ logic

Liste, Objekte,
WinPropGet(),

Siehe WinPropSet(),

PrtPropGet(),
PrtPropSet()

Ist der Wert der Eigenschaft true, wird die Beschriftung vertikal (um 90° gedreht) dargestellt, ansonsten von links nach rechts.

Kontakt

ViewId

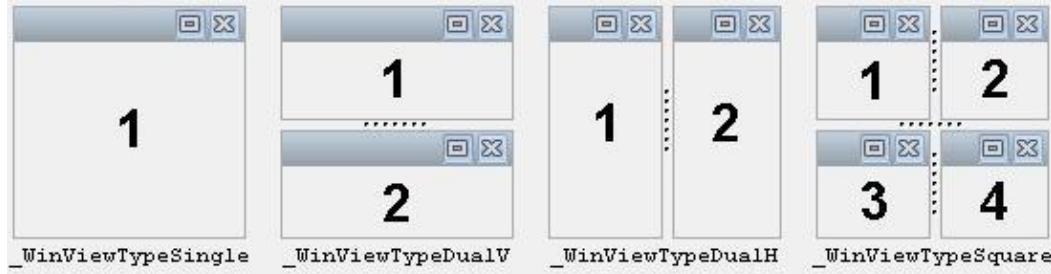


Nummer des Views zur Anzeige von Overlays Typ int

Siehe Liste, Objekte

Hier wird die Nummer des Views eingetragen (1 bis 4) in dem das Overlay angezeigt werden soll. Das Overlay ist nur sichtbar, wenn auch die Bedingung der Eigenschaft ViewType gegeben ist und das angegebene View angezeigt wird.

Nummern der Views:



Kontakt

ViewMode

Anzeigemodus

Typ int

[Liste, Objekte,](#)

Siehe [WinPropGet\(\)](#),

[WinPropSet\(\)](#)

Über diese Eigenschaft kann gesteuert werden, in welchem Anzeigemodus das Dokument im [CtxDocEdit](#)-Objekt angezeigt werden soll. Folgende Werte können in dieser Eigenschaft angegeben werden:

[WinViewModeNormal](#) Seitenränder, Rahmen und Lücken werden nicht angezeigt.

[WinViewModePage](#) Dreidimensionale Ansicht der Dokumentseite.

[WinViewModeFloating](#) Das Dokument wird auf die Breite des [CtxDocEdit](#)-Objektes umgebrochen.

[WinViewModeSimple](#) Das Dokument wird an das [CtxDocEdit](#)-Objekt angepasst. Es werden keine Scollbars angezeigt.

Kontakt

ViewType

Bedingungen zur Anzeige von Overlays

Typ int

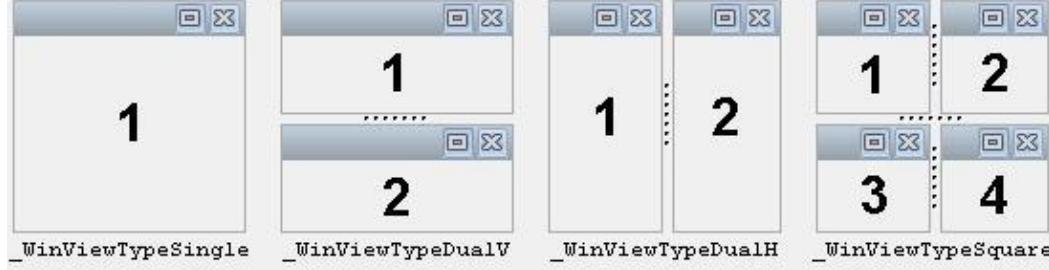
Siehe Liste, Objekte

In dieser Eigenschaft wird angegeben, unter welchen Bedingungen ein Overlay (siehe DocView) angezeigt wird. Folgende Optionen bzw. Kombinationen daraus, können angegeben werden:

- _WinViewTypeSingle (1) Es ist ein View vorhanden.
- _WinViewTypeDualH (2) Es sind zwei nebeneinanderliegende Views sichtbar.
- _WinViewTypeDualV (4) Es sind zwei übereinanderliegende Views sichtbar.
- _WinViewTypeSquare (8) Es sind alle vier Views sichtbar.

Das Overlay ist nur sichtbar, wenn ein View mit der in ViewId eingestellten Nummer angezeigt und das Objekt im angegebenen Modus ist.

ViewType und Nummer der Views:



Kontakt

VisibleOrder

Angezeigte Spaltenposition

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

Mit dieser Eigenschaft wird die angezeigte Position der Spalte festgelegt.

Die Eigenschaft kann im Editor oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Mit dem Befehl WinInfo(..., WinNext) werden die Spalten in der im RecView-Editor angezeigten Reihenfolge ermittelt.

WaitCursor

Mauszeiger als Sanduhr darstellen

Typ logic
Siehe Liste, Objekte, WinPropGet(),

WinPropSet(), Blog

Wird diese Eigenschaft auf true gesetzt, nimmt der Mauszeiger die Form einer Sanduhr an. Mit dem Mauszeiger können dann innerhalb der Applikation keine Aktionen ausgeführt werden. Die Eigenschaft muss durch das Programm wieder auf den Wert false gesetzt werden.

Kontakt

WidthPen



Strichstärke

Typ int

Liste,

Siehe Objekte,

PrtPropGet(),

PrtPropSet()

Die Eigenschaft definiert die Strichstärke zum Zeichnen des PrtDivider- oder PrtBorder-Objektes in logischen Einheiten.

Kontakt

WordBreak | | Automatischer Umbruch bei zu langem
Text Typ logic

Siehe Liste, Objekte, WinPropGet(),
WinPropSet(), PrtPropGet(), PrtPropSet()

Wenn diese Eigenschaft auf true gesetzt ist, erfolgt ein Zeilenumbruch sobald der Text länger ist als das Objekt.

Der Umbruch der Zeile wird hinter einem Leerzeichen und bei dem Objekt PrtText zusätzlich hinter einem Bindestrich vorgenommen.

Kontakt

XmlFileName

Dateiname des XML-Dokumentes

Typ alpha(8192)

Siehe Liste, Objekte, PrtJobClose()

Soll mit der Anweisung PrtJobClose() eine XML-Datei erzeugt werden, wird in dieser Eigenschaft der Dateiname angegeben.

Beispiel

```
tHdlPrintJob->ppXmlFileName # _Sys->spPathMyDocuments + '\Rechnung080412.xml';tErr # tHdlPrintJob
```

Kontakt

ZoomEffective

Vergrößerungsfaktor

Typ int
Siehe Liste, Objekte,

ZoomFactor

Mit dieser Eigenschaft kann der Vergrößerungsfaktor ermittelt werden, auch wenn das dargestellte Dokument auf die Fensterbreite oder zur Darstellung der ganzen Seite skaliert wurde. Ist dem Objekt kein Dokument zugeordnet, hat die Eigenschaft den Wert 0.

Kontakt

ZoomFactor



Benutzerdefinierter Vergrößerungsfaktor

Typ int

Liste, Objekte, WinPropGet(),

Siehe WinPropSet(), PrtPropGet(),
PrtPropSet()

- Picture-Objekt

Mit dieser Eigenschaft kann eine Grafik prozentual vergrößert oder verkleinert werden.

Bei einem Faktor von 100 wird das Bild in seiner Originalgröße dargestellt (100)

Wird der Zoomfaktor mit einem negativen Vorzeichen versehen, wird das Bild zusätzlich im Objekt zentriert dargestellt. Der zulässige Wertebereich liegt somit bei -500 bis 500.



Ist die Eigenschaft ModeZoom gesetzt, hat ein Eintrag in dieser Eigenschaft keine Wirkung.

Beispiele:

100 Bild in Orginalgröße

50 Bild in halber Größe

0 Objektfüllende Darstellung

-50 Bild in halber Größe zentriert zum Objekt -100

Bild in Orginalgröße zentriert zum Objekt

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

- PrintForm-/PrintDoc-Objekt

Mit dieser Eigenschaft wird der Inhalt des Formulars auf den gegebenen Wert gezoomt (vergrößert oder verkleinert). Es sind Zoomfaktoren im Bereich von 25 - 500 Prozent zulässig. Wurde ein Vergrößerungsfaktor relativ zum dargestellten Objekt gewählt (Anzeige Fensterbreite oder Ganze Seite), kann der Vergrößerungsfaktor über die Eigenschaft ZoomEffective des PrtJobPreview-Objektes ermittelt werden.

In diesem Fall können die Befehle PrtPropGet() und PrtPropSet() zum Lesen bzw. Setzen der Eigenschaft verwendet werden.

- RtfEdit-Objekt

Mit dieser Eigenschaft kann der Inhalt des RtfEdit-Objektes prozentual vergrößert oder verkleinert werden.



Damit sich diese Eigenschaft auswirkt, muss in der Eigenschaft PrtDevice der Deskriptor eines PrintDevice angegeben werden.

- Chromium-Objekt

Kontakt

Gibt den Zoom-Faktor des angezeigten Inhaltes an. Gültige Werte liegen im Bereich 25 - 500. Der Versuch einen ungültigen Wert zu setzen, führt zum Laufzeitfehler ErrValueInvalid.

Der Zoom-Faktor ist objektübergreifend, wirkt sich also auf andere Instanzen des Chromium-Objektes aus, wenn diese dieselbe (Sub-)Domain anzeigen.

ZOrder

Darstellungsreihenfolge

Typ int

Siehe Liste, Objekte

Die Eigenschaft ZOrder wird bei der Darstellung von Intervallen in einem GanttGraph bzw. PrtGanttGraph und bei der Bestimmung der Druckreihenfolge in einem Druck-Job verwendet.

- Verwendung in einem Gantt-Graphen

Die Eigenschaft bestimmt in welcher Reihenfolge die Intervall-Objekte gezeichnet werden.

Die Eigenschaft wird nur ausgewertet, wenn bei den Objekten GanttGraph bzw.

PrtGanttGraph die Option WinGanttZOrderDraw bzw. PrtGanttZOrderDraw in der Eigenschaft GanttFlags gesetzt ist.

In der Eigenschaft können Werte im Bereich 0 - 2.147.483.647 angegeben werden. Ein Objekt wird dann über ein anderes Objekt gezeichnet, wenn es einen höheren Wert in der Eigenschaft ZOrder hat. Sollten zwei Intervalle den gleichen Wert besitzen, kommt das Intervall in den Vordergrund, das den kleineren horizontalen Endpunkt besitzt.

- Reihenfolge der Druckausgabe

Die Eigenschaft bestimmt in welcher Reihenfolge die Druck-Objekte dem Druckjob hinzugefügt werden. Die Eigenschaft wird nur ausgewertet, wenn bei der Anweisung PrtJobOpen() die Konstante PrtJobOpenSort angegeben ist.

In der Eigenschaft können Werte im Bereich 0 - 2.147.483.647 angegeben werden. Ein Objekt wird nach einem anderen Objekt gedruckt, wenn es einen höheren Wert in der Eigenschaft ZOrder hat. Sollten beide Objekte den gleichen Wert besitzen, kommt das Objekt in den Vordergrund, das weiter unten beginnt.

Function (Funktions-Eigenschaften)

Eigenschaften

sortiert nach

Gruppen,

Siehe

Alphabetische

Liste aller

FnCount Anzahl der Funktionen

FnIndex1 Auswahl der 1. Funktionsnummer

FnIndex2 Auswahl der 2. Funktionsnummer

FnIndex3 Auswahl der 3. Funktionsnummer

FnMode1 1. Funktionsmodus FnMode2 2.

Funktionsmodus

FnMode3 3. Funktionsmodus

FnResult Index des Funktionsergebnisses

Kontakt

FnCount

Anzahl der Funktionen

Typ int

Liste, Objekte,

Siehe PrtPropGet(),

PrtPropSet()

Mit dieser Eigenschaft wird eingestellt, wie viele Funktionen bei einem PrintDocRecord bei der Aufbereitung von einem PrintJob angesprochen werden können.

Die Funktionsnummern können erst nachdem die Anzahl der Funktionen angegeben ist, in die Eigenschaften FnIndex1, FnIndex2 und FnIndex3 eingetragen werden.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle PrtPropGet() und PrtPropSet() gelesen bzw. geändert werden.

Kontakt

FnIndex1



Auswahl der 1. Funktionsnummer

Typ int
Siehe Liste, Objekte, PrtPropGet(),
PrtPropSet()

Die Eigenschaft definiert den ersten Funktionsindex.

Die Eigenschaft kann erst dann gesetzt werden, wenn in der Eigenschaft DbFieldName ein Datenbankfeld eingetragen wurde. Die Eigenschaft FnCount des Objektes PrintDoc oder PrintDocRecord bestimmt die Anzahl der zur Verfügung stehenden Funktionen und muss ebenfalls gesetzt werden. Neben dem Funktionsindex muss noch der Funktionsmodus in der Eigenschaft FnMode1 gesetzt werden.

Das Ergebnis einer Funktion kann über ein PrtText-Objekt ausgegeben werden, wenn die Eigenschaft StyleCaption auf PrtStyleCapFnResult gesetzt ist.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle PrtPropGet() und PrtPropSet() gelesen bzw. geändert werden.

Kontakt

FnIndex2



Auswahl der 2. Funktionsnummer

Typ int
Siehe Liste, Objekte, PrtPropGet(),
PrtPropSet()

Die Eigenschaft definiert den zweiten Funktionsindex.

Die Eigenschaft kann erst dann gesetzt werden, wenn in der Eigenschaft DbFieldName ein Datenbankfeld eingetragen wurde. Die Eigenschaft FnCount des Objektes PrintDoc oder PrintDocRecord bestimmt die Anzahl der zur Verfügung stehenden Funktionen und muss ebenfalls gesetzt werden. Neben dem Funktionsindex muss noch der Funktionsmodus in der Eigenschaft FnMode2 gesetzt werden.

Das Ergebnis einer Funktion kann über ein PrtText-Objekt ausgegeben werden, wenn die Eigenschaft StyleCaption auf PrtStyleCapFnResult gesetzt ist.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle PrtPropGet() und PrtPropSet() gelesen bzw. geändert werden.

Kontakt

FnIndex3



Auswahl der 3. Funktionsnummer

Typ int
Siehe Liste, Objekte, PrtPropGet(),
PrtPropSet()

Die Eigenschaft definiert den dritten Funktionsindex.

Die Eigenschaft kann erst dann gesetzt werden, wenn in der Eigenschaft DbFieldName ein Datenbankfeld eingetragen wurde. Die Eigenschaft FnCount des Objektes PrintDoc oder PrintDocRecord bestimmt die Anzahl der zur Verfügung stehenden Funktionen und muss ebenfalls gesetzt werden. Neben dem Funktionsindex muss noch der Funktionsmodus in der Eigenschaft FnMode3 gesetzt werden.

Das Ergebnis einer Funktion kann über ein PrtText-Objekt ausgegeben werden, wenn die Eigenschaft StyleCaption auf PrtStyleCapFnResult gesetzt ist.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle PrtPropGet() und PrtPropSet() gelesen bzw. geändert werden.

Kontakt

FnMode1



1. Funktionsmodus

Typ **int**

[Liste, Objekte](#),

Siehe [PrtPropGet\(\)](#),

[PrtPropSet\(\)](#)

Die Eigenschaft definiert den ersten Funktionsmodus.

Die Eigenschaft kann erst dann gesetzt werden, wenn in der Eigenschaft [DbFieldName](#) ein Datenbankfeld eingetragen wurde.

Kontakt

FnMode2



2. Funktionsmodus

Typ **int**

[Liste, Objekte](#),

Siehe [PrtPropGet\(\)](#),

[PrtPropSet\(\)](#)

Die Eigenschaft definiert den zweiten Funktionsmodus.

Die Eigenschaft kann erst dann gesetzt werden, wenn in der Eigenschaft DbFieldName ein Datenbankfeld eingetragen wurde.

Kontakt

FnMode3



3. Funktionsmodus

Typ int

Liste, Objekte,

Siehe PrtPropGet(),

PrtPropSet()

Die Eigenschaft definiert den dritten Funktionsmodus.

Die Eigenschaft kann erst dann gesetzt werden, wenn in der Eigenschaft DbFieldName ein Datenbankfeld eingetragen wurde.

FnResult

Nummer des Funktionsergebnisses

Typ int
Siehe Liste, Objekte, PrtPropGet(),

Die Nummer im Objekt ist der Funktionsindex der Funktion, deren Ergebnis im Objekt dargestellt werden soll. Die Eigenschaft StyleCaption muss den Wert PrtStyleCapFnResult enthalten, damit das Ergebnis dargestellt wird.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle PrtPropGet() und PrtPropSet() gelesen bzw. geändert werden.

Theming (Theme-Eigenschaften)

Eigenschaften

sortiert nach

Gruppen,

Siehe

Alphabetische

Liste aller

StyleTheme Darstellung von Oberflächen-Objekten

ThemeName Name des zu verwendenden Themes

ThemeSetId Nummer des ThemeSets

Kontakt

StyleTheme



Darstellung von Oberflächen-Objekten

Typ int

Liste, Objekte, ThemeName,

Siehe ThemeSetId,

Betriebssystemdarstellung (Blog)

Über diese Eigenschaft des Application-Objekts kann die betriebssystemabhängige Darstellung von Oberflächenobjekten konfiguriert werden. Dazu können dieser Eigenschaft die folgenden Werte zugeordnet werden:

WinStyleThemeSystem

Objekte werden in der betriebssystemabhängigen Theme-Darstellung angezeigt (Vorgabe für das _App-Objekt und Calendar-Objekte).

WinStyleThemeNone

Objekte werden nicht betriebssystemabhängig angezeigt.

WinStyleThemeModern

Das Objekt und seine Unterobjekte werden im Modern Theme Style dargestellt. Das zu verwendende Theme wird über die Eigenschaft ThemeName des Objektes definiert. Ist diese Eigenschaft nicht vorhanden oder leer, wird der beim _App-Objekt eingetragene Theme-Name verwendet.

WinStyleThemeParent

Es wird in der Elternhierarchie nach dem ersten übergeordneten Objekt gesucht, bei dem die Eigenschaft StyleTheme nicht auf _WinStyleThemeParent steht und entsprechend Modern Theme Style aktiviert (bei WinStyleThemeModern) bzw. deaktiviert (bei WinStyleThemeSystem). Hat auch die Eigenschaft StyleTheme des obersten Frame-Objektes den Wert WinStyleThemeParent, dann wird die systemabhängige Darstellung aktiviert. Dieser Wert ist Vorgabe für alle Objekte, die den Modern Theme Style unterstützen.



Die Eigenschaft kann beim _App-Objekt nicht während der laufenden Applikation geändert werden. Sie muss im Ereignis EvtInit des Frames gesetzt werden.



Die Eigenschaft StyleTheme kann beim _App- und beim Calendar-Objekt nicht auf WinStyleThemeModern bzw. WinStyleThemeParent gesetzt werden.



Die Ausprägung WinStyleThemeNone kann nur bei den Objekten App, Calendar, DateEdit, WinComFileOpen und WinComFileSave gesetzt werden. Beim DateEdit wirkt sich diese Ausprägung nur auf das enthaltene Calendar-Pop-up aus.

Bei den Konstanten WinStyleThemeModern und WinStyleThemeParent kann mit der Eigenschaft ThemeSetId zusätzlich noch ein ThemeSet des Themes ausgewählt werden. Dieses erbt die Eigenschaften des Themes und ermöglicht somit kleinere Anpassungen am Theme für einzelne Unterobjekte.

Über den Befehl WinInfo() mit der Option WinThemeActive kann ermittelt werden, ob die betriebssystemabhängige Darstellung eingestellt ist.

Die Theme-Darstellung wirkt sich auf folgende Eigenschaften und Objekte aus:

- ColFg

Kontakt

Das Setzen der Vordergrundfarbe hat bei den Objekten CheckBox, RadioButton, Notebook, Button und GroupBox keine Auswirkung.

- ColBkg

Das Setzen der Hintergrundfarbe bei dem Objekt Button bewirkt, dass nicht mehr die Schaltfläche, sondern nur noch ein Rand eingefärbt wird.

- Darstellung des HelpTip

Ist die Eigenschaft HelpTipSysFont = true erfolgt eine andere Darstellung des Helptipps. Im angezeigten Objekt befindet sich ein Farbverlauf im Hintergrund und die Ecken sind abgerundet.

Ausnahmen

Die betriebssystemabhängige Darstellung wirkt sich in der Regel auf alle Oberflächenobjekte aus. Ausnahmen bilden die folgenden Objekte. Dort wird die entsprechende Betriebssystemdarstellung separat definiert.

- Notebook - Hier wird der Stil über die Eigenschaft StyleTab mit dem Wert WinStyleTabTheme festgelegt.
- Progress - Hier wird der Stil über die Eigenschaft StyleShade mit dem Wert WinStyleShadeTheme festgelegt.
- Frame - Hier wird der betriebssystemabhängige Stil über das logische Setzen der Eigenschaft ThemeBkg aktiviert. Dies trifft ebenfalls auf die Objekte AppFrame und MdiFrame zu.
- DateEdit und Calendar - Hier ist die Eigenschaft StyleTheme ebenfalls vorhanden. Das Objekt wird nur in der betriebssystemabhängigen Darstellung dargestellt, wenn sowohl bei dem App-Objekt, als auch bei dem DateEdit- bzw. Calendar-Objekt die Eigenschaft auf WinStyleThemeSystem gesetzt ist.
- WinComFileOpen und WinComFileSave - Hier ist die Eigenschaft StyleTheme zusätzlich vorhanden. Ihr Wert wird beim Öffnen des Systemdialogs (WinOpen()) vom App-Objekt kopiert.

ThemeName

Name des Themes

Typ alpha(60)

Liste, Objekte,

StyleTheme,

Siehe ThemeSetId,

WinPropGet(),

WinPropSet()

Diese Eigenschaft enthält den Name des Themes, das aktiviert werden soll.

Ist die Eigenschaft leer, wird der im App-Objekt eingetragene Theme-Name herangezogen. Ist auch dieser leer, wird das vordefinierte Theme _OfficeBlue geladen.

Ist die Eigenschaft gesetzt, enthält sie den Name eines in der Datenbank abgelegten Themes. Die Eigenschaft ist bei allen Container-Objekten definiert, die Modern Theme Style unterstützen. Somit kann für das Container-Objekt und seine Unterobjekte ein abweichendes Theme aktiviert werden. Falls ein Theme von einem anderen Theme abgeleitet werden soll, können zudem auch ThemeSets definiert werden. Diese erben alle nicht gesetzten Eigenschaften vom definierten Theme.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

ThemeSetId

Nummer des ThemeSets

Typ int

Liste, Objekte,

StyleTheme,

Siehe ThemeName,

WinPropGet(),

WinPropSet()

Diese Eigenschaft enthält die Nummer des ThemeSets, das aktiviert werden soll.

Das zu verwendende Theme wird in Abhängigkeit der Eigenschaften ThemeName und StyleTheme ermittelt:

- StyleTheme = WinStyleThemeSystem Die

Eigenschaft ThemeSetId wird ignoriert.

- StyleTheme = WinStyleThemeParent

Es wird das Theme des Elternobjektes verwendet. Das verwendete ThemeSet hängt von der Eigenschaft ThemeSetId des Objektes ab.

> 0

Es wird das ThemeSet verwendet, dass durch die Eigenschaft ThemeSetId definiert ist.

WinThemeSetParent Es wird das ThemeSet des Elternobjektes verwendet, dass durch die Eigenschaft ThemeSetId des Elternobjektes definiert ist.

WinThemeSetNone

Es wird das Theme ohne ThemeSet des Elternobjektes verwendet.

- StyleTheme = WinStyleThemeModern

Besitzt das Objekt die Eigenschaft ThemeName, definiert diese das für die Darstellung zu verwendende Theme. Der Wert der Eigenschaft ThemeSetId definiert das gewünschte ThemeSet.

Besitzt das Objekt keine ThemeName-Eigenschaft, wird das Theme über die Eigenschaft ThemeName des App-Objektes definiert. Der Wert der Eigenschaft ThemeSetId referenziert das ThemeSet aus diesem Theme. Enthält die Eigenschaft ThemeSetId den Wert WinThemeSetParent oder WinThemeSetNone, wird kein ThemeSet, sondern das Theme selber verwendet.



Existiert kein ThemeSet mit der angegebenen Nummer in dem eingesetzten Theme, werden die Eigenschaften des Themes verwendet. Der Wert der Eigenschaft ThemeSetId behält den gesetzten Wert.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle WinPropGet() und WinPropSet() gelesen bzw. geändert werden.

Kontakt

Fill Color (Farb-Eigenschaften für Füllfarbe von Themes)

Eigenschaften

sortiert nach

Gruppen,

Siehe

Alphabetische

Liste aller

- [ThemeColFillActive](#)
- [ThemeColFillBarDisabled](#)
- [ThemeColFillBarNormal](#)
- [ThemeColFillBarShadeDisabled](#)
- [ThemeColFillBarShadeNormal](#)
- [ThemeColFillBase](#)
- [ThemeColFillCaption](#)
- [ThemeColFillCaptionActive](#)
- [ThemeColFillCaptionInactive](#)
- [ThemeColFillChannelDisabled](#)
- [ThemeColFillChannelHighlight](#)
- [ThemeColFillChannelNormal](#)
- [ThemeColFillChannelPressed](#)
- [ThemeColFillCheckDisabled](#)
- [ThemeColFillChecked](#)
- [ThemeColFillCheckHighlight](#)
- [ThemeColFillCheckNormal](#)
- [ThemeColFillDisabled](#)
- [ThemeColFillEnd](#)
- [ThemeColFillError](#)
- [ThemeColFillFocussed](#)
- [ThemeColFillHdrBase](#)
- [ThemeColFillHdrDisabled](#)
- [ThemeColFillHdrHighlight](#)
- [ThemeColFillHdrNormal](#)
- [ThemeColFillHdrPressed](#)
- [ThemeColFillHighlight](#)
- [ThemeColFillHover](#)
- [ThemeColFillInactive](#)
- [ThemeColFillIndentNormal](#)
- [ThemeColFillNormal](#)
- [ThemeColFillPressed](#)
- [ThemeColFillReadOnly](#)
- [ThemeColFillRulerNormal](#)
- [ThemeColFillSelectedActive](#)
- [ThemeColFillSelectedInactive](#)
- [ThemeColFillShadeChecked](#)
- [ThemeColFillShadeDisabled](#)
- [ThemeColFillShadeHighlight](#)
- [ThemeColFillShadeNormal](#)
- [ThemeColFillShadePressed](#)
- [ThemeColFillTabNormal](#)
- [ThemeColFillThumbDisabled](#)

Kontakt

- ThemeColFillThumbHighlight
- ThemeColFillThumbNormal
- ThemeColFillThumbPressed
- ThemeColFillVisited
- ThemeOptFillShade

Kontakt

ThemeColFillActive Füllfarbe

bei Status "aktiv" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe, wenn das Oberflächenobjekt den Eingabefokus besitzt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEditListColumn
- WinTypeColumnCheckbox
- WinTypeColumnRadiobutton
- WinTypeDataListColumn
- WinTypeDataListPopupColumn
- WinTypeRecListColumn
- WinTypeRecListPopupColumn
- WinTypeStoListColumn
- WinTypeStoListPopupColumn
- WinTypeTreeNode

Beispiele:

```
tTheme->wpThemeColFillActive(_WinTypeDataListColumn) # _WinColRed;tColor # tTheme->wpThemeColFill
```

Kontakt

ThemeColFillBarDisabled



Füllfarbe der Bar bei Status Disabled

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe der Bar, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeProgress

Beispiele:

```
tTheme->wpThemeColFillBarDisabled(_WinTypeProgress) # _WinColRed;tColor # tTheme->wpThemeColFillB
```

Kontakt

ThemeColFillBarNormal



Füllfarbe der Bar bei Status "normal"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe der Bar, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt nicht gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeProgress

Beispiele:

```
tTheme->wpThemeColFillBarNormal(_WinTypeProgress) # _WinColRed;tColor # tTheme->wpThemeColFillBar
```

Kontakt

ThemeColFillBarShadeDisabled



Zielfüllfarbe des Farbverlaufs der Bar bei Status Disabled

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Zielfüllfarbe des Farbverlaufs der Bar, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.



Zusätzlich muss die Eigenschaft ThemeOptFillShade auf einen Wert ungleich WinShadeSolid gesetzt sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeProgress

Beispiele:

```
tTheme->wpThemeColFillBarShadeDisabled(_WinTypeProgress) # _WinColRed;tColor # tTheme->wpThemeCol
```

Kontakt

ThemeColFillBarShadeNormal



Zielfüllfarbe des Farbverlaufs der Bar bei Status "normal"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Zielfüllfarbe des Farbverlaufs der Bar, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt nicht gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.



Zusätzlich muss die Eigenschaft ThemeOptFillShade auf einen Wert ungleich WinShadeSolid gesetzt sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeProgress

Beispiele:

```
tTheme->wpThemeColFillBarShadeNormal(_WinTypeProgress) # _WinColRed;tColor # tTheme->wpThemeColFi
```

Kontakt

ThemeColFillBase

Füllfarbe

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCheckBox
- WinTypeCodeEditList
- WinTypeDataList
- WinTypeDataListPopup
- WinTypeDocView
- WinTypeGanttAxis
- WinTypeGanttGraph
- WinTypeMenuItem
- WinTypeMetaPicture
- WinTypeNotebook
- WinTypeNotebookPage
- WinTypePicture
- WinTypePopupColor
- WinTypePopupList
- WinTypeRadioButton
- WinTypeRecList
- WinTypeRecListPopup
- WinTypeScrollbox
- WinTypeStatusbar
- WinTypeStatusBarButton
- WinTypeStoList
- WinTypeStoListPopup
- WinTypeToolbar
- WinTypeToolbarButton
- WinTypeToolbarMenu
- WinTypeWindowbar

Beispiele:

```
tTheme->wpThemeColFillBase(_WinTypeCheckBox) # _WinColRed;tColor # tTheme->wpThemeColFillBase(_Wi
```

Kontakt

ThemeColFillCaption

Füllfarbe der Titelzeile

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe der Titelzeile. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeGanttAxis

Beispiele:

```
tTheme->wpThemeColFillCaption(_WinTypeGanttAxis) # _WinColRed;tColor # tTheme->wpThemeColFillCapt
```

Kontakt

ThemeColFillCaptionActive



Füllfarbe der Titelzeile bei Status "aktiv"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe der Titelzeile, wenn das Objekt oder eines der Unterobjekte den Fokus besitzt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeAppFrame
- WinTypeFrame
- WinTypeGroupSplit
- WinTypeGroupTile
- WinTypeMdiFrame
- WinTypeNotebook
- WinTypePrtPreviewDlg

Beispiele:

```
tTheme->wpThemeColFillCaptionActive(_WinTypeAppFrame) # _WinColRed;tColor # tTheme->wpThemeColFil
```

Kontakt

ThemeColFillCaptionInactive



Füllfarbe der Titelzeile bei Status "inaktiv"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe der Titelzeile, wenn weder das Objekt, noch eines der Unterobjekte den Fokus besitzt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeAppFrame
- WinTypeFrame
- WinTypeGroupSplit
- WinTypeGroupTile
- WinTypeMdiFrame
- WinTypeNotebook
- WinTypePrtPreviewDlg

Beispiele:

```
tTheme->wpThemeColFillCaptionInactive(_WinTypeAppFrame) # _WinColRed;tColor # tTheme->wpThemeColF
```

Kontakt

ThemeColFillChannelDisabled Füllfarbe des
Kanals bei Status Disabled Typ int



Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe des Kanals, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeSlider

Beispiele:

```
tTheme->wpThemeColFillChannelDisabled(_WinTypeSlider) # _WinColRed;tColor # tTheme->wpThemeColFil
```

Kontakt

ThemeColFillChannelHighlight



Füllfarbe des Kanals bei Status "hervorgehoben"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe des Kanals, wenn sich die Maus über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeSlider

Beispiele:

```
tTheme->wpThemeColFillChannelHighlight(_WinTypeSlider) # _WinColRed;tColor # tTheme->wpThemeColFi
```

Kontakt

ThemeColFillChannelNormal



Füllfarbe des Kanals bei Status "normal"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe des Kanals, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt nicht gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeSlider

Beispiele:

```
tTheme->wpThemeColFillChannelNormal(_WinTypeSlider) # _WinColRed;tColor # tTheme->wpThemeColFillC
```

Kontakt

ThemeColFillChannelPressed



Füllfarbe des Kanals bei Status "gedrückt"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe des Kanals, wenn die linke Maustaste auf dem jeweiligen Oberflächenobjekt gedrückt wird. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeSlider

Beispiele:

```
tTheme->wpThemeColFillChannelPressed(_WinTypeSlider) # _WinColRed;tColor # tTheme->wpThemeColFill
```

Kontakt

ThemeColFillCheckDisabled



Füllfarbe des Bildes bei gesetztem MenuCheck bei Status Disabled

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe des Bildes, wenn MenuCheck und Disabled gesetzt sind. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeMenuItem

Beispiele:

```
tTheme->wpThemeColFillCheckDisabled(_WinTypeMenuItem) # _WinColRed;tColor # tTheme->wpThemeColFil
```

Kontakt

ThemeColFillChecked Füllfarbe

bei Status Toggled Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe, wenn die Eigenschaft Toggled des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeButton
- WinTypeColorButton
- WinTypeMenuButton
- WinTypeToolbarButton

Beispiele:

```
tTheme->wpThemeColFillChecked(_WinTypeToolbarButton) # _WinColRed;tColor # tTheme->wpThemeColFill
```

Kontakt



ThemeColFillCheckHighlight

Füllfarbe des Bildes bei gesetztem MenuCheck bei Status "hervorgehoben" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe des Bildes, wenn sich die Maus über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeMenuItem

Beispiele:

```
tTheme->wpThemeColFillCheckHighlight(_WinTypeMenuItem) # _WinColRed;tColor # tTheme->wpThemeColFi
```

Kontakt

ThemeColFillCheckHighlight



Füllfarbe des Bildes bei gesetztem MenuCheck bei Status "normal"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe des Bildes, wenn sich die Maus nicht über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeMenuItem

Beispiele:

```
tTheme->wpThemeColFillCheckNormal(_WinTypeMenuItem) # _WinColRed;tColor # tTheme->wpThemeColFillC
```

Kontakt

ThemeColFillDisabled Füllfarbe

bei Status Disabled Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBarcode
- WinTypeBigIntEdit
- WinTypeButton
- WinTypeCheckBox
- WinTypeCodeEdit
- WinTypeCodeEditListColumn
- WinTypeColorButton
- WinTypeColorEdit
- WinTypeColumnCheckbox
- WinTypeColumnRadiobutton
- WinTypeDataListColumn
- WinTypeDataListPopupColumn
- WinTypeDateEdit
- WinTypeDecimalEdit
- WinTypeEdit
- WinTypeFloatEdit
- WinTypeFontNameEdit
- WinTypeFontSizeEdit
- WinTypeGanttGraph
- WinTypeGroupBox
- WinTypeHyperLink
- WinTypeIntEdit
- WinTypeLabel
- WinTypeMenuButton
- WinTypeMenuItem
- WinTypeMetaPicture
- WinTypeNotebook
- WinTypePicture
- WinTypeProgress
- WinTypeRadioButton
- WinTypeRecListColumn
- WinTypeRecListPopupColumn
- WinTypeSlider
- WinTypeStoListColumn
- WinTypeStoListPopupColumn
- WinTypeTextEdit
- WinTypeTimeEdit

Kontakt

- WinTypeToolbarButton
- WinTypeTreeNode
- WinTypeTreeView
- WinTypeWebNavigator



Bei dem WebNavigator wirkt sich diese Eigenschaft nur aus, solange die Caption-Eigenschaft leer ist.

Beispiele:

```
tTheme->wpThemeColFillDisabled(_WinTypeToolbarButton) # _WinColRed;tColor # tTheme->wpThemeColFil
```

Kontakt

ThemeColFillEnd



Füllfarbe des Unbenutzen Bereiches der Achse

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe des Unbenutzen Bereiches der Achsen-Skala. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeGanttAxis

Beispiele:

```
tTheme->wpThemeColFillEnd(_WinTypeGanttAxis) # _WinColRed;tColor # tTheme->wpThemeColFillEnd(_Win
```

Kontakt

ThemeColFillError Füllfarbe bei

Status "Fehler" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe, wenn der Objektinhalt ungültig ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBigintEdit
- WinTypeColorEdit
- WinTypeDateEdit
- WinTypeDecimalEdit
- WinTypeEdit
- WinTypeFloatEdit
- WinTypeFontNameEdit
- WinTypeFontSizeEdit
- WinTypeIntEdit
- WinTypeTextEdit
- WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColFillError(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeColFillError
```

Kontakt

ThemeColFillFocussed Füllfarbe bei

Status "fokussiert" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe, wenn das jeweilige Oberflächenobjekt den Fokus besitzt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBigIntEdit
- WinTypeColorEdit
- WinTypeDateEdit
- WinTypeDecimalEdit
- WinTypeDocView
- WinTypeEdit
- WinTypeFloatEdit
- WinTypeFontNameEdit
- WinTypeFontSizeEdit
- WinTypeIntEdit
- WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColFillFocussed(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeColFillFo
```

Kontakt

ThemeColFillHdrBase



Füllfarbe für Spaltenköpfe Typ

int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe für den Bereich, in dem die Spaltenköpfe dargestellt werden.

Die Spaltenköpfe selber erscheinen hingegen nur in der Farbe, wenn die durch den Status definierte Füllfarbe der nachfolgend aufgelisteten Eigenschaften auf WinColUndefined oder WinColTransparent steht:

- ThemeColFillHdrNormal
- ThemeColFillHdrHighlight
- ThemeColFillHdrPressed
- ThemeColFillHdrDisabled

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeDataList
- WinTypeDataListPopup
- WinTypeRecList
- WinTypeRecListPopup
- WinTypeStoList
- WinTypeStoListPopup

Beispiele:

```
tTheme->wpThemeColFillHdrBase(_WinTypeRecList) # _WinColRed;tColor # tTheme->wpThemeColFillHdrBas
```

Kontakt

ThemeColFillHdrDisabled



Füllfarbe bei Status Disabled Typ

int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe der Spaltenköpfe, wenn die Eigenschaft Disabled des jeweiligen Listenobjektes gesetzt ist.

Ist der Wert der Eigenschaft WinColUndefined oder WinColTransparent, dann definiert ThemeColFillHdrBase die Füllfarbe.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeDataList
- WinTypeDataListPopup
- WinTypeRecList
- WinTypeRecListPopup
- WinTypeStoList
- WinTypeStoListPopup

Beispiele:

```
tTheme->wpThemeColFillHdrDisabled(_WinTypeRecList) # _WinColRed;tColor # tTheme->wpThemeColFillHd
```

Kontakt

ThemeColFillHdrHighlight



Füllfarbe bei Status "hervorgehoben"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe der Spaltenköpfe, wenn sich die Maus über der jeweiligen Spalte befindet.

Ist der Wert der Eigenschaft WinColUndefined oder WinColTransparent, dann definiert ThemeColFillHdrBase die Füllfarbe.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeDataList
- WinTypeDataListPopup
- WinTypeRecList
- WinTypeRecListPopup
- WinTypeStoList
- WinTypeStoListPopup

Beispiele:

```
tTheme->wpThemeColFillHdrHighlight(_WinTypeRecList) # _WinColRed;tColor # tTheme->wpThemeColFillH
```

Kontakt

ThemeColFillHdrNormal



Füllfarbe bei Status "normal" Typ

int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe der Spaltenköpfen, wenn die Eigenschaft Disabled bei der Liste nicht gesetzt ist und sich die Maus nicht über der Spalte befindet.

Ist der Wert der Eigenschaft WinColUndefined oder WinColTransparent, dann definiert ThemeColFillHdrBase die Füllfarbe.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeDataList
- WinTypeDataListPopup
- WinTypeRecList
- WinTypeRecListPopup
- WinTypeStoList
- WinTypeStoListPopup

Beispiele:

```
tTheme->wpThemeColFillHdrNormal(_WinTypeRecList) # _WinColRed;tColor # tTheme->wpThemeColFillHdrN
```

Kontakt

ThemeColFillHdrPressed Füllfarbe 

bei Status "gedrückt" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe der Spaltenköpfe, wenn die linke Maustaste auf der jeweiligen Spalte gedrückt wird.

Ist der Wert der Eigenschaft WinColUndefined oder WinColTransparent, dann definiert ThemeColFillHdrBase die Füllfarbe.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeDataList
- WinTypeDataListPopup
- WinTypeRecList
- WinTypeRecListPopup
- WinTypeStoList
- WinTypeStoListPopup

Beispiele:

```
tTheme->wpThemeColFillHdrPressed(_WinTypeRecList) # _WinColRed;tColor # tTheme->wpThemeColFillHdr
```

Kontakt

ThemeColFillHighlight



Füllfarbe bei Status "hervorgehoben"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe, wenn sich die Maus über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBigIntEdit
- WinTypeButton
- WinTypeCheckBox
- WinTypeColorButton
- WinTypeColorEdit
- WinTypeDateEdit
- WinTypeDecimalEdit
- WinTypeEdit
- WinTypeFloatEdit
- WinTypeFontNameEdit
- WinTypeFontSizeEdit
- WinTypeIntEdit
- WinTypeMenuButton
- WinTypeMenuItem
- WinTypeNotebook
- WinTypeRadioButton
- WinTypeTextEdit
- WinTypeTimeEdit
- WinTypeToolbarButton

Beispiele:

```
tTheme->wpThemeColFillHighlight(_WinTypeToolbarButton) # _WinColRed;tColor # tTheme->wpThemeColFi
```

Kontakt

ThemeColFillHover



Füllfarbe bei Status "mouseover"

Typ [int](#)

Siehe [Liste, Objekte](#)

In dieser Eigenschaft können [Farbkonstanten](#) gesetzt werden.

Diese Eigenschaft definiert in einem [Theme](#)-Objekt die Füllfarbe, wenn sich die Maus über dem jeweiligen Oberflächenobjekt befindet. Das [Theme](#)-Objekt muss zuvor mit dem Befehl [WinThemeOpen\(\)](#) geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- [WinTypeHyperLink](#)

Beispiele:

```
tTheme->wpThemeColFillHover(_WinTypeHyperLink) # _WinColRed;tColor # tTheme->wpThemeColFillHover(
```

Kontakt

ThemeColFillInactive Füllfarbe

bei Status "inaktiv" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe, wenn das Oberflächenobjekt nicht den Eingabefokus besitzt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEditListColumn
- WinTypeColumnCheckbox
- WinTypeColumnRadiobutton
- WinTypeDataListColumn
- WinTypeDataListPopupColumn
- WinTypeRecListColumn
- WinTypeRecListPopupColumn
- WinTypeStoListColumn
- WinTypeStoListPopupColumn
- WinTypeTreeNode

Beispiele:

```
tTheme->wpThemeColFillInactive(_WinTypeDataListColumn) # _WinColRed;tColor # tTheme->wpThemeColFi
```

Kontakt



ThemeColFillIndentNormal

Füllfarbe der Zeileneinzugssymbole des RtfEdit-Objektes bei Status "normal" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe der Zeileneinzugssymbole im RtfEdit-Objekt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeRtfEdit

Beispiele:

```
tTheme->wpThemeColFillIndentNormal(_WinTypeRtfEdit) # _WinColRed;tColor # tTheme->wpThemeColFillI
```

Kontakt

ThemeColFillNormal Füllfarbe

bei Status "normal" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe, wenn sich die Maus nicht über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeAppFrame
- WinTypeBarcode
- WinTypeBigIntEdit
- WinTypeButton
- WinTypeCheckBox
- WinTypeCodeEdit
- WinTypeColorButton
- WinTypeColorEdit
- WinTypeDateEdit
- WinTypeDecimalEdit
- WinTypeDivider
- WinTypeDocView
- WinTypeEdit
- WinTypeFloatEdit
- WinTypeFontNameEdit
- WinTypeFontSizeEdit
- WinTypeFrame
- WinTypeGanttAxis
- WinTypeGanttGraph
- WinTypeGroupBox
- WinTypeGroupSplit
- WinTypeGroupTile
- WinTypeHelpTip
- WinTypeHyperLink
- WinTypeIcon
- WinTypeIntEdit
- WinTypeLabel
- WinTypeMdiFrame
- WinTypeMenuItem
- WinTypeMetaPicture
- WinTypePicture
- WinTypeProgress
- WinTypePrtPreviewDlg
- WinTypeRadioButton
- WinTypeRTFEdit
- WinTypeScrollbox

Kontakt

- WinTypeSlider
- WinTypeStatusbar
- WinTypeStatusbarButton
- WinTypeTextEdit
- WinTypeTimeEdit
- WinTypeToolbar
- WinTypeToolbarButton
- WinTypeToolbarDock
- WinTypeToolbarMenu
- WinTypeTreeView
- WinTypeWebNavigator
- WinTypeWindowbar



Bei dem WebNavigator wirkt sich diese Eigenschaft nur aus, solange die Caption-Eigenschaft leer ist.

Beispiele:

```
tTheme->wpThemeColFillNormal(_WinTypeToolbarButton) # _WinColRed;tColor # tTheme->wpThemeColFillN
```

Kontakt

ThemeColFillPressed Füllfarbe bei

Status "gedrückt" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe, wenn das jeweilige Oberflächenobjekt gedrückt wird. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeButton
- WinTypeCheckBox
- WinTypeColorButton
- WinTypeMenuButton
- WinTypeNotebook
- WinTypeRadioButton
- WinTypeToolbarButton

Beispiele:

```
tTheme->wpThemeColFillPressed(_WinTypeToolbarButton) # _WinColRed;tColor # tTheme->wpThemeColFill
```

Kontakt

ThemeColFillReadOnly Füllfarbe bei

Status "read-only" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe, wenn die Eigenschaft EditorReadOnly des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEdit

Beispiele:

```
tTheme->wpThemeColFillReadOnly(_WinTypeCodeEdit) # _WinColLightGray;tColor # tTheme->wpThemeColFi
```

Kontakt

ThemeColFillRulerNormal



Füllfarbe des Lineals des RtfEdit-Objektes bei Status "normal"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe des Lineals im RtfEdit-Objekt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeRtfEdit

Beispiele:

```
tTheme->wpThemeColFillRulerNormal(_WinTypeRtfEdit) # _WinColRed;tColor # tTheme->wpThemeColFillRu
```

Kontakt

ThemeColFillSelectedActive



Füllfarbe bei Status "selektiert und aktiv"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe, wenn das Oberflächenobjekt selektiert ist und den Eingabefokus besitzt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEditListColumn
- WinTypeColumnCheckbox
- WinTypeColumnRadiobutton
- WinTypeDataListColumn
- WinTypeDataListPopupColumn
- WinTypeRecListColumn
- WinTypeRecListPopupColumn
- WinTypeStoListColumn
- WinTypeStoListPopupColumn
- WinTypeTreeNode

Beispiele:

```
tTheme->wpThemeColFillSelectedActive(_WinTypeDataListColumn) # _WinColRed;tColor # tTheme->wpThem
```

Kontakt

ThemeColFillSelectedInactive



Füllfarbe bei Status "selektiert und inaktiv"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe, wenn das Oberflächenobjekt selektiert ist, jedoch nicht den Eingabefokus besitzt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEditListColumn
- WinTypeColumnCheckbox
- WinTypeColumnRadiobutton
- WinTypeDataListColumn
- WinTypeDataListPopupColumn
- WinTypeRecListColumn
- WinTypeRecListPopupColumn
- WinTypeStoListColumn
- WinTypeStoListPopupColumn
- WinTypeTreeNode

Beispiele:

```
tTheme->wpThemeColFillSelectedInactive(_WinTypeDataListColumn) # _WinColRed;tColor # tTheme->wpTh
```

Kontakt

ThemeColFillShadeChecked



Zielfüllfarbe des Farbverlaufs bei Status Toggled

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Zielfüllfarbe des Farbverlaufs, wenn die Eigenschaft Toggled des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.



Zusätzlich muss die Eigenschaft ThemeOptFillShade auf einen Wert ungleich WinShadeSolid gesetzt sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeButton
- WinTypeToolbarButton

Beispiele:

```
tTheme->wpThemeColFillShadeChecked(_WinTypeToolbarButton) # _WinColRed;tColor # tTheme->wpThemeCo
```

Kontakt

ThemeColFillShadeDisabled



Zielfüllfarbe des Farbverlaufs bei Status Disabled

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Zielfüllfarbe des Farbverlaufs, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.



Zusätzlich muss die Eigenschaft ThemeOptFillShade auf einen Wert ungleich WinShadeSolid gesetzt sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeButton
- WinTypeGroupBox
- WinTypeProgress
- WinTypeSlider
- WinTypeToolbarButton

Beispiele:

```
tTheme->wpThemeColFillShadeDisabled(_WinTypeToolbarButton) # _WinColRed;tColor # tTheme->wpThemeC
```

Kontakt

ThemeColFillShadeHighlight



Zielfüllfarbe des Farbverlaufs bei Status "hervorgehoben"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Zielfüllfarbe des Farbverlaufs, wenn sich die Maus über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.



Zusätzlich muss die Eigenschaft ThemeOptFillShade auf einen Wert ungleich WinShadeSolid gesetzt sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeButton
- WinTypeToolbarButton

Beispiele:

```
tTheme->wpThemeColFillShadeHighlight(_WinTypeToolbarButton) # _WinColRed;tColor # tTheme->wpTheme
```

Kontakt

ThemeColFillShadeNormal



Zielfüllfarbe des Farbverlaufs bei Status "normal"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Zielfüllfarbe des Farbverlaufs, wenn sich die Maus nicht über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.



Zusätzlich muss die Eigenschaft ThemeOptFillShade auf einen Wert ungleich WinShadeSolid gesetzt sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeButton
- WinTypeGroupBox
- WinTypeGroupTile
- WinTypeNotebookPage
- WinTypeProgress
- WinTypeSlider
- WinTypeToolbarButton

Beispiele:

```
tTheme->wpThemeColFillShadeNormal(_WinTypeToolbarButton) # _WinColRed;tColor # tTheme->wpThemeCol
```

Kontakt

ThemeColFillShadePressed



Zielfüllfarbe des Farbverlaufs bei Status "gedrückt"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Zielfüllfarbe des Farbverlaufs, wenn die linke Maustaste auf dem jeweiligen Oberflächenobjekt gedrückt wird. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.



Zusätzlich muss die Eigenschaft ThemeOptFillShade auf einen Wert ungleich WinShadeSolid gesetzt sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeButton
- WinTypeToolbarButton

Beispiele:

```
tTheme->wpThemeColFillShadePressed(_WinTypeToolbarButton) # _WinColRed;tColor # tTheme->wpThemeCo
```

Kontakt

ThemeColFillTabNormal



Füllfarbe der Tabulatoren des **RtfEdit**-Objektes bei Status "normal"

Typ **int**

Siehe [Liste](#), [Objekte](#)

In dieser Eigenschaft können [Farbkonstanten](#) gesetzt werden.

Diese Eigenschaft definiert in einem [Theme](#)-Objekt die Füllfarbe für Tabulatorenzeichen im [RtfEdit](#)-Objekt. Das [Theme](#)-Objekt muss zuvor mit dem Befehl [WinThemeOpen\(\)](#) geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- [WinTypeRtfEdit](#)

Beispiele:

```
tTheme->wpThemeColFillTabNormal(_WinTypeRtfEdit) # _WinColRed;tColor # tTheme->wpThemeColFillTabN
```

Kontakt

ThemeColFillThumbDisabled



Füllfarbe des Schiebereglers bei Status Disabled

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe des Schiebereglers, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeSlider

Beispiele:

```
tTheme->wpThemeColFillThumbDisabled(_WinTypeSlider) # _WinColRed;tColor # tTheme->wpThemeColFillT
```

Kontakt

ThemeColFillThumbHighlight



Füllfarbe des Schiebereglers bei Status "hervorgehoben"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe des Schiebereglers, wenn sich die Maus über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeSlider

Beispiele:

```
tTheme->wpThemeColFillThumbHighlight(_WinTypeSlider) # _WinColRed;tColor # tTheme->wpThemeColFill
```

Kontakt

ThemeColFillThumbNormal



Füllfarbe des Schiebereglers bei Status "normal"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe des Schiebereglers, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt nicht gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeSlider

Beispiele:

```
tTheme->wpThemeColFillThumbNormal(_WinTypeSlider) # _WinColRed;tColor # tTheme->wpThemeColFillThu
```

Kontakt

ThemeColFillThumbPressed



Füllfarbe des Schiebereglers bei Status "gedrückt"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe des Schiebereglers, wenn die linke Maustaste auf dem jeweiligen Oberflächenobjekt gedrückt wird. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeSlider

Beispiele:

```
tTheme->wpThemeColFillThumbPressed(_WinTypeSlider) # _WinColRed;tColor # tTheme->wpThemeColFillTh
```

Kontakt

ThemeColFillVisited Füllfarbe bei

Status "besucht" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe, wenn sich die Maus nicht über dem jeweiligen Oberflächenobjekt befindet und die verknüpfte Seite besucht wurde. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeHyperLink

Beispiele:

```
tTheme->wpThemeColFillVisited(_WinTypeHyperLink) # _WinColRed;tColor # tTheme->wpThemeColFillVisi
```

Kontakt

ThemeOptFillShade

Modus des Farbverlaufs

Typ int

Siehe Liste, Objekte

Diese Eigenschaft definiert in einem Theme-Objekt den Modus des Farbverlaufs. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Folgende Konstanten können angegeben werden:

WinShadeSolid

Das Objekt wird mit der Hauptfarbe (ThemeColFill...) des jeweiligen Status gefüllt.

WinShadeTopBottom Der Farbverlauf wird von oben (ThemeColFill...) nach unten

(ThemeColFill*Shade...) gezeichnet.

WinShadeBottomTop Der Farbverlauf wird von unten nach oben gezeichnet.

WinShadeLeftRight Der Farbverlauf wird von links nach rechts gezeichnet.

WinShadeRightLeft Der Farbverlauf wird von rechts nach links gezeichnet.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeButton
- WinTypeGroupBox
- WinTypeGroupTile
- WinTypeNotebookPage
- WinTypeProgress
- WinTypeSlider
- WinTypeToolbarButton

Beispiele:

```
tTheme->wpThemeOptFillShade(_WinTypeToolbarButton) # _WinShadeLeftRight;tShade # tTheme->wpThemeO
```

Kontakt

Text Color (Farb-Eigenschaften für Textfarbe von Themes)

Eigenschaften

sortiert nach

Gruppen,

Siehe

Alphabetische

Liste aller

- ThemeColTextActive
- ThemeColTextCaption
- ThemeColTextCaptionActive
- ThemeColTextCaptionInactive
- ThemeColTextChecked
- ThemeColTextDisabled
- ThemeColTextError
- ThemeColTextFocussed
- ThemeColTextHdrDisabled
- ThemeColTextHdrHighlight
- ThemeColTextHdrNormal
- ThemeColTextHdrPressed
- ThemeColTextHighlight
- ThemeColTextHover
- ThemeColTextInactive
- ThemeColTextNormal
- ThemeColTextPressed
- ThemeColTextRulerNormal
- ThemeColTextSelectedActive
- ThemeColTextSelectedInactive
- ThemeColTextVisited

Kontakt

ThemeColTextActive Textfarbe

bei Status "aktiv" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe, wenn das Oberflächenobjekt den Eingabefokus besitzt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEditListColumn
- WinTypeDataListColumn
- WinTypeDataListPopupColumn
- WinTypeRecListColumn
- WinTypeRecListPopupColumn
- WinTypeStoListColumn
- WinTypeStoListPopupColumn
- WinTypeTreeNode

Beispiele:

```
tTheme->wpThemeColTextActive(_WinTypeDataListColumn) # _WinColRed;tColor # tTheme->wpThemeColText
```

Kontakt

ThemeColTextCaption

Textfarbe der Titelzeile

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe der Titelzeile. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeGanttAxis

Beispiele:

```
tTheme->wpThemeColTextCaption(_WinTypeGanttAxis) # _WinColRed;tColor # tTheme->wpThemeColTextCapt
```

Kontakt

ThemeColTextCaptionActive



Textfarbe der Titelzeile bei Status "aktiv"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe der Titelzeile, wenn das Objekt oder eines der Unterobjekte den Fokus besitzt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeAppFrame
- WinTypeFrame
- WinTypeGroupSplit
- WinTypeGroupTile
- WinTypeMdiFrame
- WinTypeNotebook
- WinTypePrtPreviewDlg

Beispiele:

```
tTheme->wpThemeColTextCaptionActive(_WinTypeAppFrame) # _WinColRed;tColor # tTheme->wpThemeColTex
```

Kontakt

ThemeColTextCaptionInactive



Textfarbe der Titelzeile bei Status "inaktiv"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe der Titelzeile, wenn weder das Objekt, noch eines der Unterobjekte den Fokus besitzt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeAppFrame
- WinTypeFrame
- WinTypeGroupSplit
- WinTypeGroupTile
- WinTypeMdiFrame
- WinTypeNotebook
- WinTypePrtPreviewDlg

Beispiele:

```
tTheme->wpThemeColTextCaptionInactive(_WinTypeAppFrame) # _WinColRed;tColor # tTheme->wpThemeColT
```

Kontakt

ThemeColTextChecked Textfarbe

bei Status Toggled Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe, wenn die Eigenschaft Toggled des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeButton
- WinTypeColorButton
- WinTypeMenuButton
- WinTypeToolbarButton

Beispiele:

```
tTheme->wpThemeColTextChecked(_WinTypeButton) # _WinColRed;tColor # tTheme->wpThemeColTextChecked
```

Kontakt

ThemeColTextDisabled Textfarbe

bei Status Disabled Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBarcode
- WinTypeBigIntEdit
- WinTypeButton
- WinTypeCheckBox
- WinTypeCodeEditListColumn
- WinTypeColorButton
- WinTypeColorEdit
- WinTypeDataListColumn
- WinTypeDataListPopupColumn
- WinTypeDateEdit
- WinTypeDecimalEdit
- WinTypeEdit
- WinTypeFloatEdit
- WinTypeFontNameEdit
- WinTypeFontSizeEdit
- WinTypeGroupBox
- WinTypeHyperLink
- WinTypeIntEdit
- WinTypeLabel
- WinTypeMenuButton
- WinTypeMenuItem
- WinTypeMetaPicture
- WinTypeNotebook
- WinTypePicture
- WinTypeProgress
- WinTypeRadioButton
- WinTypeRecListColumn
- WinTypeRecListPopupColumn
- WinTypeSlider
- WinTypeStoListColumn
- WinTypeStoListPopupColumn
- WinTypeTextEdit
- WinTypeTimeEdit
- WinTypeToolbarButton
- WinTypeTreeNode

Beispiele:

Kontakt

```
tTheme->wpThemeColTextDisabled(_WinTypeButton) # _WinColRed;tColor # tTheme->wpThemeColTextDisabl
```

Kontakt

ThemeColTextError Textfarbe bei

Status "Fehler" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe, wenn der Objektinhalt ungültig ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBigintEdit
- WinTypeColorEdit
- WinTypeDateEdit
- WinTypeDecimalEdit
- WinTypeEdit
- WinTypeFloatEdit
- WinTypeFontNameEdit
- WinTypeFontSizeEdit
- WinTypeIntEdit
- WinTypeTextEdit
- WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColTextError(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeColTextError
```

Kontakt

ThemeColTextFocussed Textfarbe bei

Status "fokussiert" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe, wenn das jeweilige Oberflächenobjekt den Fokus besitzt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBigIntEdit
- WinTypeColorEdit
- WinTypeDateEdit
- WinTypeDecimalEdit
- WinTypeEdit
- WinTypeFloatEdit
- WinTypeFontNameEdit
- WinTypeFontSizeEdit
- WinTypeIntEdit
- WinTypeTextEdit
- WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColTextFocussed(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeColTextFo
```

Kontakt

ThemeColTextHdrDisabled



Textfarbe bei Status Disabled Typ

int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe der Spaltenköpfe, wenn die Eigenschaft Disabled des jeweiligen Listenobjektes gesetzt ist.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeDataList
- WinTypeDataListPopup
- WinTypeRecList
- WinTypeRecListPopup
- WinTypeStoList
- WinTypeStoListPopup

Beispiele:

```
tTheme->wpThemeColTextHdrDisabled(_WinTypeRecList) # _WinColGray;tColor # tTheme->wpThemeColTextH
```

Kontakt

ThemeColTextHdrHighlight Textfarbe bei

Status "hervorgehoben" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe der Spaltenköpfe, wenn sich die Maus über der jeweiligen Spalte befindet.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeDataList
- WinTypeDataListPopup
- WinTypeRecList
- WinTypeRecListPopup
- WinTypeStoList
- WinTypeStoListPopup

Beispiele:

```
tTheme->wpThemeColTextHdrHighlight(_WinTypeRecList) # _WinColLightBlue;tColor # tTheme->wpThemeCo
```

Kontakt

ThemeColTextHdrNormal



Textfarbe bei Status "normal" Typ

int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe der Spaltenköpfe, wenn die Eigenschaft Disabled bei der Liste nicht gesetzt ist und sich die Maus nicht über der Spalte befindet.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeDataList
- WinTypeDataListPopup
- WinTypeRecList
- WinTypeRecListPopup
- WinTypeStoList
- WinTypeStoListPopup

Beispiele:

```
tTheme->wpThemeColTextHdrNormal(_WinTypeRecList) # _WinColBlack;tColor # tTheme->wpThemeColTextHd
```

Kontakt

ThemeColTextHdrPressed Textfarbe

bei Status "gedrückt" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe der Spaltenköpfe, wenn die linke Maustaste auf der jeweiligen Spalte gedrückt wird.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeDataList
- WinTypeDataListPopup
- WinTypeRecList
- WinTypeRecListPopup
- WinTypeStoList
- WinTypeStoListPopup

Beispiele:

```
tTheme->wpThemeColTextHdrPressed(_WinTypeRecList) # _WinColRed;tColor # tTheme->wpThemeColTextHdr
```

Kontakt

ThemeColTextHighlight



Textfarbe bei Status "hervorgehoben"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe, wenn sich die Maus über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBigIntEdit
- WinTypeButton
- WinTypeCheckBox
- WinTypeColorButton
- WinTypeColorEdit
- WinTypeDateEdit
- WinTypeDecimalEdit
- WinTypeEdit
- WinTypeFloatEdit
- WinTypeFontNameEdit
- WinTypeFontSizeEdit
- WinTypeIntEdit
- WinTypeMenuButton
- WinTypeMenuItem
- WinTypeNotebook
- WinTypeRadioButton
- WinTypeTextEdit
- WinTypeTimeEdit
- WinTypeToolbarButton

Beispiele:

```
tTheme->wpThemeColTextHighlight(_WinTypeButton) # _WinColRed;tColor # tTheme->wpThemeColTextHighl
```

Kontakt

ThemeColTextHover



Textfarbe bei Status "mouseover"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe, wenn sich die Maus über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeHyperLink

Beispiele:

```
tTheme->wpThemeColTextHover(_WinTypeHyperLink) # _WinColRed;tColor # tTheme->wpThemeColTextHover(
```

Kontakt

ThemeColTextInactive Textfarbe

bei Status "inaktiv" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe, wenn das Oberflächenobjekt nicht den Eingabefokus besitzt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEditListColumn
- WinTypeDataListColumn
- WinTypeDataListPopupColumn
- WinTypeRecListColumn
- WinTypeRecListPopupColumn
- WinTypeStoListColumn
- WinTypeStoListPopupColumn
- WinTypeTreeNode

Beispiele:

```
tTheme->wpThemeColTextInactive(_WinTypeDataListColumn) # _WinColRed;tColor # tTheme->wpThemeColTe
```

Kontakt

ThemeColTextNormal Textfarbe

bei Status "normal" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe, wenn sich die Maus nicht über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBarcode
- WinTypeBigIntEdit
- WinTypeButton
- WinTypeCheckBox
- WinTypeColorButton
- WinTypeColorEdit
- WinTypeDateEdit
- WinTypeDecimalEdit
- WinTypeEdit
- WinTypeFloatEdit
- WinTypeFontNameEdit
- WinTypeFontSizeEdit
- WinTypeGanttAxis
- WinTypeGroupBox
- WinTypeHelpTip
- WinTypeHyperLink
- WinTypeIntEdit
- WinTypeLabel
- WinTypeMenuButton
- WinTypeMenuItem
- WinTypeMetaPicture
- WinTypePicture
- WinTypeProgress
- WinTypeRadioButton
- WinTypeSlider
- WinTypeStatusBarButton
- WinTypeTextEdit
- WinTypeTimeEdit
- WinTypeToolbarButton

Beispiele:

```
tTheme->wpThemeColTextNormal(_WinTypeButton) # _WinColRed;tColor # tTheme->wpThemeColTextNormal(_
```

Kontakt

ThemeColTextPressed Textfarbe bei

Status "gedrückt" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe, wenn die linke Maustaste auf dem jeweiligen Oberflächenobjekt gedrückt wird. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeButton
- WinTypeCheckBox
- WinTypeColorButton
- WinTypeMenuButton
- WinTypeNotebook
- WinTypeRadioButton
- WinTypeToolbarButton

Beispiele:

```
tTheme->wpThemeColTextPressed(_WinTypeButton) # _WinColRed;tColor # tTheme->wpThemeColTextPressed
```

Kontakt

ThemeColTextRulerNormal



Textfarbe des Lineals des **RtfEdit**-Objektes bei Status "normal"

Typ **int**

Siehe [Liste](#), [Objekte](#)

In dieser Eigenschaft können **Farbkonstanten** gesetzt werden.

Diese Eigenschaft definiert in einem **Theme**-Objekt die Textfarbe des Lineals im **RtfEdit**-Objekt. Das **Theme**-Objekt muss zuvor mit dem Befehl [**WinThemeOpen\(\)**](#) geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- [**WinTypeRtfEdit**](#)

Beispiele:

```
tTheme->wpThemeColTextRulerNormal(_WinTypeRtfEdit) # _WinColRed;tColor # tTheme->wpThemeColTextRu
```

Kontakt

ThemeColTextSelectedActive



Textfarbe bei Status "selektiert und aktiv"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe, wenn das Oberflächenobjekt selektiert ist und den Eingabefokus besitzt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEditListColumn
- WinTypeDataListColumn
- WinTypeDataListPopupColumn
- WinTypeRecListColumn
- WinTypeRecListPopupColumn
- WinTypeStoListColumn
- WinTypeStoListPopupColumn
- WinTypeTreeNode

Beispiele:

```
tTheme->wpThemeColTextSelectedActive(_WinTypeDataListColumn) # _WinColRed;tColor # tTheme->wpThem
```

Kontakt

ThemeColTextSelectedInactive



Textfarbe bei Status "selektiert und inaktiv"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe, wenn das Oberflächenobjekt selektiert ist, jedoch nicht den Eingabefokus besitzt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEditListColumn
- WinTypeDataListColumn
- WinTypeDataListPopupColumn
- WinTypeRecListColumn
- WinTypeRecListPopupColumn
- WinTypeStoListColumn
- WinTypeStoListPopupColumn
- WinTypeTreeNode

Beispiele:

```
tTheme->wpThemeColTextSelectedInactive(_WinTypeDataListColumn) # _WinColRed;tColor # tTheme->wpTh
```

Kontakt

ThemeColTextVisited Textfarbe bei

Status "besucht" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe, wenn sich die Maus nicht über dem jeweiligen Oberflächenobjekt befindet und die verknüpfte Seite besucht wurde. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeHyperLink

Beispiele:

```
tTheme->wpThemeColTextVisited(_WinTypeHyperLink) # _WinColRed;tColor # tTheme->wpThemeColTextVisi
```

Kontakt

Arrow Color (Farb-Eigenschaften für Pfeilfarbe von Themes)

Eigenschaften

sortiert nach

Gruppen,

Siehe

Alphabetische

Liste aller

- ThemeColArrowChecked
- ThemeColArrowDisabled
- ThemeColArrowHighlight
- ThemeColArrowIvl
- ThemeColArrowNormal
- ThemeColArrowPressed
- ThemeColSortHdrDisabled
- ThemeColSortHdrHighlight
- ThemeColSortHdrNormal
- ThemeColSortHdrPressed

Kontakt

ThemeColArrowChecked



Pfeilfarbe bei Status Toggled Typ

int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für das Pfeilsymbol, wenn die Eigenschaft Toggled des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeColorButton
- WinTypeMenuButton
- WinTypeToolbarButton



Der Objekttyp WinTypeToolbarButton bezieht sich auf die ColorButton- und MenuButton-Objekte mit StyleButton = WinStyleButtonTBar.

Beispiele:

```
tTheme->wpThemeColArrowChecked(_WinTypeColorButton) # _WinColRed;tColor # tTheme->wpThemeColArrow
```

Kontakt

ThemeColArrowDisabled



Pfeilfarbe bei Status Disabled Typ

int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für das Pfeilsymbol, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeColorButton
- WinTypeMenuButton
- WinTypeToolbarButton



Der Objekttyp WinTypeToolbarButton bezieht sich auf die ColorButton- und MenuButton-Objekte mit StyleButton = WinStyleButtonTBar.

Beispiele:

```
tTheme->wpThemeColArrowDisabled(_WinTypeColorButton) # _WinColRed;tColor # tTheme->wpThemeColArro
```

Kontakt

ThemeColArrowHighlight



Pfeilfarbe bei Status "hervorgehoben"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für das Pfeilsymbol, wenn sich die Maus über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeColorButton
- WinTypeMenuButton
- WinTypeToolbarButton



Der Objekttyp WinTypeToolbarButton bezieht sich auf die ColorButton- und MenuButton-Objekte mit StyleButton = WinStyleButtonTBar.

Beispiele:

```
tTheme->wpThemeColArrowHighlight(_WinTypeColorButton) # _WinColRed;tColor # tTheme->wpThemeColArr
```

Kontakt

ThemeColArrowIvl

Pfeilfarbe für Intervalle

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für das Pfeilsymbol für Intervalle. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeGanttGraph

Beispiele:

```
tTheme->wpThemeColArrowIvl(_WinTypeGanttGraph) # _WinColRed;tColor # tTheme->wpThemeColArrowIvl(_
```

Kontakt

ThemeColArrowNormal Pfeilfarbe

bei Status "normal" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für das Pfeilsymbol, wenn weder die Eigenschaften Toggled, noch Disabled gesetzt sind und sich die Maus nicht über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeColorButton
- WinTypeMenuButton
- WinTypeToolbarButton



Der Objekttyp WinTypeToolbarButton bezieht sich auf die ColorButton- und MenuButton-Objekte mit StyleButton = WinStyleButtonTBar.

Beispiele:

```
tTheme->wpThemeColArrowNormal(_WinTypeColorButton) # _WinColRed;tColor # tTheme->wpThemeColArrowN
```

Kontakt

ThemeColArrowPressed Pfeilfarbe

bei Status "gedrückt" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für das Pfeilsymbol, wenn die linke Maustaste auf dem jeweiligen Oberflächenobjekt gedrückt wird. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeColorButton
- WinTypeMenuButton
- WinTypeToolbarButton



Der Objekttyp WinTypeToolbarButton bezieht sich auf die ColorButton- und MenuButton-Objekte mit StyleButton = WinStyleButtonTBar.

Beispiele:

```
tTheme->wpThemeColArrowPressed(_WinTypeColorButton) # _WinColRed;tColor # tTheme->wpThemeColArrow
```

Kontakt

ThemeColSortHdrDisabled



Farbe für Sortierrichtung bei Status Disabled

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für die Anzeige der Sortierrichtung, wenn die Eigenschaft Disabled des jeweiligen Listenobjektes gesetzt ist.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeDataList
- WinTypeDataListPopup
- WinTypeRecList
- WinTypeRecListPopup
- WinTypeStoList
- WinTypeStoListPopup

Beispiele:

```
tTheme->wpThemeColSortHdrDisabled(_WinTypeRecList) # _WinColGray;tColor # tTheme->wpThemeColSortH
```

Kontakt

ThemeColSortHdrHighlight



Farbe für Sortierrichtung bei Status "hervorgehoben"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für die Anzeige der Sortierrichtung, wenn sich die Maus über der jeweiligen Spalte befindet.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeDataList
- WinTypeDataListPopup
- WinTypeRecList
- WinTypeRecListPopup
- WinTypeStoList
- WinTypeStoListPopup

Beispiele:

```
tTheme->wpThemeColSortHdrHighlight(_WinTypeRecList) # _WinColBlack;tColor # tTheme->wpThemeColSor
```

Kontakt

ThemeColSortHdrNormal



Farbe für Sortierrichtung bei Status "normal"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für die Anzeige der Sortierrichtung, wenn die Eigenschaft Disabled bei der Liste nicht gesetzt ist und sich die Maus nicht über der Spalte befindet.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeDataList
- WinTypeDataListPopup
- WinTypeRecList
- WinTypeRecListPopup
- WinTypeStoList
- WinTypeStoListPopup

Beispiele:

```
tTheme->wpThemeColSortHdrNormal(_WinTypeRecList) # _WinColBlack;tColor # tTheme->wpThemeColSortHd
```

Kontakt

ThemeColSortHdrPressed



Farbe für Sortierrichtung bei Status "gedrückt"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für die Anzeige der Sortierrichtung, wenn die linke Maustaste auf der jeweiligen Spalte gedrückt wird.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeDataList
- WinTypeDataListPopup
- WinTypeRecList
- WinTypeRecListPopup
- WinTypeStoList
- WinTypeStoListPopup

Beispiele:

```
tTheme->wpThemeColSortHdrPressed(_WinTypeRecList) # _WinColBlack;tColor # tTheme->wpThemeColSortH
```

Kontakt

Border Color (Farb-Eigenschaften für Rahmenfarbe von Themes)

Eigenschaften

sortiert nach

Gruppen,

Siehe

Alphabetische

Liste aller

- [ThemeColBorderActive](#)
- [ThemeColBorderBase](#)
- [ThemeColBorderBoundary](#)
- [ThemeColBorderCellActiveHorz](#)
- [ThemeColBorderCellActiveVert](#)
- [ThemeColBorderCheckDisabled](#)
- [ThemeColBorderChecked](#)
- [ThemeColBorderCheckHighlight](#)
- [ThemeColBorderCheckNormal](#)
- [ThemeColBorderDisabled](#)
- [ThemeColBorderError](#)
- [ThemeColBorderFocussed](#)
- [ThemeColBorderHdrBase](#)
- [ThemeColBorderHighlight](#)
- [ThemeColBorderHover](#)
- [ThemeColBorderInactive](#)
- [ThemeColBorderIvl](#)
- [ThemeColBorderNormal](#)
- [ThemeColBorderPageNormal](#)
- [ThemeColBorderPressed](#)
- [ThemeColBorderReadOnly](#)
- [ThemeColBorderSelectedActive](#)
- [ThemeColBorderSelectedInactive](#)
- [ThemeColBorderVisited](#)

Kontakt

ThemeColBorderActive



Rahmenfarbe bei Status "aktiv" Typ

int

Siehe [Liste, Objekte](#)

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Rahmenfarbe, wenn das Oberflächenobjekt den Eingabefokus besitzt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeColumnCheckbox
- WinTypeColumnRadiobutton

Beispiele:

```
tTheme->wpThemeColBorderActive(_WinTypeColumnCheckbox) # _WinColRed;tColor # tTheme->wpThemeColBo
```

Kontakt

ThemeColBorderBase

Rahmenfarbe

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Rahmenfarbe. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeNotebook

Beispiele:

```
tTheme->wpThemeColBorderBase(_WinTypeNotebook) # _WinColRed;tColor # tTheme->wpThemeColBorderBase
```

Kontakt

ThemeColBorderBoundary



Farbe für das Begrenzungsrechteck

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für das Begrenzungsrechteck um die Intervalle in einem GanttGraph. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeGanttGraph

Beispiele:

```
tTheme->wpThemeColBorderBoundary(_WinTypeGanttGraph) # _WinColRed;tColor # tTheme->wpThemeColBord
```

Kontakt

ThemeColBorderCellActiveHorz



Farbe für die horizontale Zellmarkierung

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für die horizontale Zellmarkierung (aktuelle Spalte) in einem GanttGraph. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeGanttGraph

Beispiele:

```
tTheme->wpThemeColBorderCellActiveHorz(_WinTypeGanttGraph) # _WinColRed;tColor # tTheme->wpThemeC
```

Kontakt

ThemeColBorderCellActiveVert



Farbe für die vertikale Zellmarkierung

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für die vertikale Zellmarkierung (aktuelle Zeile) in einem GanttGraph. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeGanttGraph

Beispiele:

```
tTheme->wpThemeColBorderCellActiveVert(_WinTypeGanttGraph) # _WinColRed;tColor # tTheme->wpThemeC
```

Kontakt

ThemeColBorderCheckDisabled



Rahmenfarbe des Bildes bei gesetztem MenuCheck bei Status Disabled Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Rahmenfarbe des Bildes, wenn MenuCheck und Disabled gesetzt sind. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeMenuItem

Beispiele:

```
tTheme->wpThemeColBorderCheckDisabled(_WinTypeMenuItem) # _WinColRed;tColor # tTheme->wpThemeColB
```

Kontakt

ThemeColBorderChecked



Rahmenfarbe bei Status Toggled Typ

int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Rahmenfarbe, wenn die Eigenschaft Toggled des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeButton
- WinTypeColorButton
- WinTypeMenuButton

Beispiele:

```
tTheme->wpThemeColBorderChecked(_WinTypeColorButton) # _WinColRed;tColor # tTheme->wpThemeColBord
```

Kontakt

ThemeColBorderCheckHighlight



Rahmenfarbe des Bildes bei gesetztem MenuCheck bei Status "hervorgehoben" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Rahmenfarbe des Bildes, wenn sich die Maus über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeMenuItem

Beispiele:

```
tTheme->wpThemeColBorderCheckHighlight(_WinTypeMenuItem) # _WinColRed;tColor # tTheme->wpThemeCol
```

Kontakt



ThemeColBorderCheckNormal

Rahmenfarbe des Bildes bei gesetztem MenuCheck bei Status "normal"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Rahmenfarbe des Bildes, wenn sich die Maus nicht über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeMenuItem

Beispiele:

```
tTheme->wpThemeColBorderCheckNormal(_WinTypeMenuItem) # _WinColRed;tColor # tTheme->wpThemeColBor
```

Kontakt

ThemeColBorderDisabled



Rahmenfarbe bei Status Disabled Typ

int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Rahmenfarbe, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeAnimation
- WinTypeBarcode
- WinTypeBigIntEdit
- WinTypeButton
- WinTypeCheckBox
- WinTypeCodeEdit
- WinTypeCodeEditList
- WinTypeColorButton
- WinTypeColorEdit
- WinTypeColumnCheckbox
- WinTypeColumnRadiobutton
- WinTypeDataList
- WinTypeDateEdit
- WinTypeDecimalEdit
- WinTypeDocView
- WinTypeEdit
- WinTypeFloatEdit
- WinTypeFontNameEdit
- WinTypeFontSizeEdit
- WinTypeGanttGraph
- WinTypeGroupBox
- WinTypeHyperLink
- WinTypeIntEdit
- WinTypeLabel
- WinTypeMenuButton
- WinTypeMetaPicture
- WinTypePicture
- WinTypeProgress
- WinTypeRadioButton
- WinTypeRecList
- WinTypeRTFEdit
- WinTypeSlider
- WinTypeStoList
- WinTypeTextEdit
- WinTypeTimeEdit
- WinTypeTreeView
- WinTypeWebNavigator

Kontakt

Beispiele:

```
tTheme->wpThemeColBorderDisabled(_WinTypeButton) # _WinColRed;tColor # tTheme->wpThemeColBorderDi
```

Kontakt

ThemeColBorderError Rahmenfarbe

bei Status "Fehler" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Rahmenfarbe, wenn der Objektinhalt ungültig ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBigIntEdit
- WinTypeColorEdit
- WinTypeDateEdit
- WinTypeDecimalEdit
- WinTypeEdit
- WinTypeFloatEdit
- WinTypeFontNameEdit
- WinTypeFontSizeEdit
- WinTypeIntEdit
- WinTypeTextEdit
- WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColBorderError(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeColBorderE
```

Kontakt

ThemeColBorderFocussed Rahmenfarbe

bei Status "fokussiert" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Rahmenfarbe, wenn das jeweilige Oberflächenobjekt den Fokus besitzt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBigIntEdit
- WinTypeColorEdit
- WinTypeDateEdit
- WinTypeDecimalEdit
- WinTypeEdit
- WinTypeFloatEdit
- WinTypeFontNameEdit
- WinTypeFontSizeEdit
- WinTypeIntEdit
- WinTypeTextEdit
- WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColBorderFocussed(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeColBord
```

Kontakt

ThemeColBorderHdrBase —

Rahmenfarbe

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für die Umrandung des Bereichs in dem die Spaltenköpfe dargestellt werden. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeDataList
- WinTypeDataListPopup
- WinTypeRecList
- WinTypeRecListPopup
- WinTypeStoList
- WinTypeStoListPopup

Beispiele:

```
tTheme->wpThemeColBorderHdrBase(_WinTypeRecList) # _WinColRed;tColor # tTheme->wpThemeColBorderHd
```

Kontakt

ThemeColBorderHighlight



Rahmenfarbe bei Status "hervorgehoben"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Rahmenfarbe, wenn sich die Maus über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBigIntEdit
- WinTypeButton
- WinTypeCheckBox
- WinTypeColorButton
- WinTypeColorEdit
- WinTypeDateEdit
- WinTypeDecimalEdit
- WinTypeEdit
- WinTypeFloatEdit
- WinTypeFontNameEdit
- WinTypeFontSizeEdit
- WinTypeIntEdit
- WinTypeMenuButton
- WinTypeRadioButton
- WinTypeTextEdit
- WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColBorderHighlight(_WinTypeButton) # _WinColRed;tColor # tTheme->wpThemeColBorderH
```

Kontakt

ThemeColBorderHover Rahmenfarbe bei

Status "mouseover" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Rahmenfarbe, wenn sich die Maus über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeHyperLink

Beispiele:

```
tTheme->wpThemeColBorderHover(_WinTypeHyperLink) # _WinColRed;tColor # tTheme->wpThemeColBorderHo
```

Kontakt

ThemeColBorderInactive



Rahmenfarbe bei Status "inaktiv" Typ

int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Rahmenfarbe, wenn das Oberflächenobjekt nicht den Eingabefokus besitzt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeColumnCheckbox
- WinTypeColumnRadiobutton

Beispiele:

```
tTheme->wpThemeColBorderInactive(_WinTypeColumnCheckbox) # _WinColRed;tColor # tTheme->wpThemeCol
```

Kontakt

ThemeColBorderIvl

Rahmenfarbe für Intervalle Typ

int

Siehe [Liste](#), [Objekte](#)

In dieser Eigenschaft können [Farbkonstanten](#) gesetzt werden.

Diese Eigenschaft definiert in einem [Theme](#)-Objekt die Rahmenfarbe für Intervalle in einem [GanttGraph](#). Das [Theme](#)-Objekt muss zuvor mit dem Befehl [WinThemeOpen\(\)](#) geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- [WinTypeGanttGraph](#)

Beispiele:

```
tTheme->wpThemeColBorderIvl(_WinTypeGanttGraph) # _WinColRed;tColor # tTheme->wpThemeColBorderIvl
```

Kontakt

ThemeColBorderNormal Rahmenfarbe

bei Status "normal" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Rahmenfarbe, wenn sich die Maus nicht über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeAnimation
- WinTypeAppFrame
- WinTypeBarcode
- WinTypeBigIntEdit
- WinTypeButton
- WinTypeCheckBox
- WinTypeCodeEdit
- WinTypeCodeEditList
- WinTypeColorButton
- WinTypeColorEdit
- WinTypeDataList
- WinTypeDateEdit
- WinTypeDecimalEdit
- WinTypeDocView
- WinTypeEdit
- WinTypeFloatEdit
- WinTypeFontNameEdit
- WinTypeFontSizeEdit
- WinTypeFrame
- WinTypeGanttGraph
- WinTypeGroupBox
- WinTypeGroupSplit
- WinTypeGroupTile
- WinTypeHyperLink
- WinTypeIcon
- WinTypeIntEdit
- WinTypeLabel
- WinTypeMdiFrame
- WinTypeMenuButton
- WinTypeMetaPicture
- WinTypeNotebookPage
- WinTypePicture
- WinTypePopupColor
- WinTypePopupList
- WinTypeProgress
- WinTypePrtPreviewDlg
- WinTypeRadioButton

Kontakt

- WinTypeRecList
- WinTypeRTFEdit
- WinTypeScrollbox
- WinTypeSlider
- WinTypeStoList
- WinTypeTextEdit
- WinTypeTimeEdit
- WinTypeTreeView
- WinTypeWebNavigator

Beispiele:

```
tTheme->wpThemeColBorderNormal(_WinTypeButton) # _WinColRed;tColor # tTheme->wpThemeColBorderNorm
```

Kontakt

ThemeColBorderPageNormal



Rahmenfarbe der Seiten bei Status "normal"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Rahmenfarbe der Seiten. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeDocView

Beispiele:

```
tTheme->wpThemeColBorderPageNormal(_WinTypeDocView) # _WinColRed;tColor # tTheme->wpThemeColBorde
```

Kontakt

ThemeColBorderPressed Rahmenfarbe

bei Status "gedrückt" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Rahmenfarbe, wenn die linke Maustaste auf dem jeweiligen Oberflächenobjekt gedrückt wird. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeButton
- WinTypeCheckBox
- WinTypeColorButton
- WinTypeMenuButton
- WinTypeRadioButton

Beispiele:

```
tTheme->wpThemeColBorderPressed(_WinTypeButton) # _WinColRed;tColor # tTheme->wpThemeColBorderPre
```

Kontakt

ThemeColBorderReadOnly Rahmenfarbe

bei Status "read-only" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Rahmenfarbe, wenn die Eigenschaft EditorReadOnly des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEdit

Beispiele:

```
tTheme->wpThemeColBorderReadOnly(_WinTypeCodeEdit) # _WinColRed;tColor # tTheme->wpThemeColBorder
```

Kontakt

ThemeColBorderSelectedActive Rahmenfarbe bei

Status "selektiert und aktiv" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Rahmenfarbe, wenn das Oberflächenobjekt selektiert ist und den Eingabefokus besitzt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeColumnCheckbox
- WinTypeColumnRadiobutton

Beispiele:

```
tTheme->wpThemeColBorderSelectedActive(_WinTypeColumnCheckbox) # _WinColRed;tColor # tTheme->wpTh
```

Kontakt

ThemeColBorderSelectedInactive Rahmenfarbe bei
Status "selektiert und inaktiv" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Rahmenfarbe, wenn das Oberflächenobjekt selektiert ist, jedoch nicht den Eingabefokus besitzt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeColumnCheckbox
- WinTypeColumnRadiobutton

Beispiele:

```
tTheme->wpThemeColBorderSelectedInactive(_WinTypeColumnCheckbox) # _WinColRed;tColor # tTheme->wp
```

Kontakt

ThemeColBorderVisited Rahmenfarbe

bei Status "besucht" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Rahmenfarbe, wenn sich die Maus nicht über dem jeweiligen Oberflächenobjekt befindet und die verknüpfte Seite besucht wurde. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeHyperLink

Beispiele:

```
tTheme->wpThemeColBorderVisited(_WinTypeHyperLink) # _WinColRed;tColor # tTheme->wpThemeColBorder
```

Kontakt

Check Color (Farb-Eigenschaften für Markierungsfarbe von Themes)

Eigenschaften

sortiert nach

Gruppen,

Siehe

Alphabetische

Liste aller

- ThemeColCheckActive
- ThemeColCheckDisabled
- ThemeColCheckHighlight
- ThemeColCheckInactive
- ThemeColCheckNormal
- ThemeColCheckPressed
- ThemeColCheckSelectedActive
- ThemeColCheckSelectedInactive

Kontakt

ThemeColCheckActive Markierungsfarbe

bei Status "aktiv" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für die Markierung, wenn das Oberflächenobjekt den Eingabefokus besitzt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeColumnCheckbox
- WinTypeColumnRadiobutton

Beispiele:

```
tTheme->wpThemeColCheckActive(_WinTypeColumnCheckbox) # _WinColRed;tColor # tTheme->wpThemeColChe
```

Kontakt

ThemeColCheckDisabled Markierungsfarbe

bei Status Disabled Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für die Markierung, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCheckBox
- WinTypeColumnCheckbox
- WinTypeColumnRadiobutton
- WinTypeRadioButton

Beispiele:

```
tTheme->wpThemeColCheckDisabled(_WinTypeCheckBox) # _WinColRed;tColor # tTheme->wpThemeColCheckDi
```

Kontakt

ThemeColCheckHighlight



Markierungsfarbe bei Status "hervorgehoben"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für die Markierung, wenn sich die Maus über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCheckBox
- WinTypePopupColor
- WinTypeRadioButton

Beispiele:

```
tTheme->wpThemeColCheckHighlight(_WinTypeCheckBox) # _WinColRed;tColor # tTheme->wpThemeColCheckH
```

Kontakt

ThemeColCheckInactive Markierungsfarbe

bei Status "inaktiv" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für die Markierung, wenn das Oberflächenobjekt nicht den Eingabefokus besitzt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeColumnCheckbox
- WinTypeColumnRadiobutton

Beispiele:

```
tTheme->wpThemeColCheckInactive(_WinTypeColumnCheckbox) # _WinColRed;tColor # tTheme->wpThemeColC
```

Kontakt

ThemeColCheckNormal Markierungsfarbe

bei Status "normal" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für die Markierung, wenn sich die Maus nicht über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCheckBox
- WinTypePopupColor
- WinTypeRadioButton

Beispiele:

```
tTheme->wpThemeColCheckNormal(_WinTypeCheckBox) # _WinColRed;tColor # tTheme->wpThemeColCheckNorm
```

Kontakt

ThemeColCheckPressed Markierungsfarbe bei

Status "gedrückt" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für die Markierung, wenn die linke Maustaste auf dem jeweiligen Oberflächenobjekt gedrückt wird. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCheckBox
- WinTypePopupColor
- WinTypeRadioButton

Beispiele:

```
tTheme->wpThemeColCheckPressed(_WinTypeCheckBox) # _WinColRed;tColor # tTheme->wpThemeColCheckPre
```

Kontakt

ThemeColCheckSelectedActive



Markierungsfarbe bei Status "selektiert und aktiv"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für die Markierung, wenn das Oberflächenobjekt selektiert ist und den Eingabefokus besitzt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeColumnCheckbox
- WinTypeColumnRadiobutton

Beispiele:

```
tTheme->wpThemeColCheckSelectedActive(_WinTypeColumnCheckbox) # _WinColRed;tColor # tTheme->wpThe
```

Kontakt

ThemeColCheckSelectedInactive



Markierungsfarbe bei Status "selektiert und inaktiv"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für die Markierung, wenn das Oberflächenobjekt selektiert ist, jedoch nicht den Eingabefokus besitzt. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeColumnCheckbox
- WinTypeColumnRadiobutton

Beispiele:

```
tTheme->wpThemeColCheckSelectedInactive(_WinTypeColumnCheckbox) # _WinColRed;tColor # tTheme->wpT
```

Kontakt

Separator Color (Farb-Eigenschaften für Separatorfarbe von Themes)

Eigenschaften

sortiert nach
Gruppen,

Siehe

Alphabetische
Liste aller

- [ThemeColSeparator](#)
- [ThemeColSeparatorClmBase](#)
- [ThemeColSeparatorClmFixedBase](#)
- [ThemeColSeparatorHdrBase](#)
- [ThemeColSeparatorLineBase](#)

Kontakt

ThemeColSeparator

Farbe des Separators

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe des Separators. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeDivider
- WinTypeGanttAxis
- WinTypeGanttGraph
- WinTypeMenuItem
- WinTypeStatusBarButton
- WinTypeToolbarButton

Beispiele:

```
tTheme->wpThemeColSeparator(_WinTypeStatusBarButton) # _WinColRed;tColor # tTheme->wpThemeColSepa
```

Kontakt

ThemeColSeparatorClmBase Basisfarbe

des Spalten-Separators Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für die vertikalen Separatoren zwischen den Spalten.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEditList
- WinTypeDataList
- WinTypeDataListPopup
- WinTypeRecList
- WinTypeRecListPopup
- WinTypeStoList
- WinTypeStoListPopup

Beispiele:

```
tTheme->wpThemeColSeparatorClmBase(_WinTypeRecList) # _WinColLightBlue;tColor # tTheme->wpThemeCo
```

Kontakt

ThemeColSeparatorClmFixedBase



Basisfarbe des Separators für die fixierte Spalte

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für den vertikalen Separator nach den fixierten Spalten.

Die fixierten Spalten werden über die Eigenschaft ClmFixed definiert.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEditList
- WinTypeDataList
- WinTypeDataListPopup
- WinTypeRecList
- WinTypeRecListPopup
- WinTypeStoList
- WinTypeStoListPopup

Beispiele:

```
tTheme->wpThemeColSeparatorClmFixedBase(_WinTypeRecList) # _WinColLightBlue;tColor # tTheme->wpTh
```

Kontakt

ThemeColSeparatorHdrBase —

Basisfarbe des Separators

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für Separatoren zwischen den Spaltenköpfen.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeDataList
- WinTypeDataListPopup
- WinTypeRecList
- WinTypeRecListPopup
- WinTypeStoList
- WinTypeStoListPopup

Beispiele:

```
tTheme->wpThemeColSeparatorHdrBase(_WinTypeRecList) # _WinColLightGray;tColor # tTheme->wpThemeCo
```

Kontakt

ThemeColSeparatorLineBase



Basisfarbe des Zeilen-Separators Typ

int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für die horizontalen Separatoren zwischen den Zeilen.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEditList
- WinTypeDataList
- WinTypeDataListPopup
- WinTypeRecList
- WinTypeRecListPopup
- WinTypeStoList
- WinTypeStoListPopup

Beispiele:

```
tTheme->wpThemeColSeparatorLineBase(_WinTypeRecList) # _WinColLightBlue;tColor # tTheme->wpThemeC
```

Kontakt

Splitter Color (Farb-Eigenschaften für Splitterfarbe von Themes)

Eigenschaften

sortiert nach

Gruppen,

Siehe

Alphabetische

Liste aller

- ThemeColFillSplitterDisabled
- ThemeColFillSplitterNormal
- ThemeColForeSplitterDisabled
- ThemeColForeSplitterNormal

Kontakt

ThemeColFillSplitterDisabled



Füllfarbe des Splitters bei Status Disabled

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe des Splitters, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEdit
- WinTypeDataList
- WinTypeDocView
- WinTypeGanttGraph
- WinTypeGroupSplit
- WinTypeRecList
- WinTypeStoList

Beispiele:

```
tTheme->wpThemeColFillSplitterDisabled(_WinTypeGroupSplit) # _WinColRed;tColor # tTheme->wpThemeC
```

Kontakt

ThemeColFillSplitterNormal



Füllfarbe des Splitters bei Status "normal"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe des Splitter, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt nicht gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEdit
- WinTypeDataList
- WinTypeDocView
- WinTypeGanttGraph
- WinTypeGroupSplit
- WinTypeRecList
- WinTypeStoList

Beispiele:

```
tTheme->wpThemeColFillSplitterNormal(_WinTypeGroupSplit) # _WinColRed;tColor # tTheme->wpThemeCol
```

Kontakt

ThemeColForeSplitterDisabled



Vordergrundfarbe des Splitters bei Status Disabled

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Vordergrundfarbe des

Splitters, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt gesetzt ist.

Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEdit
- WinTypeDataList
- WinTypeDocView
- WinTypeGroupSplit
- WinTypeRecList
- WinTypeStoList

Beispiele:

```
tTheme->wpThemeColForeSplitterDisabled(_WinTypeGroupSplit) # _WinColRed;tColor # tTheme->wpThemeC
```

Kontakt

ThemeColForeSplitterNormal



Vordergrundfarbe des Splitters bei Status "normal"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Vordergrundfarbe des Splitters, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt nicht gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEdit
- WinTypeDataList
- WinTypeDocView
- WinTypeGroupSplit
- WinTypeRecList
- WinTypeStoList

Beispiele:

```
tTheme->wpThemeColForeSplitterNormal(_WinTypeGroupSplit) # _WinColRed;tColor # tTheme->wpThemeCol
```

Kontakt

Grip Color (Farb-Eigenschaften für Ziehpunktfarbe von Themes)

Eigenschaften

sortiert nach

Gruppen,

Alphabetische

Liste aller

Siehe

- ThemeColGrip

Kontakt

ThemeColGrip



Farbe des Ziehpunktes

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe des Ziehpunktes. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeStatusBar
- WinTypeToolbar
- WinTypeToolbarMenu
- WinTypeWindowbar

Beispiele:

```
tTheme->wpThemeColGrip(_WinTypeStatusBar) # _WinColRed;tColor # tTheme->wpThemeColGrip(_WinTypeTo
```

Kontakt

Border Size (Größen-Eigenschaften für Rahmen von Themes)

Eigenschaften

sortiert nach

Gruppen,

Siehe

Alphabetische

Liste aller

- ThemeSizeBorderChecked
- ThemeSizeBorderDefault
- ThemeSizeBorderDisabled
- ThemeSizeBorderHighlight
- ThemeSizeBorderNormal
- ThemeSizeBorderPressed

ThemeSizeBorderChecked

Rahmengröße bei Status Toggled

Typ int

Siehe Liste, Objekte

Diese Eigenschaft definiert in einem Theme-Objekt die Rahmengröße, wenn die Eigenschaft Toggled des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeButton
- WinTypeColorButton
- WinTypeMenuButton

Beispiele:

```
tTheme->wpThemeSizeBorderChecked(_WinTypeButton) # 2;tSize # tTheme->wpThemeSizeBorderChecked(_Wi
```

ThemeSizeBorderDefault

Rahmengröße bei Status Default

Typ int

Siehe Liste, Objekte

Diese Eigenschaft definiert in einem Theme-Objekt die Rahmengröße, wenn die Eigenschaft Default des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeButton
- WinTypeColorButton
- WinTypeMenuButton

Beispiele:

```
tTheme->wpThemeSizeBorderDefault(_WinTypeButton) # 2;tSize # tTheme->wpThemeSizeBorderDefault(_Wi
```

ThemeSizeBorderDisabled

Rahmengröße bei Status Disabled

Typ int

Siehe Liste, Objekte

Diese Eigenschaft definiert in einem Theme-Objekt die Rahmengröße, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeButton
- WinTypeCheckBox
- WinTypeColorButton
- WinTypeMenuButton
- WinTypeRadioButton

Beispiele:

```
tTheme->wpThemeSizeBorderDisabled(_WinTypeButton) # 2;tSize # tTheme->wpThemeSizeBorderDisabled(_
```

Kontakt

ThemeSizeBorderHighlight

Rahmengröße bei Status "hervorgehoben"

Typ int

Siehe Liste, Objekte

Diese Eigenschaft definiert in einem Theme-Objekt die Rahmengröße, wenn sich die Maus über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeButton
- WinTypeCheckBox
- WinTypeColorButton
- WinTypeMenuButton
- WinTypeRadioButton

Beispiele:

```
tTheme->wpThemeSizeBorderHighlight(_WinTypeButton) # 2;tSize # tTheme->wpThemeSizeBorderHighlight
```

ThemeSizeBorderNormal

Rahmengröße bei Status "normal"

Typ int

Siehe Liste, Objekte

Diese Eigenschaft definiert in einem Theme-Objekt die Rahmengröße, wenn sich die Maus nicht über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeButton
- WinTypeCheckBox
- WinTypeColorButton
- WinTypeMenuButton
- WinTypeRadioButton

Beispiele:

```
tTheme->wpThemeSizeBorderNormal(_WinTypeButton) # 2;tSize # tTheme->wpThemeSizeBorderNormal(_WinT
```

Kontakt

ThemeSizeBorderPressed

Rahmengröße bei Status "gedrückt"

Typ **int**

Siehe [Liste, Objekte](#)

Diese Eigenschaft definiert in einem [Theme](#)-Objekt die Rahmengröße, wenn die linke Maustaste auf dem jeweiligen Oberflächenobjekt gedrückt wird. Das [Theme](#)-Objekt muss zuvor mit dem Befehl [WinThemeOpen\(\)](#) geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- [WinTypeButton](#)
- [WinTypeCheckBox](#)
- [WinTypeColorButton](#)
- [WinTypeMenuButton](#)
- [WinTypeRadioButton](#)

Beispiele:

```
tTheme->wpThemeSizeBorderPressed(_WinTypeButton) # 2;tSize # tTheme->wpThemeSizeBorderPressed(_Wi
```

Kontakt

Responsive Size (Größen-Eigenschaften von Themes)

Eigenschaften

sortiert nach

Gruppen,

Alphabetische

Liste aller

Siehe

- ThemeSizeFixed

ThemeSizeFixed

Feste Höhe eines Objektes

Typ int

Siehe Liste, Objekte

Diese Eigenschaft definiert in einem Theme-Objekt die feste Höhe (falls > 0) des jeweiligen Oberflächenobjektes. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeAppFrame
- WinTypeFrame
- WinTypeMdiFrame
- WinTypeNotebook

 Bei den Typen WinTypeAppFrame, WinTypeFrame und WinTypeMdiFrame beträgt die kleinstmögliche Höhe 16 Pixel. Für diese Objekte betrifft die Eigenschaft die Höhe der Titelzeile.

Bei WinTypeNotebook wird mit der Eigenschaft die Höhe der Registerreiter definiert.

 Die Schriftgröße wird in den Objekten durch setzen dieser Eigenschaft auf einen Wert ungleich 0 durch die verfügbare Größe und nicht mehr durch die jeweilige Schriftart bestimmt.

Beispiele:

```
tTheme->wpThemeSizeFixed(_WinTypeFrame) # 20;tSize # tTheme->wpThemeSizeFixed(_WinTypeNotebook);
```

Kontakt

Spin Color (Farb-Eigenschaften für Spin-Bereich von Themes)

Eigenschaften

sortiert nach

Gruppen,

Siehe

Alphabetische

Liste aller

- [ThemeColFillSpinArrowDisabled](#)
- [ThemeColFillSpinArrowError](#)
- [ThemeColFillSpinArrowHighlight](#)
- [ThemeColFillSpinArrowNormal](#)
- [ThemeColFillSpinArrowPressed](#)
- [ThemeColFillSpinDisabled](#)
- [ThemeColFillSpinError](#)
- [ThemeColFillSpinHighlight](#)
- [ThemeColFillSpinNormal](#)
- [ThemeColFillSpinPressed](#)

Kontakt

ThemeColFillSpinArrowDisabled



Pfeilfarbe im Spin-Bereich bei Status Disabled

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Pfeilfarbe im Spin-Bereich, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBigIntEdit
- WinTypeIntEdit
- WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColFillSpinArrowDisabled(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpTheme
```

Kontakt

ThemeColFillSpinArrowError



Pfeilfarbe im Spin-Bereich bei Status "Fehler"

Typ [int](#)

Siehe [Liste](#), [Objekte](#)

In dieser Eigenschaft können [Farbkonstanten](#) gesetzt werden.

Diese Eigenschaft definiert in einem [Theme](#)-Objekt die Pfeilfarbe im Spin-Bereich, wenn der Objektinhalt ungültig ist. Das [Theme](#)-Objekt muss zuvor mit dem Befehl [WinThemeOpen\(\)](#) geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- [WinTypeBigIntEdit](#)
- [WinTypeIntEdit](#)
- [WinTypeTimeEdit](#)

Beispiele:

```
tTheme->wpThemeColFillSpinArrowError(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeCol
```

Kontakt

ThemeColFillSpinArrowHighlight



Pfeilfarbe im Spin-Bereich bei Status "hervorgehoben"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Pfeilfarbe im Spin-Bereich, wenn sich die Maus über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBigIntEdit
- WinTypeIntEdit
- WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColFillSpinArrowHighlight(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThem
```

Kontakt

ThemeColFillSpinArrowNormal



Pfeilfarbe im Spin-Bereich bei Status "normal"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Pfeilfarbe im Spin-Bereich, wenn sich die Maus nicht über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBigIntEdit
- WinTypeIntEdit
- WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColFillSpinArrowNormal(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeCo
```

Kontakt

ThemeColFillSpinArrowPressed



Pfeilfarbe im Spin-Bereich bei Status "gedrückt"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Pfeilfarbe im Spin-Bereich, wenn der Pfeil des InputControls gedrückt wurde. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBigIntEdit
- WinTypeIntEdit
- WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColFillSpinArrowPressed(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeC
```

Kontakt

ThemeColFillSpinDisabled



Füllfarbe des Spin-Bereiches bei Status Disabled

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe des Spin-Bereiches, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBigIntEdit
- WinTypeIntEdit
- WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColFillSpinDisabled(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeColFi
```

Kontakt

ThemeColFillSpinError



Füllfarbe des Spin-Bereiches bei Status "Fehler"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe des Spin-Bereiches, wenn der Objektinhalt ungültig ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBigIntEdit
- WinTypeIntEdit
- WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColFillSpinError(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeColFillS
```

Kontakt

ThemeColFillSpinHighlight



Füllfarbe des Spin-Bereiches bei Status "hervorgehoben"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe des Spin-Bereiches, wenn sich die Maus über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBigIntEdit
- WinTypeIntEdit
- WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColFillSpinHighlight(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeColF
```

Kontakt

ThemeColFillSpinNormal



Füllfarbe des Spin-Bereiches bei Status "normal"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe des Spin-Bereiches, wenn sich die Maus nicht über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBigIntEdit
- WinTypeIntEdit
- WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColFillSpinNormal(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeColFill
```

Kontakt

ThemeColFillSpinPressed



Füllfarbe des Spin-Bereiches bei Status "gedrückt"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe des Spin-Bereiches, wenn ein Pfeil des InputControls gedrückt wurde. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBigIntEdit
- WinTypeIntEdit
- WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColFillSpinPressed(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeColFil
```

Kontakt

Popup button (Farb-Eigenschaften für Popup-Button von Themes)

Eigenschaften

sortiert nach

Gruppen,

Siehe

Alphabetische

Liste aller

- ThemeColArrowDropDisabled
- ThemeColArrowDropError
- ThemeColArrowDropHighlight
- ThemeColArrowDropNormal
- ThemeColArrowDropPressed
- ThemeColFillDropDisabled
- ThemeColFillDropError
- ThemeColFillDropHighlight
- ThemeColFillDropNormal
- ThemeColFillDropPressed

Kontakt

ThemeColArrowDropDisabled 

Pfeilfarbe bei Status Disabled

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für das Pfeilsymbol in der Popup-Schaltfläche, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBigIntEdit
- WinTypeColorEdit
- WinTypeDateEdit
- WinTypeDecimalEdit
- WinTypeEdit
- WinTypeFloatEdit
- WinTypeFontNameEdit
- WinTypeFontSizeEdit
- WinTypeIntEdit
- WinTypeTimeEdit
- WinTypeToolbarButton

Beispiele:

```
tTheme->wpThemeColArrowDrawDisabled(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeColA
```

Kontakt

ThemeColArrowDropError



Pfeilfarbe bei Status "Fehler" Typ

int

Siehe [Liste, Objekte](#)

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für das Pfeilsymbol in der Popup-Schaltfläche, wenn der Objektinhalt ungültig ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- [WinTypeBarcode](#)
- [WinTypeBigintEdit](#)
- [WinTypeColorEdit](#)
- [WinTypeDateEdit](#)
- [WinTypeDecimalEdit](#)
- [WinTypeEdit](#)
- [WinTypeFloatEdit](#)
- [WinTypeFontNameEdit](#)
- [WinTypeFontSizeEdit](#)
- [WinTypeIntEdit](#)
- [WinTypeTimeEdit](#)

Beispiele:

```
tTheme->wpThemeColArrowDropError(_WinTypeBarcode) # _WinColRed;tColor # tTheme->wpThemeColArrowDr
```

Kontakt

ThemeColArrowDropHighlight Pfeilfarbe

bei Status "hervorgehoben" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für das Pfeilsymbol in der Popup-Schaltfläche, wenn sich die Maus über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBigIntEdit
- WinTypeColorEdit
- WinTypeDateEdit
- WinTypeDecimalEdit
- WinTypeEdit
- WinTypeFloatEdit
- WinTypeFontNameEdit
- WinTypeFontSizeEdit
- WinTypeIntEdit
- WinTypeTimeEdit
- WinTypeToolbarButton

Beispiele:

```
tTheme->wpThemeColArrowDropHighlight(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeCol
```

Kontakt

ThemeColArrowDropNormal

Pfeilfarbe bei Status "normal" Typ

int

Siehe [Liste, Objekte](#)

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für das Pfeilsymbol in der Popup-Schaltfläche, wenn weder die Eigenschaften Toggled, noch Disabled gesetzt sind und sich die Maus nicht über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- [WinTypeBigIntEdit](#)
- [WinTypeColorEdit](#)
- [WinTypeDateEdit](#)
- [WinTypeDecimalEdit](#)
- [WinTypeEdit](#)
- [WinTypeFloatEdit](#)
- [WinTypeFontNameEdit](#)
- [WinTypeFontSizeEdit](#)
- [WinTypeIntEdit](#)
- [WinTypeTimeEdit](#)
- [WinTypeToolbarButton](#)

Beispiele:

```
tTheme->wpThemeColArrowDropNormal(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeColArr
```

Kontakt

ThemeColArrowDropPressed 

Pfeilfarbe bei Status "gedrückt" Typ

int

Siehe [Liste, Objekte](#)

In dieser Eigenschaft können [Farbkonstanten](#) gesetzt werden.

Diese Eigenschaft definiert in einem [Theme](#)-Objekt die Farbe für das Pfeilsymbol in der Popup-Schaltfläche, wenn die linke Maustaste auf dem jeweiligen Oberflächenobjekt gedrückt wird. Das [Theme](#)-Objekt muss zuvor mit dem Befehl [WinThemeOpen\(\)](#) geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- [WinTypeBigIntEdit](#)
- [WinTypeColorEdit](#)
- [WinTypeDateEdit](#)
- [WinTypeDecimalEdit](#)
- [WinTypeEdit](#)
- [WinTypeFloatEdit](#)
- [WinTypeFontNameEdit](#)
- [WinTypeFontSizeEdit](#)
- [WinTypeIntEdit](#)
- [WinTypeTimeEdit](#)
- [WinTypeToolbarButton](#)

Beispiele:

```
tTheme->wpThemeColArrowDropPressed(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeColAr
```

Kontakt

ThemeColFillDropDisabled



Füllfarbe bei Status Disabled Typ

int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe der Popup-Schaltfläche, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBigIntEdit
- WinTypeColorEdit
- WinTypeDateEdit
- WinTypeDecimalEdit
- WinTypeEdit
- WinTypeFloatEdit
- WinTypeFontNameEdit
- WinTypeFontSizeEdit
- WinTypeIntEdit
- WinTypeTimeEdit
- WinTypeToolbarButton

Beispiele:

```
tTheme->wpThemeColFillDropDisabled(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeColFi
```

Kontakt

ThemeColFillDropError



Füllfarbe bei Status "Fehler" Typ

int

Siehe [Liste, Objekte](#)

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe der Popup-Schaltfläche, wenn der Objektinhalt ungültig ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- [WinTypeBigintEdit](#)
- [WinTypeColorEdit](#)
- [WinTypeDateEdit](#)
- [WinTypeDecimalEdit](#)
- [WinTypeEdit](#)
- [WinTypeFloatEdit](#)
- [WinTypeFontNameEdit](#)
- [WinTypeFontSizeEdit](#)
- [WinTypeIntEdit](#)
- [WinTypeTimeEdit](#)

Beispiele:

```
tTheme->wpThemeColFillDropError(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeColFillD
```

Kontakt

ThemeColFillDropHighlight Füllfarbe bei

Status "hervorgehoben" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe der Popup-Schaltfläche, wenn sich die Maus über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBigIntEdit
- WinTypeColorEdit
- WinTypeDateEdit
- WinTypeDecimalEdit
- WinTypeEdit
- WinTypeFloatEdit
- WinTypeFontNameEdit
- WinTypeFontSizeEdit
- WinTypeIntEdit
- WinTypeTimeEdit
- WinTypeToolbarButton

Beispiele:

```
tTheme->wpThemeColFillDropHighlight(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeColF
```

Kontakt

ThemeColFillDropNormal



Füllfarbe bei Status "normal" Typ

int

Siehe [Liste, Objekte](#)

In dieser Eigenschaft können [Farbkonstanten](#) gesetzt werden.

Diese Eigenschaft definiert in einem [Theme](#)-Objekt die Füllfarbe der Popup-Schaltfläche, wenn sich die Maus nicht über dem jeweiligen Oberflächenobjekt befindet. Das [Theme](#)-Objekt muss zuvor mit dem Befehl [WinThemeOpen\(\)](#) geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- [WinTypeBigIntEdit](#)
- [WinTypeColorEdit](#)
- [WinTypeDateEdit](#)
- [WinTypeDecimalEdit](#)
- [WinTypeEdit](#)
- [WinTypeFloatEdit](#)
- [WinTypeFontNameEdit](#)
- [WinTypeFontSizeEdit](#)
- [WinTypeIntEdit](#)
- [WinTypeTimeEdit](#)
- [WinTypeToolbarButton](#)

Beispiele:

```
tTheme->wpThemeColFillDropNormal(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeColFill
```

Kontakt

ThemeColFillDropPressed Füllfarbe 

bei Status "gedrückt" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe der Popup-Schaltfläche, wenn die linke Maustaste auf dem jeweiligen Oberflächenobjekt gedrückt wird. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeBigIntEdit
- WinTypeColorEdit
- WinTypeDateEdit
- WinTypeDecimalEdit
- WinTypeEdit
- WinTypeFloatEdit
- WinTypeFontNameEdit
- WinTypeFontSizeEdit
- WinTypeIntEdit
- WinTypeTimeEdit
- WinTypeToolbarButton

Beispiele:

```
tTheme->wpThemeColFillDropPressed(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeColFil
```

Kontakt

Scroll button (Farb-Eigenschaften für Scroll-Button von Themes)

Eigenschaften

sortiert nach

Gruppen,

Siehe

Alphabetische

Liste aller

- ThemeColArrowScrollHighlight
- ThemeColArrowScrollNormal
- ThemeColArrowScrollPressed
- ThemeColFillScrollHighlight
- ThemeColFillScrollNormal
- ThemeColFillScrollPressed

Kontakt

ThemeColArrowScrollHighlight Pfeilfarbe

bei Status "hervorgehoben" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für das Pfeilsymbol in der Scroll-Schaltfläche, wenn sich die Maus über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeNotebook

Beispiele:

```
tTheme->wpThemeColArrowScrollHighlight(_WinTypeNotebook) # _WinColRed;tColor # tTheme->wpThemeCol
```

Kontakt

ThemeColArrowScrollNormal 

Pfeilfarbe bei Status "normal"

Typ [int](#)

Siehe [Liste](#), [Objekte](#)

In dieser Eigenschaft können [Farbkonstanten](#) gesetzt werden.

Diese Eigenschaft definiert in einem [Theme](#)-Objekt die Farbe für das Pfeilsymbol in der Scroll-Schaltfläche, wenn sich die Maus nicht über dem jeweiligen Oberflächenobjekt befindet. Das [Theme](#)-Objekt muss zuvor mit dem Befehl [WinThemeOpen\(\)](#) geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- [WinTypeNotebook](#)

Beispiele:

```
tTheme->wpThemeColArrowScrollNormal(_WinTypeNotebook) # _WinColRed;tColor # tTheme->wpThemeColArr
```

Kontakt

ThemeColArrowScrollPressed

Pfeilfarbe bei Status "gedrückt" Typ

int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe für das Pfeilsymbol in der Scroll-Schaltfläche, wenn die linke Maustaste auf dem jeweiligen Oberflächenobjekt gedrückt wird. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeNotebook

Beispiele:

```
tTheme->wpThemeColArrowScrollPressed(_WinTypeNotebook) # _WinColRed;tColor # tTheme->wpThemeColAr
```

Kontakt

ThemeColFillScrollHighlight Füllfarbe bei

Status "hervorgehoben" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe der Scroll-Schaltfläche, wenn sich die Maus über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeNotebook

Beispiele:

```
tTheme->wpThemeColFillScrollHighlight(_WinTypeNotebook) # _WinColRed;tColor # tTheme->wpThemeColF
```

Kontakt

ThemeColFillScrollNormal



Füllfarbe bei Status "normal" Typ

int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe der Scroll-Schaltfläche, wenn sich die Maus nicht über dem jeweiligen Oberflächenobjekt befindet. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeNotebook

Beispiele:

```
tTheme->wpThemeColFillScrollNormal(_WinTypeNotebook) # _WinColRed;tColor # tTheme->wpThemeColFill
```

Kontakt

ThemeColFillPressed



Füllfarbe bei Status "gedrückt" Typ

int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe der Scroll-Schaltfläche, wenn die linke Maustaste auf dem jeweiligen Oberflächenobjekt gedrückt wird. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeNotebook

Beispiele:

```
tTheme->wpThemeColFillPressed(_WinTypeNotebook) # _WinColRed;tColor # tTheme->wpThemeColFil
```

Kontakt

Editor Color (Farb-Eigenschaften für CodeEdit-Objekt von Themes)

Eigenschaften

sortiert nach

Gruppen,

Siehe

Alphabetische

Liste aller

- [ThemeColEditorAlpha](#)
- [ThemeColEditorCKeywords](#)
- [ThemeColEditorComment](#)
- [ThemeColEditorConstants](#)
- [ThemeColEditorCurrentLine](#)
- [ThemeColEditorCustomHighlight](#)
- [ThemeColEditorDataTypes](#)
- [ThemeColEditorFunctions](#)
- [ThemeColEditorKeywords](#)
- [ThemeColEditorNumber](#)
- [ThemeColEditorOccurrences](#)
- [ThemeColEditorPreProcessor](#)
- [ThemeColEditorSelection](#)
- [ThemeColEditorText](#)

Kontakt

ThemeColEditorAlpha



Textfarbe für Zeichenketten im CodeEdit-Objekt

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe für Zeichenketten in CodeEdit-Objekten. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEdit

Beispiele:

```
tTheme->wpThemeColEditorAlpha(_WinTypeCodeEdit) # _WinColCyan;tColor # tTheme->wpThemeColEditorAl
```

Kontakt

ThemeColEditorCKeywords



Textfarbe für benutzerdefinierte Schlüsselwörter im CodeEdit-Objekt

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe für benutzerdefinierte Schlüsselwörter in CodeEdit-Objekten. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEdit

Beispiele:

```
tTheme->wpThemeColEditorCKeywords(_WinTypeCodeEdit) # _WinColBlue;tColor # tTheme->wpThemeColEdit
```

Kontakt

ThemeColEditorComment



Textfarbe für Kommentare im CodeEdit-Objekt

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe für Kommentare in CodeEdit-Objekten. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEdit

Beispiele:

```
tTheme->wpThemeColEditorComment(_WinTypeCodeEdit) # _WinColGreen;tColor # tTheme->wpThemeColEdito
```

Kontakt

ThemeColEditorConstants



Textfarbe für Konstanten im CodeEdit-Objekt

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe für Konstanten in CodeEdit-Objekten. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEdit

Beispiele:

```
tTheme->wpThemeColEditorConstants(_WinTypeCodeEdit) # _WinColBlue;tColor # tTheme->wpThemeColEdit
```

Kontakt

ThemeColEditorCurrentLine



Hintergrundfarbe für aktuelle Zeile im CodeEdit-Objekt

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Hintergrundfarbe für die aktuelle Zeile in CodeEdit-Objekten. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEdit



Die Hintergrundfarbe der aktuellen Zeile wird mit 30% Deckkraft gezeichnet. Für einen dunklen Hintergrund (ThemeColFillNormal / ThemeColFillDisabled / ThemeColFillReadOnly) empfiehlt sich WinColWhite, andernfalls WinColBlack.

Beispiele:

```
tTheme->wpThemeColEditorCurrentLine(_WinTypeCodeEdit) # _WinColBlack;tColor # tTheme->wpThemeColE
```

Kontakt



ThemeColEditorCustomHighlight

Hintergrundfarbe für benutzerdefinierte Hervorhebung im CodeEdit-Objekt Typ int

Siehe Liste, Objekte, WinEditorHighlight()

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Hintergrundfarbe für benutzerdefinierte Hervorhebungen in CodeEdit-Objekten. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEdit



Die Hintergrundfarbe für benutzerdefinierte Hervorhebungen wird mit 30% Deckkraft gezeichnet.

Beispiele:

```
tTheme->wpThemeColEditorCustomHighlight(_WinTypeCodeEdit) # _WinColBlack;tColor # tTheme->wpTheme
```

Kontakt

ThemeColEditorDataTypes



Textfarbe für Datentypen im CodeEdit-Objekt

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe für Datentypen in CodeEdit-Objekten. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEdit

Beispiele:

```
tTheme->wpThemeColEditorDataTypes(_WinTypeCodeEdit) # _WinColBlue;tColor # tTheme->wpThemeColEdit
```

Kontakt

ThemeColEditorFunctions



Textfarbe für Funktionen im CodeEdit-Objekt

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe für Funktionen in CodeEdit-Objekten. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEdit

Beispiele:

```
tTheme->wpThemeColEditorFunctions(_WinTypeCodeEdit) # _WinColBlue;tColor # tTheme->wpThemeColEdit
```

Kontakt

ThemeColEditorKeywords



Textfarbe für Schlüsselwörter im CodeEdit-Objekt

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe für Schlüsselwörter in CodeEdit-Objekten. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEdit

Beispiele:

```
tTheme->wpThemeColEditorKeywords(_WinTypeCodeEdit) # _WinColBlue;tColor # tTheme->wpThemeColEdito
```

Kontakt

ThemeColEditorNumber



Textfarbe für Zahlen im CodeEdit-Objekt

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe für Zahlen in CodeEdit-Objekten.

Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEdit

Beispiele:

```
tTheme->wpThemeColEditorNumber(_WinTypeCodeEdit) # _WinColLightRed;tColor # tTheme->wpThemeColEdi
```

Kontakt

ThemeColEditorOccurrences



Hintergrundfarbe für Suchtreffer im CodeEdit-Objekt

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Hintergrundfarbe für Suchtreffer in CodeEdit-Objekten. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEdit



Die Hintergrundfarbe des aktuellen Suchtreffers wird von der ausgewählten Farbe um 25% verdunkelt bzw. erhellt, je nach Helligkeit der Farbe. Die Hintergrundfarbe für weitere Suchtreffer wird mit 70% Deckkraft gezeichnet.

Beispiele:

```
tTheme->wpThemeColEditorOccurrences(_WinTypeCodeEdit) # _WinColLightCyan;tColor # tTheme->wpTheme
```

Kontakt

ThemeColEditorPreProcessor



Textfarbe für Präprozessoranweisungen im CodeEdit-Objekt

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe für Präprozessoranweisungen in CodeEdit-Objekten. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEdit

Beispiele:

```
tTheme->wpThemeColEditorPreProcessor(_WinTypeCodeEdit) # _WinColBlue;tColor # tTheme->wpThemeColE
```

Kontakt

ThemeColEditorSelection



Hintergrundfarbe für selektierte Bereiche im CodeEdit-Objekt

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Hintergrundfarbe für selektierte Bereiche in CodeEdit-Objekten. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEdit

Beispiele:

```
tTheme->wpThemeColEditorSelection(_WinTypeCodeEdit) # _WinColLightGray;tColor # tTheme->wpThemeCo
```

Kontakt

ThemeColEditText



Textfarbe für normalen Text im CodeEdit-Objekt

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Textfarbe für normalen Text in CodeEdit-Objekten. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeCodeEdit

Beispiele:

```
tTheme->wpThemeColEditText(_WinTypeCodeEdit) # _WinColBlack;tColor # tTheme->wpThemeColEditText
```

Kontakt

Image Color (Farb-Eigenschaften für Bilder von Themes)

Eigenschaften

sortiert nach

Gruppen,

Siehe

Alphabetische

Liste aller

- ThemeColImageNormal

Kontakt

ThemeColImageNormal Bildfarbe

bei Status "normal" Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Füllfarbe des Bildes im normalen Status, wenn TileTheme auf WinTileThemeFlatMono gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeIcon
- WinTypeTreeNode

Beispiele:

```
tTheme->wpThemeColImageNormal(_WinTypeIcon) # _WinColWhite;tColor # tTheme->wpThemeColImageNormal
```

Kontakt

Tick Color (Farb-Eigenschaften für Markierungsfarbe von Themes)

Eigenschaften

sortiert nach

Gruppen,

Siehe

Alphabetische

Liste aller

- ThemeColTickDisabled

- ThemeColTickNormal

Kontakt

ThemeColTickDisabled



Farbe der Markierungen bei Status Disabled

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe der Markierungen, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeSlider

Beispiele:

```
tTheme->wpThemeColTickDisabled(_WinTypeSlider) # _WinColRed;tColor # tTheme->wpThemeColTickDisabl
```

Kontakt

ThemeColTickNormal



Farbe der Markierungen bei Status "normal"

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Diese Eigenschaft definiert in einem Theme-Objekt die Farbe der Markierungen, wenn die Eigenschaft Disabled des jeweiligen Oberflächenobjekt nicht gesetzt ist. Das Theme-Objekt muss zuvor mit dem Befehl WinThemeOpen() geöffnet worden sein.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeSlider

Beispiele:

```
tTheme->wpThemeColTickNormal(_WinTypeSlider) # _WinColRed;tColor # tTheme->wpThemeColTickNormal(_
```

Kontakt

Ermitteln und Setzen von Eigenschaften

Aufteilung der Eigenschaften in Gruppen

WinPropGet(),
WinPropSet(),
PrtPropGet(),
Siehe PrtPropSet(),
SysPropGet(),
SysPropSet(),
UrmPropGet(),
UrmPropSet()

Die Eigenschaften sind in vier Gruppen aufgeteilt:

- Eigenschaften von Oberflächen-Objekten
- Eigenschaften von Druck-Objekten
- Eigenschaften von System-Objekten
- Eigenschaften des Benutzersystems

Jede Gruppe besitzt ihre eigenen Befehle Win-, Prt-, Sys- und UrmPropGet/Set mit denen die Eigenschaft ausgelesen bzw. gesetzt werden kann. Das Ansprechen von Eigenschaften in Prozeduren funktioniert prinzipiell wie zur Entwurfszeit im Designer. Ist der Name der Eigenschaft bekannt, kann durch eine Objektreferenz und dem Namen der Eigenschaft, die Eigenschaft bei dem Objekt verändern oder auslesen.

Der Name der Eigenschaft wird für den Zugriff je nach Gruppe um _WinProp, _PrtProp oder _SysProp erweitert. Die Eigenschaften von Benutzer-Objekten können nur über die _UrmProp-Konstanten angesprochen werden.

Beispiele:

Setzen der Eigenschaft Caption bei einem Oberflächen-Objekt:

```
tErg # tHdlFrame->WinPropSet(_WinPropCaption, 'Suchen');
```

Setzen der Eigenschaft Caption bei einem Druck-Objekt:

```
tErg # tObjPrtText->PrtPropSet(_PrtPropCaption, 'Suchen');
```

Auslesen des temporären Pfades des Betriebssystems aus dem Sys-Objekt:

```
tErg # _Sys->SysPropGet(_SysPropTempPath, tTemp);
```

Auslesen des Namens eines Benutzers:

```
tErg # tHdlUser->UrmPropGet(_UrmPropLastname, tRealLastName);
```

In den Beispielen wird der logische Rückgabewert der Befehle nicht ausgewertet. Über diesen Rückgabewert kann festgestellt werden, ob das Lesen oder Setzen einer Eigenschaft erfolgreich war.

Zum Setzen und Lesen von Eigenschaften gibt es noch eine kürzere Schreibweise. In dieser Schreibweise wird vor dem Namen der Eigenschaft ein Kürzel wp, pp

Kontakt

oder sp gesetzt, um zu kennzeichnen, ob es sich um eine Oberflächen-, Druck-, oder System-Eigenschaft handelt. Ein Kürzel für Eigenschaften des Benutzersystems existiert nicht. Die oben angeführten Beispiele können auch wie folgt geschrieben werden:

```
tHd1Frame->wpCaption # 'Suchen'; tObjPrtText->ppCaption # 'Suchen'; tTemp # _Sys->spTempPath;
```

Diese Anweisungen geben keinen Rückgabewert zurück, der den Erfolg der Operation darstellt. Kann eine Eigenschaft nicht gesetzt oder gelesen werden, erfolgt ein entsprechender Laufzeitfehler.

Die Befehle sollten immer dann verwendet werden, wenn das Vorhandensein einer Eigenschaft zu einem Objekt nicht sichergestellt ist. Zum Beispiel in einer Funktion, die ein Objekt übergeben bekommt. In den anderen Fällen kann die kurze Schreibweise verwendet werden.

Kontakt

Liste aller Eigenschaften von Oberflächen- und Druck-Objekten

In dieser Liste sind alle Eigenschaften von Oberflächen-und Druck-Objekten aufgeführt.

[Liste](#)
[sortiert](#)

Siehe [nach](#)
[Gruppen](#)

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [Z](#)

- [AlignGrouping](#)
- [AlignGroupingBottomOrder](#)
- [AlignGroupingLeftOrder](#)
- [AlignGroupingRightOrder](#)
- [AlignGroupingTopOrder](#)
- [AlignHeight](#)
- [AlignInputCtrl](#)
- [AlignMarginBottom](#)
- [AlignMarginLeft](#)
- [AlignMarginRight](#)
- [AlignMarginTop](#)
- [AlignTab](#)
- [AlignWidth](#)
- [Area](#)
- [AreaBottom](#)
- [AreaHeight](#)
- [AreaHeightCells](#)
- [AreaHeightMax](#)
- [AreaHeightMin](#)
- [AreaId](#)
- [AreaLeft](#)
- [AreaMarginBottom](#)
- [AreaMarginLeft](#)
- [AreaMarginRight](#)
- [AreaMarginTop](#)
- [AreaRight](#)
- [AreaScaling](#)
- [AreaScalingFlags](#)
- [AreaScalingPageNo](#)
- [AreaTop](#)
- [AreaWidth](#)
- [AreaWidthCells](#)
- [AreaWidthMax](#)
- [AreaWidthMin](#)
- [ArgInt](#)
- [ArgInt64](#)
- [ArgIntCount](#)
- [AutoSize](#)
- [AutoUpdate](#)
- [AutoWrap](#)

- [BinSource](#)
- [BinSourceName](#)
- [BorderColFg](#)
- [BorderType](#)
- [BorderWidth](#)
- [Bound](#)
- [BoundAdd](#)
- [BoundMax](#)
- [BrushType](#)

- [CalculatorSize](#)
- [Caption](#)
- [CaptionBigInt](#)
- [CaptionColAuto](#)
- [CaptionColor](#)
- [CaptionDate](#)
- [CaptionDecimal](#)
- [CaptionFloat](#)
- [CaptionInt](#)
- [CaptionTime](#)
- [CellActiveHorz](#)
- [CellActiveVert](#)
- [CellCountHorz](#)
- [CellCountVert](#)
- [CellOfsHorz](#)
- [CellOfsVert](#)
- [CellSizeHorz](#)
- [CellSizeMode](#)
- [CellSizeVert](#)
- [Changed](#)
- [ChangedChild](#)
- [ChangedTrigger](#)
- [CheckState](#)
- [ClmAlign](#)
- [ClmColBkg](#)
- [ClmColFg](#)
- [ClmColFocusBkg](#)
- [ClmColFocusFg](#)
- [ClmColFocusOffBkg](#)
- [ClmColFocusOffFg](#)
- [ClmFixed](#)
- [ClmOrder](#)
- [ClmPopup](#)
- [ClmSortFlags](#)
- [ClmSortImage](#)
- [ClmStretch](#)
- [ClmStretchWidth](#)
- [ClmType](#)
- [ClmTypeImage](#)
- [ClmTypeSort](#)

- ClmWidth
- ColAuto
- ColBkg
- ColBkgApp
- ColBkgBase
- ColBkgOffPrint
- ColBkgPage
- ColBkgTabArea
- ColBkgTabSel
- ColBoundary
- ColCellActive
- ColCellActiveHorz
- ColDateFg
- ColDateMonth
- ColDateTitleBkg
- ColDateTitleFg
- ColDateTrailing
- ColDisabledBkg
- ColDisabledFg
- ColErrRangeBkg
- ColErrRangeFg
- ColFg
- ColFgTabSel
- ColFillEven
- ColFillOdd
- ColFocusBkg
- ColFocusFg
- ColFocusOffBkg
- ColFocusOffFg
- ColGrid
- ColHover
- ColIvlArrows
- ColIvlBorder
- Collate
- ColLink
- ColSeparator
- ColVisited
- ComboStyle
- Comment
- Compression
- ConnectionString
- ContentHeight
- ContentHeightMax
- ContentLeft
- ContentSource
- ContentTop
- ContentType
- ContentWidth
- Copies
- CorrectionLevel
- Created

- CreatedUser
- CroDevToolsSide
- CroDevToolsSize
- CryptKey
- Current
- CurrentInt
- Custom

- Data
- DataOwner
- DateWindow
- DbaName
- DbaUserID
- DbaUserName
- DbFieldName
- DbFileNo
- DbFilter
- DbKeyNo
- DbKeyReverseSort
- DbLinkFileNo
- DbRecBuf
- DbRecID
- DbRecID64
- DbSelection
- DbTextBuf
- DbVar
- DecEditFlags
- Default
- DefaultDate
- Design
- DialogBoxLeft
- DialogBoxTop
- Dictionaries
- Disabled
- DockSide
- DragSource
- DrawBkgAlways
- DropFiles
- DropPlace
- Duplex

- EditKeyEnd
- EditKeyStart
- EditMask
- EditMouseStart
- EditorAclFlags
- EditorAllowControlChars
- EditorBraceHighlight
- EditorFixedHighlighting

- [EditorFold](#)
- [EditorReadOnly](#)
- [EditorShowEOLCharacters](#)
- [EditorShowFlags](#)
- [EditorShowLineLimit](#)
- [EditorShowLineNumbers](#)
- [EditorShowWSCharacters](#)
- [EditorTabSize](#)
- [EditorTextModified](#)
- [EditorTextType](#)
- [EmfFileName](#)
- [EmulateKeys](#)
- [Erg](#)
- [ErrCode](#)
- [ErrLine](#)
- [Error](#)
- [ErrProc](#)
- [ErrSource](#)
- [ErrText](#)
- [ExpireTime](#)

- [FileFilter](#)
- [FileFilterNum](#)
- [FileName](#)
- [FileNameExt](#)
- [Flags](#)
- [FlagsExt](#)
- [FlagsTitlebar](#)
- [Floatable](#)
- [FloatLeft](#)
- [FloatTop](#)
- [FmtBigIntFlags](#)
- [FmtBoolString](#)
- [FmtBoolStyle](#)
- [FmtDateString](#)
- [FmtDateStyle](#)
- [FmtDecimalFlags](#)
- [FmtFloatFlags](#)
- [FmtIntFlags](#)
- [FmtOutput](#)
- [FmtPostComma](#)
- [FmtTimeFlags](#)
- [FnCount](#)
- [FnIndex1](#)
- [FnIndex2](#)
- [FnIndex3](#)
- [FnMode1](#)
- [FnMode2](#)
- [FnMode3](#)
- [FnResult](#)

- [FocusByMouse](#)
- [FocusSelect](#)
- [Font](#)
- [FontAttr](#)
- [FontAttrSel](#)
- [FontBox](#)
- [FontIvl](#)
- [FontName](#)
- [FontParent](#)
- [FontParentScala](#)
- [FontParentTitle](#)
- [FontScala](#)
- [FontSize](#)
- [FontSizeTwips](#)
- [FontTitle](#)
- [Format](#)
- [FormatDate](#)
- [FormatEnum](#)
- [FormatTime](#)
- [FormatWrap](#)
- [FormHeight](#)
- [FormType](#)
- [FormWidth](#)
- [Frame](#)
- [FrameEnd](#)
- [FrameStart](#)
- [FullName](#)

- [GanttFlags](#)
- [GrayingModeUser](#)
- [GridHeight](#)
- [GridWidth](#)
- [Group](#)
- [GroupColBkg](#)
- [GroupColFg](#)
- [GroupDrawSelect](#)
- [Grouping](#)
- [GsEncryption](#)
- [GsOptions](#)
- [GsPasswordOwner](#)
- [GsPasswordUser](#)
- [GsPdfColorProfile](#)
- [GsPermissions](#)
- [GsVersion](#)

- [HdrColFg](#)
- [HdrHeight](#)
- [HdrIcon](#)
- [HdrImageOption](#)

- [HdrImageTile](#)
- [HdrImageTileUser](#)
- [HdrShadeCol1](#)
- [HdrShadeCol2](#)
- [HdrShadeMode](#)
- [HdrTileSize](#)
- [HdrVerticalText](#)
- [HdrWordBreak](#)
- [Height](#)
- [HeightIvl](#)
- [HeightMode](#)
- [Help](#)
- [HelpFile](#)
- [HelpTip](#)
- [HelpTipSysFont](#)
- [HelpTipTimeDelay](#)
- [HelpTipTimeShow](#)
- [HvphMarginTwips](#)

- [Icon](#)
- [ID](#)
- [ImageOption](#)
- [ImageSource](#)
- [ImageTile](#)
- [ImageTileUser](#)
- [IndexColBkg](#)
- [IndexColFg](#)
- [IndexData](#)
- [IndexFont](#)
- [InputCheck](#)
- [InputCtrl](#)
- [InputMode](#)
- [InstallChromium](#)
- [InstallCtxOffice](#)
- [InstallPathCommon](#)

- [JobCount](#)
- [JobPriority](#)
- [Justify](#)
- [JustifyVert](#)
- [JustifyView](#)

- [LangDisplay](#)
- [LeaveAtEnd](#)
- [LengthMax](#)
- [LineCapEnd](#)
- [LineCapStart](#)
- [LineJustify](#)

- LineWidth
- LinkText
- Locale
- Location
- LstFlags
- LstStyle

- MarginLeft
- MaxBigInt
- MaxDate
- MaxDecimal
- MaxFloat
- MaxInt
- MaxLines
- MaxTime
- MemObjHandle
- MenuAction
- MenuBreak
- MenuCheck
- MenuDefault
- MenuItem
- MenuJustifyRight
- MenuKey
- MenuName
- MenuNameCntxt
- MenuRadioCheck
- MenuSeparator
- MinBigInt
- MinDate
- MinDecimal
- MinFloat
- MinInt
- MinTime
- MobileMode
- ModeColor
- ModeDraw
- ModeEffect
- ModeOptimize
- ModeZoom
- Modified
- ModifiedUser
- ModuleUnit
- MouseKeySort
- MultiSelect

- Name
- NameComputer
- NameDriver
- NamePort

- NamePrinter
- NameServer
- NameShare
- NameUser
- NodeDynamic
- NodeExpanded
- NodeStyle
- NotifyParent
- NumIcon

- ObjLink
- OffsetLeft
- OffsetTop
- OleDropEffectCtrl
- OleDropEffectCtrlShift
- OleDropEffectShift
- OleDropEffectStandard
- OleDropMode
- OleDropModeRelaxed
- Opacity
- OrderPass
- Orientation

- PageCount
- PageFormat
- PageFormatName
- PageFrom
- PageHeight
- PageNumClm
- PageTo
- PageWidth
- Password
- PasswordChar
- PathName
- PdfAuthor
- PdfCompression
- PdfCreator
- PdfEncryptKeyLen
- PdfFileName
- PdfImageResolution
- PdfJpegQuality
- PdfPasswdOpen
- PdfPasswdOwner
- PdfPrintOrder
- PdfRestriction
- PdfTitle
- PicBPP
- PicDpiDefaultX
- PicDpiDefaultY

- [PicDpiExifX](#)
- [PicDpiExifY](#)
- [PicDpiHeadX](#)
- [PicDpiHeadY](#)
- [PicDpiX](#)
- [PicDpiY](#)
- [PicHeight](#)
- [PictureMode](#)
- [PictureName](#)
- [PopupFlags](#)
- [PopupLineHeight](#)
- [PopupOpen](#)
- [PopupType](#)
- [PopupWidth](#)
- [PreventOverlap](#)
- [PrinterDefault](#)
- [PrinterList](#)
- [PrintMode](#)
- [PrintOptions](#)
- [PrintToFile](#)
- [Priority](#)
- [PriorityIdle](#)
- [ProcCurrent](#)
- [ProcCurrentFull](#)
- [ProcCurrentSub](#)
- [ProgressMax](#)
- [ProgressPos](#)
- [PrtRtfFlags](#)

- [QualityX](#)
- [QualityY](#)

- [Radius](#)
- [Range](#)
- [ReadOnly](#)
- [Repeat](#)
- [ReturnKeyClick](#)
- [RtfAlign](#)
- [RtfEditFlags](#)
- [RtfEffect](#)
- [Ruler](#)

- [SBarStyle](#)
- [ScalaColorBkg](#)
- [ScalaColorBkgBase](#)
- [ScalaColorEdge](#)
- [ScalaColorEnd](#)
- [ScalaColorText](#)

- [ScalaLabelCount](#)
- [ScalaLabels](#)
- [ScalaTextJustify](#)
- [ScalaTextOrigin](#)
- [ScrollbarSize](#)
- [ScrollbarVisible](#)
- [ScrollHeight](#)
- [ScrollLeft](#)
- [ScrollTop](#)
- [ScrollWidth](#)
- [SearchEnabled](#)
- [SearchFlags](#)
- [SearchKeyNext](#)
- [SearchKeyPrev](#)
- [SearchKeyStart](#)
- [SearchPattern](#)
- [SelData](#)
- [SelectedVisible](#)
- [SelectorItem](#)
- [SelectorSubItem](#)
- [SelectPrt](#)
- [SelectPvw](#)
- [ShadeCol1](#)
- [ShadeCol2](#)
- [ShadeMode](#)
- [ShapeType](#)
- [ShowArrow](#)
- [ShowFocus](#)
- [ShowGrip](#)
- [ShowPageOffset](#)
- [ShowText](#)
- [Silent](#)
- [Size](#)
- [SizeScala](#)
- [SkipPrint](#)
- [SoftLineCount](#)
- [SpellIgnoreAllCapsWords](#)
- [SpellSplitWords](#)
- [SplitFlags](#)
- [SplitStyle](#)
- [StatusItemPos](#)
- [StatusItemText](#)
- [StoClmType](#)
- [StoView](#)
- [StoViewElmTypes](#)
- [StreamSource](#)
- [StyleBorder](#)
- [StyleButton](#)
- [StyleCaption](#)
- [StyleCloseBox](#)
- [StyleConsole](#)

- [StyleCorner](#)
- [StyleDisplay](#)
- [StyleFrame](#)
- [StyleGroup](#)
- [StyleIvl](#)
- [StylePen](#)
- [StyleRow](#)
- [StyleShade](#)
- [StyleTab](#)
- [StyleTheme](#)
- [SubDivisions](#)
- [SubDivisionsCount](#)
- [SubItemStretch](#)

- [TabOrder](#)
- [TabPos](#)
- [TabSelect](#)
- [TabStop](#)
- [TaskbarButton](#)
- [TerminalSession](#)
- [TextLabel](#)
- [ThemeBaseName](#)
- [ThemeBkg](#)
- [ThemeColArrowChecked](#)
- [ThemeColArrowDisabled](#)
- [ThemeColArrowDropDisabled](#)
- [ThemeColArrowDropError](#)
- [ThemeColArrowDropHighlight](#)
- [ThemeColArrowDropNormal](#)
- [ThemeColArrowDropPressed](#)
- [ThemeColArrowHighlight](#)
- [ThemeColArrowIvl](#)
- [ThemeColArrowNormal](#)
- [ThemeColArrowPressed](#)
- [ThemeColArrowScrollHighlight](#)
- [ThemeColArrowScrollNormal](#)
- [ThemeColArrowScrollPressed](#)
- [ThemeColBorderActive](#)
- [ThemeColBorderBase](#)
- [ThemeColBorderBoundary](#)
- [ThemeColBorderCellActiveHorz](#)
- [ThemeColBorderCellActiveVert](#)
- [ThemeColBorderCheckDisabled](#)
- [ThemeColBorderChecked](#)
- [ThemeColBorderCheckHighlight](#)
- [ThemeColBorderCheckNormal](#)
- [ThemeColBorderDisabled](#)
- [ThemeColBorderError](#)
- [ThemeColBorderFocussed](#)
- [ThemeColBorderHdrBase](#)

- [ThemeColBorderHighlight](#)
- [ThemeColBorderHover](#)
- [ThemeColBorderInactive](#)
- [ThemeColBorderIvl](#)
- [ThemeColBorderNormal](#)
- [ThemeColBorderPageNormal](#)
- [ThemeColBorderPressed](#)
- [ThemeColBorderReadOnly](#)
- [ThemeColBorderSelectedActive](#)
- [ThemeColBorderSelectedInactive](#)
- [ThemeColBorderVisited](#)
- [ThemeColCheckActive](#)
- [ThemeColCheckDisabled](#)
- [ThemeColCheckHighlight](#)
- [ThemeColCheckInactive](#)
- [ThemeColCheckNormal](#)
- [ThemeColCheckPressed](#)
- [ThemeColCheckSelectedActive](#)
- [ThemeColCheckSelectedInactive](#)
- [ThemeColEditorAlpha](#)
- [ThemeColEditorCKeywords](#)
- [ThemeColEditorComment](#)
- [ThemeColEditorConstants](#)
- [ThemeColEditorCurrentLine](#)
- [ThemeColEditorCustomHighlight](#)
- [ThemeColEditorDataTypes](#)
- [ThemeColEditorFunctions](#)
- [ThemeColEditorKeywords](#)
- [ThemeColEditorNumber](#)
- [ThemeColEditorOccurrences](#)
- [ThemeColEditorPreProcessor](#)
- [ThemeColEditorSelection](#)
- [ThemeColEditorText](#)
- [ThemeColFillActive](#)
- [ThemeColFillBarDisabled](#)
- [ThemeColFillBarNormal](#)
- [ThemeColFillBarShadeDisabled](#)
- [ThemeColFillBarShadeNormal](#)
- [ThemeColFillBase](#)
- [ThemeColFillCaption](#)
- [ThemeColFillCaptionActive](#)
- [ThemeColFillCaptionInactive](#)
- [ThemeColFillChannelDisabled](#)
- [ThemeColFillChannelHighlight](#)
- [ThemeColFillChannelNormal](#)
- [ThemeColFillChannelPressed](#)
- [ThemeColFillCheckDisabled](#)
- [ThemeColFillChecked](#)
- [ThemeColFillCheckHighlight](#)
- [ThemeColFillCheckNormal](#)
- [ThemeColFillDisabled](#)

- [ThemeColFillDropDisabled](#)
- [ThemeColFillDropError](#)
- [ThemeColFillDropHighlight](#)
- [ThemeColFillDropNormal](#)
- [ThemeColFillDropPressed](#)
- [ThemeColFillEnd](#)
- [ThemeColFillError](#)
- [ThemeColFillFocussed](#)
- [ThemeColFillHdrBase](#)
- [ThemeColFillHdrDisabled](#)
- [ThemeColFillHdrHighlight](#)
- [ThemeColFillHdrNormal](#)
- [ThemeColFillHdrPressed](#)
- [ThemeColFillHighlight](#)
- [ThemeColFillHover](#)
- [ThemeColFillInactive](#)
- [ThemeColFillIndentNormal](#)
- [ThemeColFillNormal](#)
- [ThemeColFillPressed](#)
- [ThemeColFillReadOnly](#)
- [ThemeColFillRulerNormal](#)
- [ThemeColFillScrollHighlight](#)
- [ThemeColFillScrollNormal](#)
- [ThemeColFillScrollPressed](#)
- [ThemeColFillSelectedActive](#)
- [ThemeColFillSelectedInactive](#)
- [ThemeColFillShadeChecked](#)
- [ThemeColFillShadeDisabled](#)
- [ThemeColFillShadeHighlight](#)
- [ThemeColFillShadeNormal](#)
- [ThemeColFillShadePressed](#)
- [ThemeColFillSpinArrowDisabled](#)
- [ThemeColFillSpinArrowError](#)
- [ThemeColFillSpinArrowHighlight](#)
- [ThemeColFillSpinArrowNormal](#)
- [ThemeColFillSpinArrowPressed](#)
- [ThemeColFillSpinDisabled](#)
- [ThemeColFillSpinError](#)
- [ThemeColFillSpinHighlight](#)
- [ThemeColFillSpinNormal](#)
- [ThemeColFillSpinPressed](#)
- [ThemeColFillSplitterDisabled](#)
- [ThemeColFillSplitterNormal](#)
- [ThemeColFillTabNormal](#)
- [ThemeColFillThumbDisabled](#)
- [ThemeColFillThumbHighlight](#)
- [ThemeColFillThumbNormal](#)
- [ThemeColFillThumbPressed](#)
- [ThemeColFillVisited](#)
- [ThemeColForeSplitterDisabled](#)
- [ThemeColForeSplitterNormal](#)

- [ThemeColGrip](#)
- [ThemeColImageNormal](#)
- [ThemeColSeparator](#)
- [ThemeColSeparatorClmBase](#)
- [ThemeColSeparatorClmFixedBase](#)
- [ThemeColSeparatorHdrBase](#)
- [ThemeColSeparatorLineBase](#)
- [ThemeColSortHdrDisabled](#)
- [ThemeColSortHdrHighlight](#)
- [ThemeColSortHdrNormal](#)
- [ThemeColSortHdrPressed](#)
- [ThemeColTextActive](#)
- [ThemeColTextCaption](#)
- [ThemeColTextCaptionActive](#)
- [ThemeColTextCaptionInactive](#)
- [ThemeColTextChecked](#)
- [ThemeColTextDisabled](#)
- [ThemeColTextError](#)
- [ThemeColTextFocussed](#)
- [ThemeColTextHdrDisabled](#)
- [ThemeColTextHdrHighlight](#)
- [ThemeColTextHdrNormal](#)
- [ThemeColTextHdrPressed](#)
- [ThemeColTextHighlight](#)
- [ThemeColTextHover](#)
- [ThemeColTextInactive](#)
- [ThemeColTextNormal](#)
- [ThemeColTextPressed](#)
- [ThemeColTextRulerNormal](#)
- [ThemeColTextSelectedActive](#)
- [ThemeColTextSelectedInactive](#)
- [ThemeColTextVisited](#)
- [ThemeColTickDisabled](#)
- [ThemeColTickNormal](#)
- [Themed](#)
- [ThemeDescription](#)
- [ThemeFontBox](#)
- [ThemeFontCaption](#)
- [ThemeFontIvl](#)
- [ThemeFontText](#)
- [ThemeFontTextHdr](#)
- [ThemeMenuCntxtTileSize](#)
- [ThemeMenuItemSize](#)
- [ThemeName](#)
- [ThemeOptFillShade](#)
- [ThemeSetId](#)
- [ThemeSizeBorderChecked](#)
- [ThemeSizeBorderDefault](#)
- [ThemeSizeBorderDisabled](#)
- [ThemeSizeBorderHighlight](#)
- [ThemeSizeBorderNormal](#)

- ThemeSizeBorderPressed
- ThemeSizeFixed
- TickAlignment
- TickInterval
- TickLineSize
- TickPageSize
- TiffFileName
- TifMode
- TileButtonSpacing
- TileNameHeader
- TileNameMenu
- TileNameNormal
- TileNamePressed
- TileNameSelected
- TileNameUser
- TileSize
- TileTheme
- TimeExternal
- TimeStart
- TimeUntil
- TitleColorBkg
- TitleColorFg
- TitleJustify
- TitleText
- Toggled
- Toolbars
- TrayFrame
- TreeFlags
- TypeBarcode
- TypeButton
- TypeFile
- TypeIcon

- Unicode
- Unit
- Url
- UseLogSize
- Utf8

- Version
- Vertical
- VerticalText
- ViewId
- ViewMode
- ViewType
- Visible
- VisibleOrder
- VisiblePrint

Kontakt

- WaitCursor
 - Width
 - WidthPen
 - WordBreak
-
- ZoomEffective
 - ZoomFactor
 - ZOrder

Kontakt

Eigenschaften von Objekten des Benutzersystems

In dieser Liste sind alle Eigenschaften von Objekten des Benutzersystems aufgeführt. Liste sortiert

nach Gruppen,

Siehe Alphabetische

Liste aller

Eigenschaften

- UrmOldPropAutoLoadDataStruct
- UrmOldPropBatches
- UrmOldPropConfiguration
- UrmOldPropDbInfo
- UrmOldPropDbMaintenance
- UrmOldPropEditBaseParams
- UrmOldPropEditColors
- UrmOldPropEditMenus
- UrmOldPropEditPdv
- UrmOldPropExtAccess
- UrmOldPropInitProc
- UrmOldPropLimitedDataStruct
- UrmOldPropLoadDbDefinition
- UrmOldPropMenu
- UrmOldPropMenuLevel
- UrmOldPropNotes
- UrmOldPropOwnBaseParams
- UrmOldPropOwnFileParams
- UrmOldPropOwnMainMenu
- UrmOldPropPdvLevel
- UrmOldPropProcedures
- UrmOldPropProtocol
- UrmOldPropStartBatch
- UrmOldPropTermProc
- UrmOldPropTestFunctions
- UrmOldPropTextLevel
- UrmOldPropTextSystem
- UrmPropActive
- UrmPropComment
- UrmPropCreated
- UrmPropCreatedUser
- UrmPropDsgPlugInPwd
- UrmPropExpirationDate
- UrmPropFirstname
- UrmPropID
- UrmPropInitials
- UrmPropLastname
- UrmPropMaxSessions
- UrmPropModified
- UrmPropModifiedUser
- UrmPropName
- UrmPropOwner
- UrmPropOwnerGroup

Kontakt

- [UrmPropPassword](#)
- [UrmPropPwdCapitals](#)
- [UrmPropPwdDigits](#)
- [UrmPropPwdLocked](#)
- [UrmPropPwdMinLength](#)
- [UrmPropPwdSpecials](#)
- [UrmPropShortcut](#)
- [UrmPropStartProc](#)
- [UrmPropUserGroup](#)

Kontakt

_UrmOldPropAutoLoadDataStruct —

Datenstruktur laden

Typ logic

Liste, Konvertierung des

Siehe alten Benutzersystems,

Benutzerpflege

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.

Die Eigenschaft entspricht dem Benutzerparameter "Datenstruktur laden".

Kontakt

_UrmOldPropBatches

Programmbenutzung Abläufe

Typ logic

Liste, Konvertierung des

Siehe alten Benutzersystems,

Benutzerpflege

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.

Die Eigenschaft entspricht der Programmbenutzung "Abläufe".

Kontakt

_UrmOldPropConfiguration

Programmberechtigung Konfiguration

Typ logic

Siehe Liste, Konvertierung des alten

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.

Die Eigenschaft entspricht der Programmberechtigung "Konfiguration".

Kontakt

_UrmOldPropDbInfo Programmberichtigung DB-Info
im Textsystem Typ logic

Siehe Liste, Konvertierung des alten

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.
Die Eigenschaft entspricht der Programmberichtigung "DB-Info im Textsystem".

Kontakt

_UrmOldPropDbMaintenance □

Programmberechtigung Datenbankpflege Typ logic

Siehe Liste, Konvertierung des alten

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.
Die Eigenschaft entspricht der Programmberechtigung "Datenbankpflege".

Kontakt

_UrmOldPropEditBaseParams

Programmberechtigung Basisparameter

Typ logic

Siehe Liste, Konvertierung des alten

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.

Die Eigenschaft entspricht der Programmberechtigung "Basisparameter".

Kontakt

_UrmOldPropEditColors

Programmbenutzung Farbpaletten

Typ logic

Liste, Konvertierung des alten

Siehe Benutzersystems,

Benutzerpflege

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.

Die Eigenschaft entspricht der Programmbenutzung "Farbpaletten".

Kontakt

_UrmOldPropEditMenus

Programmberechtigung Menüs

Typ logic

[Liste, Konvertierung des](#)

Siehe alten Benutzersystems,

[Benutzerpflege](#)

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.

Die Eigenschaft entspricht der Programmberechtigung "Menüs".

Kontakt

_UrmOldPropEditPdv

Programmberechtigung Druckertreiber

Typ logic

Siehe Liste, Konvertierung des alten

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.

Die Eigenschaft entspricht der Programmberechtigung "Druckertreiber".

Kontakt

_UrmOldPropExtAccess □

Programmberechtigung Externer Zugriff Typ logic

Siehe Liste, Konvertierung des alten

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.
Die Eigenschaft entspricht der Programmberechtigung "Externer Zugriff".

Ist diese Eigenschaft bei der Hauptbenutzergruppe des Benutzers gesetzt, kann sich dieser mit folgenden Komponenten an die Datenbank anmelden:

- Externe Windows Programmierschnittstelle
- ODBC-Treiber
- PHP-Schnittstelle
- Druckprozessor
- SOA-Service
- Web-Schnittstelle
- DbaConnect() aus Standard- und Advanced-Client

Der Zugriff kann für einzelne Komponenten über Erweiterte Eigenschaften bei der Hauptbenutzergruppe deaktiviert werden.

Kontakt

_UrmOldPropInitProc ↗

Prozedur bei Start

Typ alpha(20)

Liste,

Konvertierung

Siehe des alten

Benutzersystems,

Benutzerpflege

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten. Die Eigenschaft entspricht dem Benutzerparameter "Prozedur bei Start". Die Eigenschaft wird nur ausgewertet, wenn der Standard-Client (c16_winc.exe) gestartet wurde.

Kontakt

_UrmOldPropLimitedDataStruct —

Limitierte Datenstruktur

Typ logic

Liste, Konvertierung des

Siehe alten Benutzersystems,

Benutzerpflege

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.

Die Eigenschaft entspricht dem Benutzerparameter "Limitierte Datenstruktur".

Kontakt

_UrmOldPropLoadDbDefinition

Programmberechtigung Datensicherung

Typ logic

Siehe Liste, Konvertierung des alten

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.
Die Eigenschaft entspricht der Programmberechtigung "Datensicherung".

Kontakt

_UrmOldPropMenu

Benutzermenü Typ

alpha(20)

Liste,

Konvertierung

Siehe des alten

Benutzersystems,

Benutzerpflege

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.

Die Eigenschaft entspricht dem Benutzerparameter "Menü".

Kontakt

_UrmOldPropMenuLevel

Berechtigungslevel Menüs

Typ **byte**

[Liste, Konvertierung](#)

Siehe [des alten](#)

[Benutzersystems,](#)

[Benutzerpflege](#)

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.

Die Eigenschaft entspricht der Programmberichtigung "Berechtigung Menüs".

Kontakt

_UrmOldPropNotes

Programmberechtigung Notizen

Typ logic

Liste, Konvertierung des

Siehe alten Benutzersystems,

Benutzerpflege

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.

Die Eigenschaft entspricht der Programmberechtigung "Notizen".

Kontakt

_UrmOldPropOwnBaseParams —

Eigene Basisparameter

Typ logic

Liste, Konvertierung des

Siehe alten Benutzersystems,

Benutzerpflege

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.

Die Eigenschaft entspricht dem Benutzerparameter "Eigene Basisparameter".

Kontakt

_UrmOldPropOwnFileParams —

Eigene Dateiparameter

Typ logic

Liste, Konvertierung des

Siehe alten Benutzersystems,

Benutzerpflege

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.

Die Eigenschaft entspricht dem Benutzerparameter "Eigene Dateiparameter".

Kontakt

_UrmOldPropOwnMainMenu —

Hauptmenü ersetzen

Typ logic

Liste, Konvertierung des

Siehe alten Benutzersystems,

Benutzerpflege

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.

Die Eigenschaft entspricht dem Benutzerparameter "Hauptmenü ersetzen".

Kontakt

_UrmOldPropPdvLevel

Berechtigungslevel Druckertreiber

Typ **byte**

[Liste, Konvertierung des alten](#)

Siehe [Benutzersystems](#),

[Benutzerpflege](#)

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.

Die Eigenschaft entspricht der Programmberichtigung "Berechtigung Druckertreiber".

Kontakt

_UrmOldPropProcedures

Programmberechtigung Prozeduren

Typ logic

Liste, Konvertierung des alten

Siehe Benutzersystems,

Benutzerpflege

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.

Die Eigenschaft entspricht der Programmberechtigung "Prozeduren".

Kontakt

_UrmOldPropProtocol

Programmbenutzung Protokoll starten

Typ logic

Siehe Liste, Konvertierung des
alten Benutzersystems,
Benutzerpflege

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.

Die Eigenschaft entspricht der Programmbenutzung "Protokoll starten".

Kontakt

_UrmOldPropStartBatch

Benutzerparameter Ablauf bei Start

Typ alpha(20)

Liste, Konvertierung des alten

Siehe Benutzersystems,

Benutzerpflege

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.

Die Eigenschaft entspricht dem Benutzerparameter "Ablauf bei Start".

Kontakt

_UrmOldPropTermProc —

Prozedur am Ende

Typ alpha(20)

Liste,

Konvertierung des

Siehe alten

Benutzersystems,

Benutzerpflege

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.

Die Eigenschaft entspricht dem Benutzerparameter "Prozedur am Ende".

Kontakt

_UrmOldPropTestFunctions

Programmberechtigung Testfunktionen

Typ logic
Siehe Liste, Konvertierung des alten

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.
Die Eigenschaft entspricht der Programmberechtigung "Testfunktionen".



Ist diese nicht gesetzt, kann der Designer nicht aus dem Standard-Client heraus gestartet werden.

Kontakt

_UrmOldPropTextLevel

Berechtigungslevel Textsystem

Typ byte

Liste, Konvertierung des

Siehe alten Benutzersystems,

Benutzerpflege

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.

Die Eigenschaft entspricht der Programmberichtigung "Berechtigung Textsystem".

Kontakt

_UrmOldPropTextSystem

Programmberechtigung Textsystem

Typ logic

Liste, Konvertierung des alten

Siehe Benutzersystems,

Benutzerpflege

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten.

Die Eigenschaft entspricht der Programmberechtigung "Textsystem".

Kontakt

_UrmPropActive []

Objekt ist aktiv

Typ logic
Liste,

Siehe Objekte

Über diese Eigenschaft kann ein Benutzer der Benutzerpflege vorübergehend deaktiviert werden. Ein deaktiverter Benutzer kann sich nicht mehr an die Datenbank anmelden.

Ob ein Objekt der Benutzerpflege aktiv ist oder nicht, wird nur bei der Anmeldung des Benutzers überprüft. Eine Änderung zur Laufzeit wirkt sich somit nicht auf angemeldete Benutzer aus. Änderungen werden erst nach einer Neuanmeldung wirksam.

Kontakt

_UrmPropComment

Kommentar

Typ [alpha\(4096\)](#)

Siehe [Liste](#), [Objekte](#)

In dieser Eigenschaft kann im Objekt der Benutzerpflege ein Kommentar hinterlegt werden. Der Kommentar kann maximal 4096 Zeichen lang sein. Er wird von CONZEPT 16 nicht ausgewertet.

Kontakt

_UrmPropCreated

Erstellungszeitpunkt

Typ **bigint**

Siehe [Liste, Objekte](#),

[UrmPropModified](#)

Diese Eigenschaft enthält den Zeitpunkt, zu dem das Objekt erzeugt wurde. Die Eigenschaft kann nur gelesen werden.

Der Inhalt der Eigenschaft kann mit Hilfe der Befehle [CnvCB\(\)](#), [CnvDB\(\)](#) oder [CnvTB\(\)](#) in einen [caltime](#)-, [date](#)- oder [time](#)-Wert umgewandelt werden. Dabei wird automatisch die lokale Zeitzone berücksichtigt.

Kontakt

_UrmPropCreatedUser

Benutzer, der das Objekt erstellt hat

Typ alpha(20)

Siehe Liste, Objekte

Diese Eigenschaft enthält den Namen des Benutzers, der das Objekt angelegt hat. Die Eigenschaft kann nur gelesen werden.

Kontakt

_UrmPropDsgPlugInPwd



**Kennwort des Benutzers für die Plugin-Schnittstelle Typ
alpha(40)**

Siehe Liste, Objekte, Benutzerpflege

In dieser Eigenschaft kann das Kennwort des Benutzers für die Plugin-Schnittstelle angegeben werden. Das Kennwort wird benötigt, wenn ein Plugin mit dem Designer des Benutzers interagieren soll. Es kann maximal 40 Zeichen lang sein. Es muss ein Kennwort vergeben sein, damit sich ein Plugin am Designer authentifizieren kann.

Beispiel

```
tHdlUser # UrmOpen(_UrmTypeUser, _UrmLock, 'SUPERUSER'); tHdlUser->UrmPropSet(_UrmPropDsgPlugInPwd
```

Kontakt

_UrmPropExpirationDate —

Gültigkeitsdatum

Typ date

Siehe Liste, Objekte,

In dieser Eigenschaft wird das Gültigkeitsdatum des Benutzers angegeben. Ist das Datum überschritten, ist der Benutzer nicht mehr gültig. Der Benutzer kann sich nicht mehr anmelden. Der Zeitpunkt, zu dem sich der Benutzer nicht mehr anmelden kann, ist der darauffolgende Tag um 00:00 Uhr UTC. D. h. eine Anmeldung ist ab dem Folgetag um 01:00 Uhr (Deutschland) bzw. um 02:00 Uhr (Deutschland, Sommerzeit) nicht mehr möglich.

Kontakt

_UrmPropFirstname

Vorname des Benutzers

Typ alpha(4096)

Liste, Objekte,

Siehe _UrmPropLastname,

_UrmPropInitials,

Benutzerpflege

In dieser Eigenschaft kann der Vorname des Benutzers angegeben werden. Die Eigenschaft wird nicht durch das System ausgewertet.

Kontakt

_UrmPropID

ID des Objekts

Typ int

Liste,

Siehe Objekte

Die Eigenschaft ID wird von CONZEPT 16 gesetzt und kann nur gelesen werden. Die ID ist innerhalb der Datenbank eindeutig und kann zur Identifizierung eines Benutzers verwendet werden. Wird der Benutzer mit dem OEM-Kit in eine andere Datenbank eingelesen, bekommt er eine neue ID. Die ID ist statischer Natur und darf nicht mit der Benutzernummer oder der Benutzer-ID verwechselt werden, die bei jeder Anmeldung des Benutzers vergeben wird.

Kontakt

_UrmPropInitials

Zweiter Vorname des Benutzers

Typ alpha(4096)

Liste, Objekte,

Siehe _UrmPropLastname,

_UrmPropFirstname,

Benutzerpflege

In dieser Eigenschaft kann der zweite Vorname des Benutzers angegeben werden. Die Eigenschaft wird vom System nicht ausgewertet.

Kontakt

_UrmPropLastname

Nachname des Benutzers

Typ **alpha(4096)**

Liste, Objekte,

Siehe [_UrmPropFirstname](#),

[_UrmPropInitials](#),

[Benutzerpflege](#)

In dieser Eigenschaft kann der Nachname des Benutzers angegeben werden. Die Eigenschaft wird nicht vom System ausgewertet.

Kontakt

_UrmPropMaxSessions □ Maximale Anzahl der gleichzeitigen, eindeutigen Sessions Typ int

Siehe Liste, Objekte, Benutzerpflege

Mit dieser Eigenschaft kann die maximale Anzahl von gleichzeitigen Sessions eines Benutzers angegeben werden. Der Benutzer kann sich nicht häufiger als die hier angegebene Anzahl gleichzeitig an die Datenbank anmelden. Wird die Anzahl der Sessions reduziert, wirkt sich das erst bei der nächsten Anmeldung des Benutzers aus.



Die Anzahl der gleichzeitigen, eindeutigen Benutzersessions wirkt sich ebenfalls auf die Verwendung des Debuggers aus. Da der Debugger einen eigenen Benutzerkontext benötigt, ist darauf zu achten, dass ein zweiter Benutzer entsprechend angemeldet werden kann.

_UrmPropModified

Zeitpunkt der letzten Änderung

Typ [bigint](#)

Siehe [Liste, Objekte](#),

[_UrmPropCreated](#)

Diese Eigenschaft enthält Datum und Uhrzeit der letzten Änderung des Objekts. Die Eigenschaft kann nur gelesen werden.

Der Inhalt der Eigenschaft kann mit Hilfe der Befehle [CnvCB\(\)](#), [CnvDB\(\)](#) oder [CnvTB\(\)](#) in einen [caltime](#)-, [date](#)- oder [time](#)-Wert umgewandelt werden. Dabei wird automatisch die lokale Zeitzone berücksichtigt.

Kontakt

_UrmPropModifiedUser



Benutzer, der das Objekt zuletzt geändert hat Typ

alpha(20)

Siehe [Liste, Objekte](#)

Diese Eigenschaft enthält den Namen des Benutzers, der das Objekt zuletzt geändert hat. Die Eigenschaft kann nur gelesen werden.

Kontakt

_UrmPropName

Name des Objekts

Typ alpha(20)

Liste, Objekte,

Siehe Benutzerpflege,

UrmCreate()

In dieser Eigenschaft wird der Name des Objekts gespeichert. Der Name wird beim Erzeugen des Objekts (zum Beispiel mit UrmCreate()) festgelegt. Der Name besitzt eine Maximallänge (siehe UrmCreate()) und darf nicht mit einem Unterstrich (_) beginnen, keine Steuerzeichen (ASCII-Code < 32) und keines der folgenden Zeichen beinhalten:

! * ? : ; / ' " \

Kontakt

_UrmPropOwner

Besitzer

Typ alpha(20)

Siehe Liste, Objekte,

Benutzerpflege

Benutzern und Benutzergruppen kann jeweils ein Besitzer zugeordnet werden.

Über die Berechtigungen UrmPermModifyOwner und UrmPermDeleteOwner auf eine Elementgruppe, in der der entsprechende Benutzer oder die Benutzergruppe enthalten ist, kann dann ausschließlich dem Besitzer eingeräumt werden, Änderungen am Objekt durchzuführen oder dieses zu löschen (Siehe dazu auch UrmPropOwnerGroup).



Beim Anlegen eines neuen Benutzers oder einer neuen Benutzergruppe wird diese Eigenschaft automatisch auf den Namen des Benutzers gesetzt, der das neue Objekt angelegt hat. Um den Besitzer nachträglich zu ändern, werden Modify-Rechte auf das Objekt benötigt.

Kontakt

_UrmPropOwnerGroup —

Besitzergruppe

Typ alpha(20)

Siehe Liste, Objekte,

Siehe Benutzerpflege

Benutzern kann jeweils eine Besitzergruppe zugeordnet werden.

Über die Berechtigungen UrmPermModifyOwner und UrmPermDeleteOwner auf eine Elementgruppe, in der der entsprechende Benutzer enthalten ist, kann somit ausschließlich der Besitzergruppe eingeräumt werden, Änderungen am Objekt durchzuführen oder dieses zu löschen (Siehe dazu auch UrmPropOwner).



Beim Anlegen eines neuen Benutzers wird diese Eigenschaft automatisch auf die Besitzergruppe des Benutzers gesetzt, der das neue Objekt angelegt hat. Um die Besitzergruppe nachträglich zu ändern, werden Modify-Rechte auf das Objekt benötigt.

Kontakt

_UrmPropPassword

Kennwort des Benutzers

Typ alpha(40)

Liste, Objekte,

Siehe

In dieser Eigenschaft kann das Kennwort des Benutzers angegeben werden. Das Kennwort wird von dem Benutzer zur Anmeldung an die Datenbank benötigt. Es kann maximal 40 Zeichen lang sein. Über weitere Eigenschaften können bestimmte Merkmale des Kennwortes definiert werden.

_UrmPropPwdMinLength Mindestlänge

_UrmPropPwdCapitals Mindestanzahl der Großbuchstaben

_UrmPropPwdDigits Mindestanzahl der Ziffern

_UrmPropPwdSpecials Mindestanzahl der Sonderzeichen

Die Eigenschaft kann nur gesetzt werden. Wird beim Setzen des Passwortes eine oder mehrere Bedingungen des Kennwortes verletzt, wird der Fehlerwert **ErrData** zurückgegeben.

Beispiel

```
tHdlUser # UrmOpen(_UrmTypeUser, _UrmLock, 'SUPERUSER'); tHdlUser->UrmPropSet(_UrmPropPwdMinLength
```

Kontakt

_UrmPropPwdCapitals

Mindestanzahl der Großbuchstaben

Typ int

Liste, Objekte,

Siehe [UrmPropPassword](#),

[Benutzerpflege](#)

Mit dieser Eigenschaft kann die Mindestanzahl der Großbuchstaben in einem Kennwort (siehe [UrmPropPassword](#)) angegeben werden. Die Eigenschaft muss vor dem Setzen des Passwortes erfolgen. Nachträgliche Änderungen wirken sich nicht auf ein vorhandenes Passwort aus.

Kontakt

_UrmPropPwdDigits

Mindestanzahl der Ziffern

Typ int

Liste, Objekte,

Siehe [UrmPropPassword](#),

[Benutzerpflege](#)

In dieser Eigenschaft wird die Mindestanzahl der Ziffern bestimmt, die das Kennwort ([UrmPropPassword](#)) enthalten muss. Es kann ein Wert zwischen 0 und 5 angegeben werden. Die Eigenschaft muss vor dem Setzen des Passwortes erfolgen. Nachträgliche Änderungen wirken sich nicht auf ein vorhandenes Passwort aus.

Kontakt

_UrmPropPwdLocked

Benutzer darf Kennwort nicht ändern

Typ logic

Liste, Objekte,

Siehe [UrmPropPassword](#),

[Benutzerpflege](#)

Mit dieser Eigenschaft kann verhindert werden, dass das Kennwort (siehe [UrmPropPassword](#)) eines Benutzers verändert wird. Dies ist zum Beispiel sinnvoll, wenn mehrere Personen die gleichen Anmeldedaten für die Datenbank verwenden. Damit diese Einstellung nicht unterlaufen wird, ist es wichtig darauf zu achten, dass die Anwender keine Änderungsrechte ([UrmPermModify](#)) auf das Benutzerobjekt haben.

Kontakt

_UrmPropPwdMinLength

Mindestlänge des Passwortes

Typ **int**

[Liste](#), [Objekte](#),

Siehe [UrmPropPassword](#),

[Benutzerpflege](#)

In dieser Eigenschaft wird die Mindestlänge des Benutzerkennwortes

([UrmPropPassword](#)) angegeben. Es kann eine Mindestlänge zwischen 1 und 20 Zeichen vorgegeben werden. Auch bei einer angegebenen Mindestlänge kann auf die Angabe eines Kennwertes verzichtet werden.

_UrmPropPwdSpecials

Mindestanzahl der Sonderzeichen

Typ int

Liste, Objekte,

Siehe [UrmPropPassword](#),

[Benutzerpflege](#)

Mit dieser Eigenschaft kann die Mindestanzahl der Sonderzeichen in einem Kennwort (siehe [UrmPropPassword](#)) angegeben werden. Die Eigenschaft muss vor dem Setzen des Passwortes erfolgen. Nachträgliche Änderungen wirken sich nicht auf ein vorhandenes Passwort aus.

Kontakt

_UrmPropShortcut

Kürzel des Benutzers

Typ [alpha\(4096\)](#)

Siehe [Liste, Objekte](#)

In dieser Eigenschaft kann ein Kürzel für einen Benutzer angegeben werden. Die Eigenschaft wird nicht vom System ausgewertet.

Kontakt

_UrmPropStartProc

Prozedur bei Start Typ

alpha(20)

Siehe [Liste](#), [Benutzerpflege](#),

In dieser Eigenschaft kann eine Prozedur angegeben werden, die unmittelbar nach der Anmeldung des Benutzers aufgerufen wird. In der Prozedur kann dann die Applikation gestartet werden. Die Eigenschaft wird nur ausgewertet, wenn der Advanced-Client (c16_apgi.exe) gestartet wurde.

Kontakt

_UrmPropUserGroup

Hauptbenutzergruppe

Typ alpha(20)

Liste, Objekte,

Benutzerpflege,

Siehe Konvertierung

des alten

Benutzersystems

Ein Benutzer kann innerhalb der Benutzerverwaltung mehreren Benutzergruppen zugeordnet sein. Durch die Zuordnung einer Hauptbenutzergruppe können alle Benutzergruppenparameter für den Anwender gesetzt werden, die sich nicht durch die Kombination mehrerer Benutzergruppen ermitteln lassen:

- Startprozedur

Jedem Benutzer kann über die Benutzergruppe nur eine Startprozedur (_UrmPropStartProc) zugeordnet werden, weshalb diese aus der Hauptbenutzergruppe ermittelt wird.

- Berechtigungseigenschaften für den Standard-Client

Die Berechtigungseigenschaften für den Standard-Client werden durch die _UrmOldProp... Konstanten definiert (Siehe Konvertierung des alten Benutzersystems). Diese Berechtigungen können nicht durch die Kombination mehrerer Benutzergruppen ermittelt werden, weshalb diese von der Hauptbenutzergruppe abgeleitet werden.

Bei der Konvertierung des alten Benutzersystems in das neue Benutzersystem wird in dieser Eigenschaft der Hauptbenutzer des Benutzers abgelegt. Die Eigenschaft ist leer, wenn der alte Benutzer keinem Hauptbenutzer untergeordnet war.

Kontakt

Ermitteln und Setzen von Eigenschaften

Aufteilung der Eigenschaften in Gruppen

WinPropGet(),
WinPropSet(),
PrtPropGet(),
Siehe PrtPropSet(),
SysPropGet(),
SysPropSet(),
UrmPropGet(),
UrmPropSet()

Die Eigenschaften sind in vier Gruppen aufgeteilt:

- Eigenschaften von Oberflächen-Objekten
- Eigenschaften von Druck-Objekten
- Eigenschaften von System-Objekten
- Eigenschaften des Benutzersystems

Jede Gruppe besitzt ihre eigenen Befehle Win-, Prt-, Sys- und UrmPropGet/Set mit denen die Eigenschaft ausgelesen bzw. gesetzt werden kann. Das Ansprechen von Eigenschaften in Prozeduren funktioniert prinzipiell wie zur Entwurfszeit im Designer. Ist der Name der Eigenschaft bekannt, kann durch eine Objektreferenz und dem Namen der Eigenschaft, die Eigenschaft bei dem Objekt verändern oder auslesen.

Der Name der Eigenschaft wird für den Zugriff je nach Gruppe um _WinProp, _PrtProp oder _SysProp erweitert. Die Eigenschaften von Benutzer-Objekten können nur über die _UrmProp-Konstanten angesprochen werden.

Beispiele:

Setzen der Eigenschaft Caption bei einem Oberflächen-Objekt:

```
tErg # tHdlFrame->WinPropSet(_WinPropCaption, 'Suchen');
```

Setzen der Eigenschaft Caption bei einem Druck-Objekt:

```
tErg # tObjPrtText->PrtPropSet(_PrtPropCaption, 'Suchen');
```

Auslesen des temporären Pfades des Betriebssystems aus dem Sys-Objekt:

```
tErg # _Sys->SysPropGet(_SysPropTempPath, tTemp);
```

Auslesen des Namens eines Benutzers:

```
tErg # tHdlUser->UrmPropGet(_UrmPropLastname, tRealLastName);
```

In den Beispielen wird der logische Rückgabewert der Befehle nicht ausgewertet. Über diesen Rückgabewert kann festgestellt werden, ob das Lesen oder Setzen einer Eigenschaft erfolgreich war.

Zum Setzen und Lesen von Eigenschaften gibt es noch eine kürzere Schreibweise. In dieser Schreibweise wird vor dem Namen der Eigenschaft ein Kürzel wp, pp

Kontakt

oder sp gesetzt, um zu kennzeichnen, ob es sich um eine Oberflächen-, Druck-, oder System-Eigenschaft handelt. Ein Kürzel für Eigenschaften des Benutzersystems existiert nicht. Die oben angeführten Beispiele können auch wie folgt geschrieben werden:

```
tHdlFrame->wpCaption # 'Suchen'; tObjPrtText->ppCaption # 'Suchen'; tTemp # _Sys->spTempPath;
```

Diese Anweisungen geben keinen Rückgabewert zurück, der den Erfolg der Operation darstellt. Kann eine Eigenschaft nicht gesetzt oder gelesen werden, erfolgt ein entsprechender Laufzeitfehler.

Die Befehle sollten immer dann verwendet werden, wenn das Vorhandensein einer Eigenschaft zu einem Objekt nicht sichergestellt ist. Zum Beispiel in einer Funktion, die ein Objekt übergeben bekommt. In den anderen Fällen kann die kurze Schreibweise verwendet werden.