

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.

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.

PdfFileName ☐

Dateiname des PDF-Dokumentes

Typ alpha(8192)

Siehe Liste, Objekte, PrtJobClose(),
PrtPropSet(), Beispiel

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
```

PdfImageResolution ☐

Auflösung der Grafiken

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 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.

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
```

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.

PdfPasswdOwner



Password des Eigentümers des PDF-Dokuments

Typ alpha

Siehe PdfPasswdOpen, 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 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.

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.

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 wird
PrtJob->ppPdfRestriction # _PdfD
```

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.

PicBPP ☐

Farbauflösung des Bildes

Typ int

Siehe Liste, Objekte,
WinPropGet()

Über diese Eigenschaft kann die Farbauflösung des Bildes ermittelt werden.

Die Eigenschaft kann nur ausgelesen werden.

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.

PicDpiDefaultY



Vertikale DPI-Standardauflösung von Bildern

Typ int

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

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.

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.

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 intSiehe Liste, Objekte, PrtPropGet(),
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 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 intSiehe Liste, Objekte, PrtPropSet(),
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 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.

PicHeight 

Höhe des Bildes

Typ int

Siehe Liste, Objekte,
WinPropGet()

In dieser Eigenschaft kann die Höhe des Bildes abgefragt werden.

Die Eigenschaft kann nur ausgelesen werden.

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.

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.

PictureName ☐
Hintergrundbild
Typ alpha(8192)
Siehe Liste,
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 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.

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.

PicWidth ☐

Breite des Bildes

Typ int

Siehe Liste, Objekte,
WinPropGet()

In dieser Eigenschaft kann die Breite des Bildes abgefragt werden.

Die Eigenschaft kann nur ausgelesen werden.

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(),
WinPropSet()

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(),
WinPropSet()

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.

PrinterDefault ☐

Standard-Drucker

Typ alpha

Siehe Liste, Objekte

Die Eigenschaft legt den Standard-Drucker fest.

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
```

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 wandeln  
PrtPdf->ppPrintMode = _PrtPrintModeTextAsCurves;
```

PrintOptions ☐

Druckoptionen

Typ int

Siehe Liste,
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 muss vor dem Aufruf von PrtJobClose() gesetzt werden.

Beispiel

```
tJob # PrtJobOpen('', 'myjob.prt', _PrtJobOpenRead);if (tJob > 0){ ... tJob->ppPrintOptions # _
```

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).

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.

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.

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.

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;
```

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 Einfügemarke an eine bestimmte Textstelle zu setzen. Im letzteren 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->
```

Repeat ☐

Abspielen des Films wiederholen

Typ logic

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

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.

WinRtfEditAddFileAsPicture

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)

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

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.

ScalaLabelCount



Anzahl der Beschriftung auf der Achsenskala

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.

ScalaLabels



Definiert die Beschriftung der Achsenskala.

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->wpScalaLabels(1) # 'Sample1'
```

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**

- `$(DATE, Start-Datum, Ende-Datum, Inc-Tage, [Format])`
- `$(TIME, Start-Zeit, Ende-Zeit, Inc-Zeit, [Format])`
- `$(INT, Start-Wert, End-Wert, Inc-Wert)`

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, ...)
w Kalenderwoche (1...53)
ww Kalenderwoche (01...53)
y Jahr 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)

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 $axsSample->wpScalaLabels # '~'; // Setzen einzelner Indizes
```

`tLabel` würde im Beispiel den Wert 'A,B~' enthalten.

ScalaTextJustify



Ausrichtung der Beschriftung für die Achsenskala

Typ int

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

Mit dieser Eigenschaft wird die Ausrichtung der Beschriftung für die Achsenskala festgelegt.

Folgende Ausrichtungen sind möglich:

_WinTextJustLeft linksbündige Ausrichtung

_WinTextJustCenter zentrierte Ausrichtung

_WinTextJustRight rechtsbündige Ausrichtung

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).

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.

SearchFlags

Optionen für die Suche im TreeNodeTyp intSiehe Liste, Objekte, WinPropGet(),
WinPropSet()

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(),
WinPropSet()

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,
WinPropGet()

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.

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.

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.

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.

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 SeitenppcObject->ppSelectPvw # 'range()
```

- **odd()**

Dieser Eintrag definiert einen Bereich ungrader Seiten für die eine Vorschau erstellt werden soll.

```
ppcObject->ppSelectPvw # 'odd()';           // Alle ungraden SeitenppcObject->ppSelectPvw # 'odd()
```

- **even()**

Dieser Eintrag definiert einen Bereich grader Seiten für die eine Vorschau erstellt werden soll.

```
ppcObject->ppSelectPvw # 'even()';           // Alle graden SeitenppcObject->ppSelectPvw # 'even()
```

Standardmäßig hat die Eigenschaft den Wert 'range()'. Es wird also für alle Seiten eine Vorschau generiert.

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(),
WinPropSet()

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.

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 Popupliste 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.

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.

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



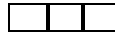
Anzeige von Meldungsfenstern unterdrücken

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.

Size



Größe / Breite des Objektes

Typ intSiehe 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 intListe, 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.

SoftLineCount ☐

Anzahl der Zeilen

Typ int

Siehe Liste, Objekte,
WinPropGet()

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.

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.

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.

SplitFlags



Optionen des GroupSplit-Objekts

Typ intSiehe 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.

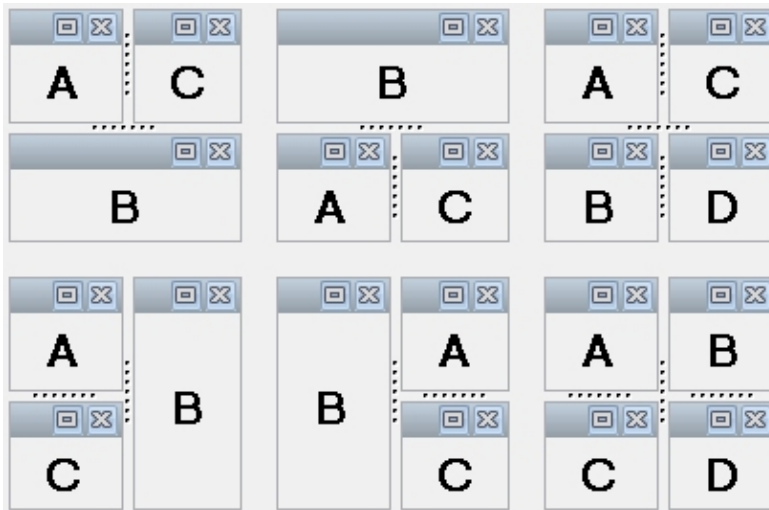
- **_WinSplitNoCrossOverMove**

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(),
WinPropSet()

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

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>PrtStatusPendDelete</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.

StoClmType ☐

Typ der Spalte

Typ intSiehe Liste,
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.

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.

- WinStreamBufField

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.

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(),
WinPropSet()

Ist die Eigenschaft auf false gesetzt, ist die Schließen-Schaltfläche in der Titelleiste des Fensters und die Menüoption "Schließen Alt+F4" des Systemmenüs deaktiviert. 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.

StyleFrame ☐

Fensterstil


Typ int

Siehe Liste,
Objekte

Diese Eigenschaft bestimmt die Darstellung des Fenster.

Es gibt folgende Darstellungen:

<u>_WinWndStandard</u>	Normales Fenster
<u>_WinWndDialog</u>	Fenster mit fester Größe
<u>_WinWndFrameHelp</u>	Fenster mit ?-Schaltfläche
<u>_WinWndDialogHelp</u>	Fenster mit fester Größe und ?-Schaltfläche
<u>_WinWndDialogMax</u>	Ein maximierbares, jedoch nicht in der Größe veränderbares Fenster
<u>_WinWndDialogMin</u>	Ein minimierbares, jedoch nicht in der Größe veränderbares Fenster
<u>_WinWndDialogMinMax</u>	Ein minimierbares / maximierbares jedoch nicht in der Größe veränderbares Fenster
<u>_WinWndDialogSize</u>	Ein in der Größe veränderbares Fenster, das nur über eine Schließen-Schaltfläche verfügt
<u>_WinWndFrameSplash</u>	Fenster ohne Rahmen und Titel (Splash-Screen Darstellung)
<u>_WinWndFrameShape</u>	transparentes Fenster ohne Rahmen und Titel (Splash-Screen Darstellung)
<u>_WinWndNone</u>	Dieser Stil wird nur gesetzt, wenn die Eigenschaft <u>StyleBorder</u> geändert wird, da für die Fälle <u>_WinBorSunken</u> und <u>_WinBorRaised</u> keiner der vordefinierten Stile zutrifft.

StyleIvl 
 Darstellung der Objekte
 im GanttGraph
 Typ int

Siehe Liste, Objekte, WinPropGet(),
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 Linie 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.

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

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 Registerreiters.

Es gibt folgende Darstellungen:

_WinStyleTabRectangle Rechteckige Registerreitertitel

_WinStyleTabSharkFin Haifischflossenähnliche Registerreitertitel


_WinStyleTabTheme Registerreitertitel in der betriebssystemabhängigen Darstellung

_WinStyleTabTrapezoid Trapezförmige Registerreitertitel

_WinStyleTabRectFlat Flache, rechteckige Registerreitertitel



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.

SubDivisionsCount



Anzahl der Skalenunterteilungen der Achse

Typ intListe, 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.

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.

Special (Objektspezifische Eigenschaften) - T bis Z

Eigenschaften sortiert
nach Gruppen,

Siehe Alphabetische Liste
aller Eigenschaften,
Liste aller
Special-Eigenschaften

<u>TaskBarButton</u>	Anzeige des Fensters in der Taskleiste des Betriebssystems
<u>ThemeBaseName</u>	Name des verwendeten <u>Basis-Themes</u>
<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>TiffFileName</u>	Dateiname des TIF-Dokuments
<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>TifMode</u>	Farb-Modus bei Drucken als TIFF-Datei
<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>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 zur Anzeige des Overlays
<u>ViewMode</u>	Anzeigemodus
<u>ViewType</u>	Bedingungen zur Anzeige des Overlays
<u>WaitCursor</u>	Mauszeiger als Sanduhr darstellen
<u>WidthPen</u>	Strichstärke
<u>WordBreak</u>	Automatischer Umbruch bei zu langem Text

Kontakt

<u>XmlFileName</u>	Dateiname des XML-Dokumentes
<u>ZoomEffective</u>	Vergrößerungsfaktor
<u>ZoomFactor</u>	Benutzerdefinierter Vergrößerungsfaktor
<u>ZOrder</u>	Darstellungsreihenfolge

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.

ThemeBaseName ☐

Name zur Identifikation des Basis-Themes

Typ alpha(60)

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

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.

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.



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.

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:

<u>WinTickAlignNone</u>	Keine Markierungen anzeigen.
<u>WinTickAlignRightBottom</u>	Die Markierungen werden in Abhängigkeit von der Eigenschaft <u>Vertical</u> rechts bzw. unterhalb des Kanals angezeigt.
<u>WinTickAlignLeftTop</u>	Die Markierungen werden in Abhängigkeit von der Eigenschaft <u>Vertical</u> links bzw. oberhalb des Kanals angezeigt.
<u>WinTickAlignBoth</u>	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 Werteabstand 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





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 ( /  /  / ) verschoben wird, festgelegt.



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.

TifFileName ☐
 Dateiname des TIF-Dokuments
 Typ alpha(8192)
 Siehe Liste, Objekte,
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 monochomes TIFF erzeugt wird.

Beispiel

```
tHdlPrintJob->ppTifFileName # _Sys->spPathMyDocuments + '\Rechnung.tif';tErr # tHdlPrintJob->PrtJ
```

TifMode



Farb-Modus bei Drucken als TIFF-Datei

Typ intSiehe 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 DarstellungTifModeMonochrome 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;
```


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.

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.

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.

TitleJustify



Ausrichtung der Beschriftung des Achsenstitels

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

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.

TrayFrame (Eigenschaft) ☐

TrayFrame-Applikation zuweisen

Typ int

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

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.

TreeFlags



Darstellung des TreeView-Objektes

Typ int

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

Über diese Eigenschaft wird die Darstellung festgelegt.

Es gibt folgende Einstellungen:

<u>_WinTreeBoxes</u>	Vor jedem Knoten wird ein Plus- bzw. Minus-Zeichen dargestellt
<u>_WinTreeClickFocusSet</u>	Fokus wird auf das TreeView-Objekt gesetzt
<u>_WinTreeImageExpand</u>	Darstellung eines Bildes bei aufgeklappten Knoten
<u>_WinTreeImageGray</u>	Alle Knoten-Symbole werden grau dargestellt
<u>_WinTreeImages</u>	Vor den Knoten wird ein Bildchen angezeigt
<u>_WinTreeLines</u>	Die Knoten sind mit Linien verbunden
<u>_WinTreeLinesAtRoot</u>	In der ersten Ebene werden Linien zu den Wurzelknoten gezeichnet
<u>_WinTreeNoHorzScroll</u>	Horizontale Scrollbar wird ausgeblendet
<u>_WinTreeNoSelectOnFold</u>	Ereignis <u>EvtNodeSelect</u> nur beim Selektieren eines Knotens auslösen
<u>_WinTreeShowSelected</u>	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 $TreeV
```


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,

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 eine größere (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), _StrFromAnsi)

Funktionscode 1 -> StrCnv(StrChar(193), _StrFromAnsi)

Funktionscode 2 -> StrCnv(StrChar(194), _StrFromAnsi)

Funktionscode 3 -> StrCnv(StrChar(195), _StrFromAnsi)

Funktionscode 4 -> StrCnv(StrChar(196), _StrFromAnsi)

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.

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

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`**

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.

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:

_WinBtnNone Keine Aktion soll stattfinden

_WinBtnClose Schließen des Fensters

_WinBtnHelp Aufruf der Hilfe

_WinBtnUserBreak Aktion durch Benutzer abgebrochen

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:

<u>WinAniExternFile</u>	Der in <u>FileName</u> eingetragene externe Film wird verwendet
<u>WinAniScanning</u>	Suchende Taschenlampe
<u>WinAniSearchComputer</u>	Lupe vor Computer
<u>WinAniSearchSheet</u>	Lupe vor Datei
<u>WinAniFileMove</u>	Seiten werden von einem Ordner in einen anderen verschoben
<u>WinAniFileCopy</u>	eine Seite wird in einen anderen Ordner kopiert
<u>WinAniFileTrash</u>	verschieben einer Seite in den Papierkorb
<u>WinAniFileDelete</u>	eine Seite aus dem Papierkorb wird gelöscht

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

Url



Aktuell angezeigte Adresse des WebNavigators

Typ alphaSiehe 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.

Utf8



Barcode im UTF8-Zeichensatz erzeugen

Typ logic

Siehe Liste, Objekte,

_WinBarcode2dORCode

In dieser Eigenschaft kann für QR-Codes (TypeBarcode = _WinBarcode2dORCode) definiert werden, ob der Barcode im UTF8-Zeichensatz (true) oder im ISO8859-1 Zeichensatz erzeugt wird. Standardmäßig ist der Wert auf true gesetzt.

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.

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.

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



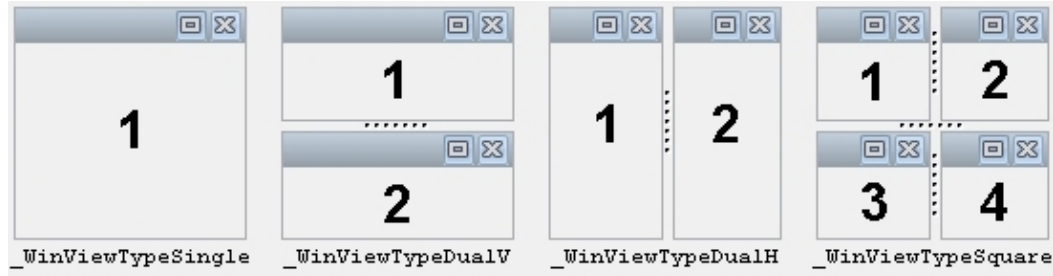
Nummer des Views zur Anzeige von Overlays

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:



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 Scrollbars angezeigt.

ViewType



Bedingungen zur Anzeige von Overlays

Typ intSiehe 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.

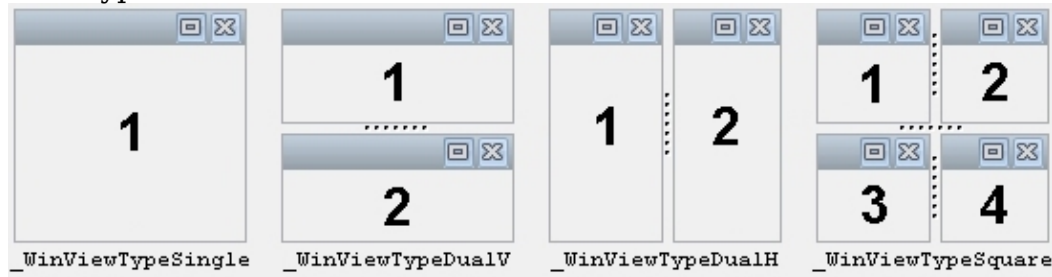
_WinViewTypeDualH (2) Es sind zwei nebeneinanderliegende Views sichtbar.

_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:



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.

WidthPen 

Strichstärke

Typ int

Liste,

Siehe Objekte,

PrtPropGet(),

PrtPropSet()

Die Eigenschaft definiert die Strichstärke zum Zeichnen des PrtDivider- oder PrtBorder-Objektes in logischen Einheiten.

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.

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
```

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.

ZoomFactor



Benutzerdefinierter Vergrößerungsfaktor

Typ intListe, 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 Originalgröße

50 Bild in halber Größe

0 Objektfüllende Darstellung

-50 Bild in halber Größe zentriert zum Objekt

-100 Bild in Originalgröße 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

Siehe Gruppen,
Alphabetische
Liste aller
Eigenschaften

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

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.

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.

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.

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.

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.

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.

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(),
PrtPropSet()

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

Siehe Gruppen,

Alphabetische

Liste aller

Eigenschaften

StyleTheme Darstellung von Oberflächen-Objekten

ThemeName Name des zu verwendenden Themes

ThemeSetId Nummer des ThemeSets

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-Popup 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**

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.

Fill Color (Farb-Eigenschaften für Füllfarbe von Themes)

Eigenschaften

sortiert nach

Siehe Gruppen,

Alphabetische

Liste aller

Eigenschaften

- 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

- ThemeColFillThumbHighlight
- ThemeColFillThumbNormal
- ThemeColFillThumbPressed
- ThemeColFillVisited
- ThemeOptFillShade

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 Oberflächenobjektyp 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
```

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

Beim Definieren / Abfragen der Eigenschaft muss ein Oberflächenobjekttyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeProgress

```
tTheme->wpThemeColFillBarDisabled(_WinTypeProgress) # _WinColRed;tColor # tTheme->wpThemeColFillBarDisabled(_WinTypeProgress)
```

ThemeColFillBarNormal



Füllfarbe der Bar bei Status "normal"

Typ intSiehe Liste, ObjekteIn 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
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeProgress

Beispiele:

```
tTheme->wpThemeColFillBarShadeDisabled(_WinTypeProgress) # _WinColRed;tColor # tTheme->wpThemeCol
```

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
```


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
- WinTypeScrollbar
- WinTypeStatusbar
- WinTypeStatusbarButton
- WinTypeStoList
- WinTypeStoListPopup
- WinTypeToolbar
- WinTypeToolbarButton
- WinTypeToolbarMenu
- WinTypeWindowbar

Beispiele:

```
tTheme->wpThemeColFillBase(_WinTypeCheckBox) # _WinColRed; tColor # tTheme->wpThemeColFillBase(_W
```

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->wpThemeColFillCaption
```

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

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

```
tTheme->wpThemeColFillCaptionActive( _WinTypeAppFrame) # _WinColRed;tColor # tTheme->wpThemeColFillCaptionActive( _WinTypeAppFrame)
```

ThemeColFillCaptionInactive



Füllfarbe der Titelzeile bei Status "inaktiv"

Typ intSiehe Liste, ObjekteIn 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 Oberflächenobjektyp 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
```

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->wpThemeColFi
```

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
```

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
```

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
```


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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeMenuItem

Beispiele:

```
tTheme->wpThemeColFillCheckDisabled(_WinTypeMenuItem) # _WinColRed;tColor # tTheme->wpThemeColFi
```

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
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeMenuItem

Beispiele:

```
tTheme->wpThemeColFillCheckHighlight(_WinTypeMenuItem) # _WinColRed; tColor # tTheme->wpThemeColFi
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeMenuItem

Beispiele:

```
tTheme->wpThemeColFillCheckNormal(_WinTypeMenuItem) # _WinColRed; tColor # tTheme->wpThemeColFillC
```

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

- 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->wpThemeColFill
```

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
```

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 Objekthalt ungültig ist. 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:

- _WinTypeBigIntEdit
- _WinTypeColorEdit
- _WinTypeDateEdit
- _WinTypeDecimalEdit
- _WinTypeEdit
- _WinTypeFloatEdit
- _WinTypeFontNameEdit
- _WinTypeFontSizeEdit
- _WinTypeIntEdit
- _WinTypeTextEdit
- _WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColFillError(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeColFillError
```


In dieser Eigenschaft können Farbkonstanten gesetzt werden.

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

```
tTheme->wpThemeColFillFocussed(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeColFillFoc
```

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 Oberflächenobjektyp 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
```

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 Oberflächenobjektyp 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->wpThemeColFillH
```

ThemeColFillHdrHighlight



Füllfarbe bei Status "hervorgehoben"

Typ intSiehe Liste, ObjekteIn 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 Oberflächenobjektyp 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
```

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 Spaltenköpfe, 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 Oberflächenobjektyp 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->wpThemeColFillHdrNormal
```

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 Oberflächenobjektyp 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
```

ThemeColFillHighlight



Füllfarbe bei Status "hervorgehoben"

Typ intSiehe Liste, ObjekteIn 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 Oberflächenobjektyp 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
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeHyperLink

Beispiele:

```
tTheme->wpThemeColFillHover(_WinTypeHyperLink) # _WinColRed; tColor # tTheme->wpThemeColFillHover
```


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
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeRtfEdit

Beispiele:

```
tTheme->wpThemeColFillIndentNormal(_WinTypeRtfEdit) # _WinColRed; tColor # tTheme->wpThemeColFill
```

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
- WinTypeMenuButton
- WinTypeMenuItem
- WinTypeMetaPicture
- WinTypePicture
- WinTypeProgress
- WinTypePrtPreviewDlg
- WinTypeRadioButton
- WinTypeRTFEdit
- WinTypeScrollbar

- 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->wpThemeColFillM
```

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 Oberflächenobjektyp 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
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeCodeEdit

Beispiele:

```
tTheme->wpThemeColFillReadOnly(_WinTypeCodeEdit) # _WinColLightGray; tColor # tTheme->wpThemeColFi
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeRtfEdit

Beispiele:

```
tTheme->wpThemeColFillRulerNormal(_WinTypeRtfEdit) # _WinColRed; tColor # tTheme->wpThemeColFillRulerNormal
```

ThemeColFillSelectedActive



Füllfarbe bei Status "selektiert und aktiv"

Typ intSiehe Liste, ObjekteIn 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->wpTher
```


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
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeButton
- _WinTypeToolBarButton

Beispiele:

```
tTheme->wpThemeColFillShadeChecked(_WinTypeToolBarButton) # _WinColRed;tColor # tTheme->wpThemeCo
```

In dieser Eigenschaft können Farbkonstanten gesetzt werden.

- WinTypeButton
- WinTypeGroupBox
- WinTypeProgress
- WinTypeSlider
- WinTypeToolBarButton

```
tTheme->wpThemeColFillShadeDisabled( WinTypeToolBarButton) #   WinColRed;tColor # tTheme->wpThemeC
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- WinTypeButton
- WinTypeToolBarButton

Beispiele:

```
tTheme->wpThemeColFillShadeHighlight(_WinTypeToolBarButton) # _WinColRed;tColor # tTheme->wpTheme
```

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 Oberflächenobjektyp 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
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeButton
- _WinTypeToolBarButton

Beispiele:

```
tTheme->wpThemeColFillShadePressed(_WinTypeToolBarButton) # _WinColRed;tColor # tTheme->wpThemeCo
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeRtfEdit

Beispiele:

```
tTheme->wpThemeColFillTabNormal(_WinTypeRtfEdit) # _WinColRed;tColor # tTheme->wpThemeColFillTabM
```

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->wpThemeColFillThumbDisabled(_WinTypeSlider) # _WinColRed; tColor
```


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
```

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->wpThemeColFillTh
```


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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeHyperLink

Beispiele:

```
tTheme->wpThemeColFillVisited(_WinTypeHyperLink) # _WinColRed;tColor # tTheme->wpThemeColFillVisi
```

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 Oberflächenobjektyp 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->wpThemeC
```

Text Color (Farb-Eigenschaften für Textfarbe von Themes)

Eigenschaften

sortiert nach

Siehe Gruppen,
Alphabetische
Liste aller
Eigenschaften

- ThemeColTextActive
- ThemeColTextCaption
- ThemeColTextCaptionActive
- ThemeColTextCaptionInactive
- ThemeColTextChecked
- ThemeColTextDisabled
- ThemeColTextError
- ThemeColTextFocussed
- ThemeColTextHdrDisabled
- ThemeColTextHdrHighlight
- ThemeColTextHdrNormal
- ThemeColTextHdrPressed
- ThemeColTextHighlight
- ThemeColTextHover
- ThemeColTextInactive
- ThemeColTextNormal
- ThemeColTextPressed
- ThemeColTextRulerNormal
- ThemeColTextSelectedActive
- ThemeColTextSelectedInactive
- ThemeColTextVisited

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
```

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->wpThemeColTextCaption
```


ThemeColTextCaptionActive



Textfarbe der Titelzeile bei Status "aktiv"

Typ intSiehe Liste, ObjekteIn 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
```

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 Oberflächenobjekttyp 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
```

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
```

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 Oberflächenobjektyp 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
```

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 Objekthalt ungültig ist. 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:

- _WinTypeBigIntEdit
- _WinTypeColorEdit
- _WinTypeDateEdit
- _WinTypeDecimalEdit
- _WinTypeEdit
- _WinTypeFloatEdit
- _WinTypeFontNameEdit
- _WinTypeFontSizeEdit
- _WinTypeIntEdit
- _WinTypeTextEdit
- _WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColTextError(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeColTextError
```

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 Oberflächenobjektyp 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->wpThemeColTextFocussed(_WinTypeBigIntEdit) # tColor
```

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
```


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
```

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->wpThemeColTextH
```

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 Oberflächenobjekttyp 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
```

ThemeColTextHighlight



Textfarbe bei Status "hervorgehoben"

Typ intSiehe Liste, ObjekteIn 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 Oberflächenobjektyp 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
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeHyperLink

Beispiele:

```
tTheme->wpThemeColTextHover(_WinTypeHyperLink) # _WinColRed; tColor # tTheme->wpThemeColTextHover
```

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->wpThemeColTextInactive(_WinTypeDataListColumn) # _WinColRed;tColor
```

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(_
```

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
```


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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeRtfEdit

Beispiele:

```
tTheme->wpThemeColTextRulerNormal(_WinTypeRtfEdit) # _WinColRed;tColor # tTheme->wpThemeColTextR
```

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->wpThen
```

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
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeHyperLink

Beispiele:

```
tTheme->wpThemeColTextVisited(_WinTypeHyperLink) # _WinColRed;tColor # tTheme->wpThemeColTextVisi
```

Arrow Color (Farb-Eigenschaften für Pfeilfarbe von Themes)

Eigenschaften

sortiert nach

Siehe Gruppen,
Alphabetische
Liste aller
Eigenschaften

- ThemeColArrowChecked
- ThemeColArrowDisabled
- ThemeColArrowHighlight
- ThemeColArrowIvl
- ThemeColArrowNormal
- ThemeColArrowPressed
- ThemeColSortHdrDisabled
- ThemeColSortHdrHighlight
- ThemeColSortHdrNormal
- ThemeColSortHdrPressed

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
```

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->wpThemeColArrow
```

ThemeColArrowHighlight



Pfeilfarbe bei Status "hervorgehoben"

Typ intSiehe Liste, ObjekteIn 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->wpThemeColAr
```


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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeGanttGraph

Beispiele:

```
tTheme->wpThemeColArrowIvl(_WinTypeGanttGraph) # _WinColRed; tColor # tTheme->wpThemeColArrowIvl(
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeColorButton
- _WinTypeMenuButton
- _WinTypeToolBarButton



Der Objektyp _WinTypeToolBarButton bezieht sich auf die ColorButton- und MenuButton-Objekte mit StyleButton = _WinStyleButtonTBar.

Beispiele:

```
tTheme->wpThemeColArrowNormal(_WinTypeColorButton) # _WinColRed;tColor # tTheme->wpThemeColArrowNormal
```

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
```

ThemeColSortHdrDisabled

Farbe für Sortierrichtung bei Status DisabledTyp intSiehe Liste, ObjekteIn 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 Oberflächenobjektyp 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
```

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 Oberflächenobjektyp 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->wpThemeColSortHdrHighlight(_WinTypeRecList)
```

ThemeColSortHdrNormal



Farbe für Sortierrichtung bei Status "normal"

Typ intSiehe Liste, ObjekteIn 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->wpThemeColSortH
```

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 Oberflächenobjektyp 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
```

Border Color (Farb-Eigenschaften für Rahmenfarbe von Themes)

Eigenschaften

sortiert nach

Siehe Gruppen,
Alphabetische
Liste aller
Eigenschaften

- ThemeColBorderActive
- ThemeColBorderBase
- ThemeColBorderBoundary
- ThemeColBorderCellActiveHorz
- ThemeColBorderCellActiveVert
- ThemeColBorderCheckDisabled
- ThemeColBorderChecked
- ThemeColBorderCheckHighlight
- ThemeColBorderCheckNormal
- ThemeColBorderDisabled
- ThemeColBorderError
- ThemeColBorderFocussed
- ThemeColBorderHdrBase
- ThemeColBorderHighlight
- ThemeColBorderHover
- ThemeColBorderInactive
- ThemeColBorderIvl
- ThemeColBorderNormal
- ThemeColBorderPageNormal
- ThemeColBorderPressed
- ThemeColBorderReadOnly
- ThemeColBorderSelectedActive
- ThemeColBorderSelectedInactive
- ThemeColBorderVisited

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
```

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
```

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
```

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
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeGanttGraph

Beispiele:

```
tTheme->wpThemeColBorderCellActiveVert(_WinTypeGanttGraph) # _WinColRed; tColor # tTheme->wpThemeC
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeMenuItem

Beispiele:

```
tTheme->wpThemeColBorderCheckDisabled(_WinTypeMenuItem) # _WinColRed; tColor # tTheme->wpThemeColB
```

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
```

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
```


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
```

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 Oberflächenobjektyp 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

Beispiele:

```
tTheme->wpThemeColBorderDisabled(_WinTypeButton) # _WinColRed;tColor # tTheme->wpThemeColBorderD
```

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 Objekthinhalt ungültig ist. 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:

- _WinTypeBigIntEdit
- _WinTypeColorEdit
- _WinTypeDateEdit
- _WinTypeDecimalEdit
- _WinTypeEdit
- _WinTypeFloatEdit
- _WinTypeFontNameEdit
- _WinTypeFontSizeEdit
- _WinTypeIntEdit
- _WinTypeTextEdit
- _WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColBorderError(_WinTypeBigIntEdit) # _WinColRed; tColor # tTheme->wpThemeColBorderError
```

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
```

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->wpThemeColBorderH
```

ThemeColBorderHighlight



Rahmenfarbe bei Status "hervorgehoben"

Typ intSiehe Liste, ObjekteIn 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 Oberflächenobjektyp 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
```

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->wpThemeColBorderH
```


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
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeGanttGraph

Beispiele:

```
tTheme->wpThemeColBorderIvl(_WinTypeGanttGraph) # _WinColRed;tColor # tTheme->wpThemeColBorderIvl
```

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 Oberflächenobjektyp 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

- WinTypeRecList
- WinTypeRTFEdit
- WinTypeScrollbox
- WinTypeSlider
- WinTypeStoList
- WinTypeTextEdit
- WinTypeTimeEdit
- WinTypeTreeView
- WinTypeWebNavigator

Beispiele:

```
tTheme->wpThemeColBorderNormal(_WinTypeButton) # _WinColRed;tColor # tTheme->wpThemeColBorderNormal
```

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
```

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
```

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
```

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
```


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
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeHyperLink

Beispiele:

```
tTheme->wpThemeColBorderVisited(_WinTypeHyperLink) # _WinColRed;tColor # tTheme->wpThemeColBorder
```

Check Color (Farb-Eigenschaften für Markierungsfarbe von Themes)

Eigenschaften

sortiert nach

Siehe Gruppen,
Alphabetische
Liste aller
Eigenschaften

- ThemeColCheckActive
- ThemeColCheckDisabled
- ThemeColCheckHighlight
- ThemeColCheckInactive
- ThemeColCheckNormal
- ThemeColCheckPressed
- ThemeColCheckSelectedActive
- ThemeColCheckSelectedInactive

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
```

ThemeColCheckDisabled

Markierungsfarbe bei Status DisabledTyp intSiehe Liste, ObjekteIn 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->wpThemeColCheckD
```

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->wpThemeColCheck
```

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->wpThemeCol
```

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->wpThemeColCheckNormal
```


ThemeColCheckPressed



Markierungsfarbe bei Status "gedrückt"

Typ intSiehe Liste, ObjekteIn 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
```

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
```

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->wp1
```

Separator Color (Farb-Eigenschaften für Separatorfarbe von Themes)

Eigenschaften

sortiert nach

Siehe Gruppen,
Alphabetische
Liste aller
Eigenschaften

- ThemeColSeparator
- ThemeColSeparatorClmBase
- ThemeColSeparatorClmFixedBase
- ThemeColSeparatorHdrBase
- ThemeColSeparatorLineBase

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
```

ThemeColSeparatorClnBase ☐

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->wpThemeColSeparatorClnBase(_WinTypeRecList) # _WinColLightBlue;tColor # tTheme->wpThemeCo
```

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 Oberflächenobjektyp 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
```

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) # _WinCollLightGray;tColor # tTheme->wpThemeCo
```


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
```

Splitter Color (Farb-Eigenschaften für Splitterfarbe von Themes)

Eigenschaften

sortiert nach

Siehe Gruppen,
Alphabetische
Liste aller
Eigenschaften

- ThemeColFillSplitterDisabled
- ThemeColFillSplitterNormal
- ThemeColForeSplitterDisabled
- ThemeColForeSplitterNormal

ThemeColFillSplitterDisabled

Füllfarbe des Splitters bei Status DisabledTyp intSiehe Liste, ObjekteIn 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 Oberflächenobjektyp 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
```

ThemeColFillSplitterNormal



Füllfarbe des Splitters bei Status "normal"

Typ intSiehe Liste, ObjekteIn 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 Oberflächenobjektyp 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
```

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
```

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 Oberflächenobjektyp 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
```

Grip Color (Farb-Eigenschaften für Ziehpunktfarbe von Themes)

Eigenschaften
sortiert nach

Siehe Gruppen,
Alphabetische
Liste aller
Eigenschaften

- ThemeColGrip

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(_WinTypeT
```


Border Size (Größen-Eigenschaften für Rahmen von Themes)

Eigenschaften

sortiert nach

Siehe Gruppen,
Alphabetische
Liste aller
Eigenschaften

- 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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeButton
- _WinTypeColorButton
- _WinTypeMenuButton

Beispiele:

```
tTheme->wpThemeSizeBorderChecked(_WinTypeButton) # 2;tSize # tTheme->wpThemeSizeBorderChecked(_W
```

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(_W
```

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(_
```

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 Oberflächenobjektyp 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
```

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(_W
```

Responsive Size (Größen-Eigenschaften von Themes)

Eigenschaften
sortiert nach

Siehe Gruppen,
Alphabetische
Liste aller
Eigenschaften

- 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);
```

Spin Color (Farb-Eigenschaften für Spin-Bereich von Themes)

Eigenschaften

sortiert nach

Siehe Gruppen,
Alphabetische
Liste aller
Eigenschaften

- ThemeColFillSpinArrowDisabled
- ThemeColFillSpinArrowError
- ThemeColFillSpinArrowHighlight
- ThemeColFillSpinArrowNormal
- ThemeColFillSpinArrowPressed
- ThemeColFillSpinDisabled
- ThemeColFillSpinError
- ThemeColFillSpinHighlight
- ThemeColFillSpinNormal
- ThemeColFillSpinPressed

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeBigIntEdit
- _WinTypeIntEdit
- _WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColFillSpinArrowDisabled(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpTheme
```

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 Objekthinhalte 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
```

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->wpTher
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeBigIntEdit
- _WinTypeIntEdit
- _WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColFillSpinArrowNormal(_WinTypeBigIntEdit) # _WinColRed; tColor # tTheme->wpThemeCo
```

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
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeBigIntEdit
- _WinTypeIntEdit
- _WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColFillSpinDisabled(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeColF
```


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 Objekthinhalt ungültig ist. 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:

- _WinTypeBigIntEdit
- _WinTypeIntEdit
- _WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColFillSpinError(_WinTypeBigIntEdit) # _WinColRed; tColor # tTheme->wpThemeColFillS
```

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
```

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
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeBigIntEdit
- _WinTypeIntEdit
- _WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColFillSpinPressed(_WinTypeBigIntEdit) # _WinColRed; tColor # tTheme->wpThemeColFillSpinPressed(_WinTypeIntEdit) # _WinColRed; tColor # tTheme->wpThemeColFillSpinPressed(_WinTypeTimeEdit) # _WinColRed; tColor
```

Popup button (Farb-Eigenschaften für Popup-Button von Themes)

Eigenschaften

sortiert nach

Siehe Gruppen,
Alphabetische
Liste aller
Eigenschaften

- ThemeColArrowDropDisabled
- ThemeColArrowDropError
- ThemeColArrowDropHighlight
- ThemeColArrowDropNormal
- ThemeColArrowDropPressed
- ThemeColFillDropDisabled
- ThemeColFillDropError
- ThemeColFillDropHighlight
- ThemeColFillDropNormal
- ThemeColFillDropPressed

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 Oberflächenobjektyp 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->wpThemeColArrowDrowDisabled(_WinTypeBigIntEdit) # _WinColRed; tColor # tTheme->wpThemeColA
```

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 Objekteinhalt ungültig ist. 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:

- _WinTypeBarcode
- _WinTypeBigintEdit
- _WinTypeColorEdit
- _WinTypeDateEdit
- _WinTypeDecimalEdit
- _WinTypeEdit
- _WinTypeFloatEdit
- _WinTypeFontNameEdit
- _WinTypeFontSizeEdit
- _WinTypeIntEdit
- _WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColArrowDropError(_WinTypeBarcode) # _WinColRed; tColor # tTheme->wpThemeColArrowD
```

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 Oberflächenobjektyp 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
```


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 Oberflächenobjektyp 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->wpThemeColAr
```

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 Oberflächenobjektyp 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->wpThemeColA
```

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 Oberflächenobjektyp 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
```

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 Objekteinhalt ungültig ist. 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:

- WinTypeBigintEdit
- WinTypeColorEdit
- WinTypeDateEdit
- WinTypeDecimalEdit
- WinTypeEdit
- WinTypeFloatEdit
- WinTypeFontNameEdit
- WinTypeFontSizeEdit
- WinTypeIntEdit
- WinTypeTimeEdit

Beispiele:

```
tTheme->wpThemeColFillDropError(_WinTypeBigIntEdit) # _WinColRed;tColor # tTheme->wpThemeColFillD
```

ThemeColFillDropHighlight



Füllfarbe bei Status "hervorgehoben"

Typ intSiehe Liste, ObjekteIn 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 Oberflächenobjektyp 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->wpThemeColFillDropHighlight(_WinTypeBigIntEdit) # tColor
```

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 Oberflächenobjektyp 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
```

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->wpThemeColFill
```

Scroll button (Farb-Eigenschaften für Scroll-Button von Themes)

Eigenschaften

sortiert nach

Siehe Gruppen,
Alphabetische
Liste aller
Eigenschaften

- ThemeColArrowScrollHighlight
- ThemeColArrowScrollNormal
- ThemeColArrowScrollPressed
- ThemeColFillScrollHighlight
- ThemeColFillScrollNormal
- ThemeColFillScrollPressed

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeNotebook

Beispiele:

```
tTheme->wpThemeColArrowScrollHighlight(_WinTypeNotebook) # _WinColRed;tColor # tTheme->wpThemeCol
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeNotebook

Beispiele:

```
tTheme->wpThemeColArrowScrollNormal(_WinTypeNotebook) # _WinColRed; tColor # tTheme->wpThemeColAr
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeNotebook

Beispiele:

```
tTheme->wpThemeColArrowScrollPressed(_WinTypeNotebook) # _WinColRed; tColor # tTheme->wpThemeColA
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeNotebook

Beispiele:

```
tTheme->wpThemeColFillScrollHighlight(_WinTypeNotebook) # _WinColRed; tColor # tTheme->wpThemeColF
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeNotebook

Beispiele:

```
tTheme->wpThemeColFillScrollNormal(_WinTypeNotebook) # _WinColRed;tColor # tTheme->wpThemeColFill
```

ThemeColFillScrollPressed	<input type="checkbox"/>
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->wpThemeColFillScrollPressed(_WinTypeNotebook) # _WinColRed;tColor # tTheme->wpThemeColFil
```

Editor Color (Farb-Eigenschaften für CodeEdit-Objekt von Themes)

Eigenschaften

sortiert nach

Siehe Gruppen,
Alphabetische
Liste aller
Eigenschaften

- ThemeColEditorAlpha
- ThemeColEditorCKeywords
- ThemeColEditorComment
- ThemeColEditorConstants
- ThemeColEditorCurrentLine
- ThemeColEditorCustomHighlight
- ThemeColEditorDataTypes
- ThemeColEditorFunctions
- ThemeColEditorKeywords
- ThemeColEditorNumber
- ThemeColEditorOccurrences
- ThemeColEditorPreProcessor
- ThemeColEditorSelection
- ThemeColEditorText

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->wpThemeColEditorAlpha
```


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
```

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->wpThemeColEditorComment(_WinTypeCodeEdit)
```

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
```

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 Oberflächenobjektyp 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
```

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
```

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
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeCodeEdit

Beispiele:

```
tTheme->wpThemeColEditorFunctions(_WinTypeCodeEdit) # _WinColBlue; tColor # tTheme->wpThemeColEdit
```


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
```

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 Oberflächenobjektyp 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
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeCodeEdit

Beispiele:

```
tTheme->wpThemeColEditorPreProcessor(_WinTypeCodeEdit) # _WinColBlue;tColor # tTheme->wpThemeColE
```

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 Oberflächenobjektyp angegeben werden, auf den sich der Aufruf bezieht. Folgende Typen können angegeben werden:

- _WinTypeCodeEdit

Beispiele:

```
tTheme->wpThemeColEditorSelection(_WinTypeCodeEdit) # _WinCollightGray;tColor # tTheme->wpThemeCo
```

ThemeColEditorText



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->wpThemeColEditorText(_WinTypeCodeEdit) # _WinColBlack; tColor # tTheme->wpThemeColEditorText
```

Image Color (Farb-Eigenschaften für Bilder von Themes)

Eigenschaften
sortiert nach

Siehe Gruppen,
Alphabetische
Liste aller
Eigenschaften

- ThemeColImageNormal

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
```

Tick Color (Farb-Eigenschaften für Markierungsfarbe von Themes)

Eigenschaften

sortiert nach

Siehe Gruppen,
Alphabetische
Liste aller
Eigenschaften

- ThemeColTickDisabled
- ThemeColTickNormal

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
```

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(_
```

Ermitteln und Setzen von Eigenschaften
Aufteilung der Eigenschaften in Gruppen

WinPropGet(),
WinPropSet(),
PrtPropGet(),
PrtPropSet(),
Siehe 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 eines der 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.

Liste aller Eigenschaften von Oberflächen- und Druck-Objekten
In dieser Liste sind alle Eigenschaften von Oberflächen- und Druck-Objekten
aufgeführt.

Liste

Siehe sortiert
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
- DbName
- DbUserID
- DbUserName
- 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](#)
- [HyphMarginTwips](#)

- [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
- MenuId
- 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
- ThemeMenuTileSize
- ThemeName
- ThemeOptFillShade
- ThemeSetId
- ThemeSizeBorderChecked
- ThemeSizeBorderDefault
- ThemeSizeBorderDisabled
- ThemeSizeBorderHighlight
- ThemeSizeBorderNormal

- ThemeSizeBorderPressed
- ThemeSizeFixed
- TickAlignment
- TickInterval
- TickLineSize
- TickPageSize
- TifFileName
- 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

- WaitCursor
- Width
- WidthPen
- WordBreak

- ZoomEffective
- ZoomFactor
- ZOrder

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

- UrmPropPassword
- UrmPropPwdCapitals
- UrmPropPwdDigits
- UrmPropPwdLocked
- UrmPropPwdMinLength
- UrmPropPwdSpecials
- UrmPropShortcut
- UrmPropStartProc
- UrmPropUserGroup

_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".

_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".

_UrmOldPropConfiguration ☐

Programmberechtigung Konfiguration

Typ logic

Siehe Liste, Konvertierung des alten
Benutzersystems, Benutzerpflege

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten. Die Eigenschaft entspricht der Programmberechtigung "Konfiguration".

_UrmOldPropDbInfo ☐
Programmberechtigung DB-Info im Textsystem
Typ logic

Siehe Liste, Konvertierung des alten
Benutzersystems, Benutzerpflege

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten. Die Eigenschaft entspricht der Programmberechtigung "DB-Info im Textsystem".

_UrmOldPropDbMaintenance ☐
Programmberechtigung Datenbankpflege
Typ logic

Siehe Liste, Konvertierung des alten
Benutzersystems, Benutzerpflege

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
Benutzersystems, Benutzerpflege

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten. Die Eigenschaft entspricht der Programmberechtigung "Basisparameter".

_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".

_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
Benutzersystems, Benutzerpflege

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten. Die Eigenschaft entspricht der Programmberechtigung "Druckertreiber".

_UrmOldPropExtAccess ☐
Programmberechtigung Externer Zugriff
Typ logic

Siehe Liste, Konvertierung des alten
Benutzersystems, Benutzerpflege

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
- DbConnect() aus Standard- und Advanced-Client

Der Zugriff kann für einzelne Komponenten über Erweiterte Eigenschaften bei der Hauptbenutzergruppe deaktiviert werden.

_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.

_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".

_UrmOldPropLoadDbDefinition ☐

Programmberechtigung Datensicherung

Typ logic

Siehe Liste, Konvertierung des alten
Benutzersystems, Benutzerpflege

Diese Eigenschaft besteht, um die Kompatibilität zum alten Berechtigungssystem zu gewährleisten. Die Eigenschaft entspricht der Programmberechtigung "Datensicherung".

_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".

_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 Programmberechtigung "Berechtigung Menüs".

_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".

_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".

_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".

_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".

_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 Programmberechtigung "Berechtigung Druckertreiber".

_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".

_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".

_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".

_UrmOldPropTestFunctions ☐
Programmberechtigung Testfunktionen
Typ logic

Siehe Liste, Konvertierung des alten
Benutzersystems, Benutzerpflege

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.

_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 Programmberechtigung "Berechtigung Textsystem".

_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".

_UrmPropActive ☐

Objekt ist aktiv

Typ logic

Siehe Liste,
Objekte

Über diese Eigenschaft kann ein Benutzer der Benutzerpflege vorübergehend deaktiviert werden. Ein deaktivierter 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 Neuansmeldung wirksam.

_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.

_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.

_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.

_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
```

_UrmPropExpirationDate ☐

Gültigkeitsdatum

Typ date

Siehe Liste, Objekte,
Benutzerpflege

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.

_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.

_UrmPropID ☐
ID des Objekts
Typ int
Siehe Liste,
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.

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.

_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.

_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.

_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.

_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:

! * ? : ; / ' " \

_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.

_UrmPropOwnerGroup ☐

Besitzergruppe

Typ alpha(20)

Siehe Liste, Objekte,

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.

_UrmPropPassword ☐
Kennwort des Benutzers

Typ alpha(40)

Siehe Liste, Objekte,
Benutzerpflege

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
```

_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.

_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.

_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.

_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 Kennwortes 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.

_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.

_UrmPropStartProc ☐
Prozedur bei Start
Typ alpha(20)
Siehe Liste, Benutzerpflege,
_UrmOldPropInitProc

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.

_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.

Ermitteln und Setzen von Eigenschaften
Aufteilung der Eigenschaften in Gruppen

WinPropGet(),
WinPropSet(),
PrtPropGet(),
PrtPropSet(),
Siehe 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 eines der 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.

Eigenschaften von System-Objekten

In dieser Liste sind alle Eigenschaften von System-Objekten aufgeführt.

Liste sortiert

nach Gruppen,

Siehe Alphabetische

Liste aller

Eigenschaften

- AttribCount
- BaseAddress
- CertificateError
- Charset
- CharsetOS
- ChartArea
- ChartBorderWidth
- ChartColBkg
- ChartColBorder
- ChartHeight
- ChartLegendColBkg
- ChartLegendColBorder
- ChartLegendKeyGap
- ChartLegendKeySize
- ChartLegendPos
- ChartNumFmtThousandSep
- ChartPieAngle
- ChartPieColEdge
- ChartPieColJoin
- ChartPieDepth
- ChartPieInnerRadius
- ChartPieLabelFont
- ChartPieShading
- ChartPieShadowMode
- ChartPieStartAngle
- ChartPyramidElevation
- ChartPyramidLayerGap
- ChartPyramidRotation
- ChartPyramidStyleData
- ChartPyramidTubeHeight
- ChartPyramidTubeRadius
- ChartPyramidTwist
- ChartSurfaceAmbient
- ChartSurfaceColContour
- ChartSurfaceColSurfaceAxis
- ChartSurfaceColSurfaceData
- ChartSurfaceContinuous
- ChartSurfaceDiffuse
- ChartSurfaceElevation
- ChartSurfaceLegendLength
- ChartSurfacePerspective
- ChartSurfacePlotHeight
- ChartSurfaceRotation

- [ChartSurfaceSamplesX](#)
- [ChartSurfaceSamplesY](#)
- [ChartSurfaceShadingMode](#)
- [ChartSurfaceShine](#)
- [ChartSurfaceSmooth](#)
- [ChartSurfaceSpecular](#)
- [ChartSurfaceTitleFontX](#)
- [ChartSurfaceTitleFontY](#)
- [ChartSurfaceTitleFontZ](#)
- [ChartSurfaceTitleX](#)
- [ChartSurfaceTitleY](#)
- [ChartSurfaceTitleZ](#)
- [ChartSurfaceTwist](#)
- [ChartSurfaceWireWidth](#)
- [ChartTitleArea](#)
- [ChartTitleColBkg](#)
- [ChartTitleColFg](#)
- [ChartTitleFont](#)
- [ChartTitleText](#)
- [ChartWidth](#)
- [ChartXYAxisColY](#)
- [ChartXYAxisOffsetY](#)
- [ChartXYAxisTitleAlignY](#)
- [ChartXYAxisTitleY](#)
- [ChartXYAxisY](#)
- [ChartXYBarGap](#)
- [ChartXYBarShading](#)
- [ChartXYBarShape](#)
- [ChartXYColBkg](#)
- [ChartXYColBkgAlt](#)
- [ChartXYColBorder](#)
- [ChartXYColData](#)
- [ChartXYColGridX](#)
- [ChartXYColGridY](#)
- [ChartXYColTrend](#)
- [ChartXYDepth](#)
- [ChartXYDepthGap](#)
- [ChartXYGapDash](#)
- [ChartXYLabelAngleX](#)
- [ChartXYLabelColData](#)
- [ChartXYLabelColSum](#)
- [ChartXYLabelFontData](#)
- [ChartXYLabelFontX](#)
- [ChartXYLabelFontY](#)
- [ChartXYLabelRotData](#)
- [ChartXYLegendText](#)
- [ChartXYLineSymbol](#)
- [ChartXYLineSymbolParam](#)
- [ChartXYLineSymbolSize](#)
- [ChartXYLineWidth](#)
- [ChartXYMinTickIncY](#)

- ChartXYStyleData
- ChartXYStyleLabel
- ChartXYSwapXY
- ChartXYTitleAlignY
- ChartXYTitleFontX
- ChartXYTitleFontY
- ChartXYTitleX
- ChartXYTitleY
- ChartXYTrendDegree
- ChartXYTrendType
- ChildCount
- CodepageOS
- Compression
- ContentLength
- CpnSetupFlagsCln
- CpnSetupFlagsDoc
- CpnSetupFlagsSrv
- Created
- CreatedUser
- CteNodeSepAttrib
- CteNodeSepPath
- Custom
- DisplayRaisingDelay
- ErrCode
- ErrLine
- ErrPos
- ErrProc
- ErrSource
- ErrSourceLine
- ErrText
- FileFilter
- FileFilterNum
- FileNameExt
- Flags
- FsiError
- FullName
- Hash
- HdlCount
- HostName
- HTTPHeader
- HttpParameters
- ID
- JobData
- JobErrorCode
- JobID
- JobMsxReadQ
- JobMsxWriteQ
- JobProcesses
- JobProcExtended
- JobSckHandle
- JobStatus

- JobThreads
- LclCurrDecimal
- LclCurrFract
- LclCurrFractIntl
- LclCurrGroup
- LclCurrNegSignPos
- LclCurrNegSpaceSep
- LclCurrNegSymPrec
- LclCurrPosSignPos
- LclCurrPosSpaceSep
- LclCurrPosSymPrec
- LclCurrSymbol
- LclCurrSymbolIntl
- LclCurrTSep
- LclDateDayLZero
- LclDateDayN
- LclDateDayNS
- LclDateLFormat
- LclDateLOrder
- LclDateMonthLZero
- LclDateMonthN
- LclDateMonthNS
- LclDateSCentury
- LclDateSep
- LclDateSFormat
- LclDateSOrder
- LclNumDecimal
- LclNumFract
- LclNumGroup
- LclNumLZero
- LclNumNegMode
- LclNumNegSign
- LclNumPosSign
- LclNumTSep
- LclTimeHourMode
- LclTimeLFormat
- LclTimeLZero
- LclTimeMarkPos
- LclTimeSep
- LclTimeSepAM
- LclTimeSepPM
- LclTimeSFormat
- Len
- LogicalProcessors
- Method
- Modified
- ModifiedUser
- Name
- ObjectsGDI
- ObjectsGDILimit
- ObjectsUser

- ObjectsUserLimit
- OdbcClmBufferLen
- OdbcClmCatalog
- OdbcClmDataDefault
- OdbcClmDataType
- OdbcClmDataTypeName
- OdbcClmDecimalDigits
- OdbcClmName
- OdbcClmNullable
- OdbcClmNumPrecRadix
- OdbcClmOrdinal
- OdbcClmRemarks
- OdbcClmSchema
- OdbcClmSize
- OdbcClmTable
- OdbcConCfcOdbcApi
- OdbcConCfcOdbcSql
- OdbcConCfcSagCli
- OdbcConDatabaseName
- OdbcConDbmsName
- OdbcConDbmsVersion
- OdbcConDriverName
- OdbcConDriverOdbcVersion
- OdbcConDriverVersion
- OdbcConDSN
- OdbcConOdbcVersion
- OdbcConReadOnly
- OdbcConServerName
- OdbcConUserName
- OdbcErrSqlMessage
- OdbcErrSqlNativeCode
- OdbcErrSqlResult
- OdbcErrSqlState
- OdbcResCountClm
- OdbcResCountRow
- OdbcTblCatalog
- OdbcTblName
- OdbcTblRemarks
- OdbcTblSchema
- OdbcTblType
- Parent
- PathAllAppData
- PathAllAutoStart
- PathAllDesktop
- PathAllPrograms
- PathAppData
- PathAppProg
- PathAppRoot
- PathAutoStart
- PathClient
- PathDesktop

- PathFonts
- PathMyDocuments
- PathMyMusic
- PathMyPictures
- PathMyVideo
- PathOffcAccess
- PathOffcExcel
- PathOffcOutlook
- PathOffcPowerPoint
- PathOffcWord
- PathPfmData
- PathPfmRoot
- PathProgramFiles
- PathProgramFilesCommon
- PathPrograms
- PathSystem
- PathTemp
- PathWindows
- PdfAttachCount
- PdfAuthor
- PdfCompany
- PdfCreator
- PdfKeywords
- PdfPageCount
- PdfPageHeight
- PdfPageRotation
- PdfPageWidth
- PdfPrintOrder
- PdfProducer
- PdfResolution
- PdfSubject
- PdfTitle
- PdfUserRights
- PdfVersion
- PhysicalMemoryMB
- Platform
- PriorityIdle
- ProcCacheKB
- ProcCacheLimitKB
- ProcCurrent
- ProcCurrentFull
- ProcCurrentSub
- ProcessArchitecture
- ProcessExitCode
- ProcessID
- ProcessMemory
- ProcessMemoryKB
- ProcessMemoryLimitMB
- Protocol
- ProxyAuthorization
- ProxyAuthType

- Size
- SizeDb
- SizeDb64
- SizeOrg
- SizeOrg64
- StatusCode
- StopRequest
- StorageID
- SvcCallDelay
- SvcCallTime
- SvcDescription
- SvcName
- SvcSckHandle
- SvcSessionID
- TapiFlags
- TapiModuleFilename
- TapiNameComputer
- TapiOwnerRequest
- TapiUserName
- TerminalSession
- TerminalSessionID
- TimeExternal
- Type
- TypeMime
- TypeUser
- URI
- Validated
- ValidatedUser
- ValueAlpha
- ValueBigInt
- ValueCaltime
- ValueColor
- ValueDate
- ValueDecimal
- ValueEvent
- ValueFloat
- ValueFont
- ValueInt
- ValueLogic
- ValuePoint
- ValueRange
- ValueRect
- ValueRtfTab
- ValueTime
- Version

Eigenschaften von System-Objekten - A bis E

In dieser Liste sind alle Eigenschaften von A bis E von System-Objekten aufgeführt.

Liste sortiert

nach Gruppen,

Siehe Alphabetische

Liste aller

Eigenschaften

- AttribCount
- BaseAddress
- CertificateError
- Charset
- CharsetOS
- ChartArea
- ChartBorderWidth
- ChartColBkg
- ChartColBorder
- ChartHeight
- ChartLegendColBkg
- ChartLegendColBorder
- ChartLegendKeyGap
- ChartLegendKeySize
- ChartLegendPos
- ChartNumFmtThousandSep
- ChartPieAngle
- ChartPieColEdge
- ChartPieColJoin
- ChartPieDepth
- ChartPieInnerRadius
- ChartPieLabelFont
- ChartPieShading
- ChartPieShadowMode
- ChartPieStartAngle
- ChartPyramidElevation
- ChartPyramidLayerGap
- ChartPyramidRotation
- ChartPyramidStyleData
- ChartPyramidTubeHeight
- ChartPyramidTubeRadius
- ChartPyramidTwist
- ChartSurfaceAmbient
- ChartSurfaceColContour
- ChartSurfaceColSurfaceAxis
- ChartSurfaceColSurfaceData
- ChartSurfaceContinuous
- ChartSurfaceDiffuse
- ChartSurfaceElevation
- ChartSurfaceLegendLength
- ChartSurfacePerspective
- ChartSurfacePlotHeight
- ChartSurfaceRotation

- [ChartSurfaceSamplesX](#)
- [ChartSurfaceSamplesY](#)
- [ChartSurfaceShadingMode](#)
- [ChartSurfaceShine](#)
- [ChartSurfaceSmooth](#)
- [ChartSurfaceSpecular](#)
- [ChartSurfaceTitleFontX](#)
- [ChartSurfaceTitleFontY](#)
- [ChartSurfaceTitleFontZ](#)
- [ChartSurfaceTitleX](#)
- [ChartSurfaceTitleY](#)
- [ChartSurfaceTitleZ](#)
- [ChartSurfaceTwist](#)
- [ChartSurfaceWireWidth](#)
- [ChartTitleArea](#)
- [ChartTitleColBkg](#)
- [ChartTitleColFg](#)
- [ChartTitleFont](#)
- [ChartTitleText](#)
- [ChartWidth](#)
- [ChartXYAxisColY](#)
- [ChartXYAxisOffsetY](#)
- [ChartXYAxisTitleAlignY](#)
- [ChartXYAxisTitleY](#)
- [ChartXYAxisY](#)
- [ChartXYBarGap](#)
- [ChartXYBarShading](#)
- [ChartXYBarShape](#)
- [ChartXYColBkg](#)
- [ChartXYColBkgAlt](#)
- [ChartXYColBorder](#)
- [ChartXYColData](#)
- [ChartXYColGridX](#)
- [ChartXYColGridY](#)
- [ChartXYColTrend](#)
- [ChartXYDepth](#)
- [ChartXYDepthGap](#)
- [ChartXYGapDash](#)
- [ChartXYLabelAngleX](#)
- [ChartXYLabelColData](#)
- [ChartXYLabelColSum](#)
- [ChartXYLabelFontData](#)
- [ChartXYLabelFontX](#)
- [ChartXYLabelFontY](#)
- [ChartXYLabelRotData](#)
- [ChartXYLegendText](#)
- [ChartXYLineSymbol](#)
- [ChartXYLineSymbolParam](#)
- [ChartXYLineSymbolSize](#)
- [ChartXYLineWidth](#)
- [ChartXYMinTickIncY](#)

- ChartXYStyleData
- ChartXYStyleLabel
- ChartXYSwapXY
- ChartXYTitleAlignY
- ChartXYTitleFontX
- ChartXYTitleFontY
- ChartXYTitleX
- ChartXYTitleY
- ChartXYTrendDegree
- ChartXYTrendType
- ChildCount
- CodepageOS
- Compression
- ContentLength
- CpnSetupFlagsCln
- CpnSetupFlagsDoc
- CpnSetupFlagsSrv
- Created
- CreatedUser
- CteNodeSepAttrib
- CteNodeSepPath
- Custom
- DisplayRaisingDelay
- ErrCode
- ErrLine
- ErrPos
- ErrProc
- ErrSource
- ErrSourceLine
- ErrText

AttribCount



Anzahl der Attribute bei CteNode-Objekten

Typ int

Siehe Liste, Objekte, ChildCount

Mit dieser Eigenschaft kann die Anzahl der Attribute eines CteNode-Objekts ermittelt werden.

```
// Anzahl der Attributemente ermitteltCount # tNode->spAttribCount;
```


BaseAddress ☐

Adresse des Speicherbereichs

Typ bigint

Siehe Liste, SysPropGet()

Dieser Wert ist die Adresse des Speicherbereichs des Memory-Objekts im aktuellen Prozess. Bei 32-Bit-Systemen werden nur die unteren 32 Bit des Wertes verwendet. Der Wert kann beispielsweise an eine DLL übergeben werden, um einen direkten Zugriff auf den Speicherinhalt durchzuführen.

Die Eigenschaft kann nur gelesen werden.

CertificateError



Fehlerwert bei der Zertifikatsüberprüfung

Typ int

Siehe Liste, Objekte

In dieser Eigenschaft kann der Fehlerwert, der bei einer fehlgeschlagenen Zertifikatsüberprüfung aufgetreten ist, abgefragt werden. Der Fehlerwert kann mit der Anweisung ErrMapText() in eine Fehlermeldung gewandelt werden.

Beispiel:

```
tErrCode # _Sys->spCertificateError;tErrMsg # ErrMapText(tErrCode, 'EN', _ErrMapX509);
```

Charset ☐

Verwendeter Zeichensatz

Typ int

Siehe Liste, Objekte

Diese Eigenschaft steht bei den Memory-Objekten, PrintJob-Objekten und dem System-Objekt zur Verfügung.

Beim System-Objekt kann über die Eigenschaft die von der Applikation verwendete Zeichensatz ermittelt werden. Das Resultat kann mit den unten stehenden Konstanten verglichen werden. In diesem Objekt kann die Eigenschaft nicht gesetzt werden.

Enthält ein Memory-Objekt alphanumerische Daten, bestimmt diese Eigenschaft den zugehörigen Zeichensatz. Dieser Wert wird bei den Funktionen MemFindStr(), MemReadStr(), MemWriteStr() und MemCnv() zur Umwandlung der Zeichen verwendet.

Beim PrintJob-Objekt bestimmt die Eigenschaft den Zeichensatz, in dem die XML-Datei erzeugt wird, wenn beim PrtJobClose() _PrtJobXml angegeben wird.

Folgende Werte sind definiert (siehe auch Zeichensätze):

- _CharsetC16_...

Dies sind die 8-Bit-Zeichensätze, in denen alphanumerische Daten innerhalb von CONZEPT 16 gespeichert werden. Es gibt drei Varianten für die Codepages 1250 (osteuropäisch), 1252 (westeuropäisch) und 1254 (türkisch). Die verwendete Variante wird anhand der der aktiven Codepage oder der eingestellten Sprache in CONZEPT 16 ausgewählt.

- _CharsetC16_1250

Dies ist der interne Zeichensatz in der Variante für osteuropäische Sprachen. Alle Zeichen der Windows Codepage 1250 sind hier definiert.

- _CharsetC16_1252

Dies ist der interne Zeichensatz in der Variante für westeuropäische Sprachen. Alle Zeichen der Windows Codepage 1252 sind hier definiert.

- _CharsetC16_1254

Dies ist der interne Zeichensatz in der Variante für die türkische Sprache. Alle Zeichen der Windows Codepage 1254 sind hier definiert.

- _CharsetWCP_1250

Dies ist der Windows-Zeichensatz für osteuropäische Sprachen (Codepage 1250).

- _CharsetWCP_1252

Dies ist der Windows-Zeichensatz für westeuropäische Sprachen (Codepage 1252).

- _CharsetWCP_1254

Dies ist der Windows-Zeichensatz für die türkische Sprachen (Codepage 1254).

- `_CharsetUndefined`

Die aktuelle Codepage wird nicht unterstützt oder es wurde der Zeichensatz eines Linux-Betriebssystems abgefragt.

- `_CharsetOEM`

Dies ist der ursprüngliche Zeichensatz des IBM PCs (IBM 437), der auch unter DOS verwendet wurde.

- `_CharsetOEM_852`

Dies ist der DOS-Zeichensatz für zentraleuropäische Sprachen (Codepage 852). Er enthält alle darstellbaren Zeichen aus ISO 8859-2.

- `_CharsetISO_8859_1`

Dies ist der 8-Bit ISO-Zeichensatz für westeuropäische Sprachen (Latin-1).

- `_CharsetISO_8859_2`

Dies ist der 8-Bit ISO-Zeichensatz für osteuropäische Sprachen (Latin-2).

- `_CharsetISO_8859_9`

Dies ist der 8-Bit ISO-Zeichensatz für die türkische Sprache (Latin-5).

- `_CharsetISO_8859_15`

Dies ist ein erweiterter 8-Bit ISO-Zeichensatz für westeuropäische Sprachen (Latin-9), der unter anderem auch das Euro-Zeichen umfasst.

- `_CharsetHTML`

Dieser Zeichensatz basiert auf Unicode 5.1, wobei alle Zeichen ab Codeposition 128 sowie die Zeichen `&`, `<`, `>`, `"` in der Form `&#...;` im Text gespeichert sind.

- `_CharsetUTF8`

Dieser Zeichensatz basiert auf Unicode 5.1, wobei die einzelnen Zeichen im UTF-8-Format gespeichert sind. Ein einzelnes Zeichen kann dabei 1 bis 4 Bytes belegen.

- `_CharsetUTF16`

Dieser Zeichensatz basiert auf Unicode 5.1, wobei die einzelnen Zeichen im UTF-16-Format gespeichert sind. Ein einzelnes Zeichen kann dabei 2 oder 4 Bytes belegen. Dieser Wert kann zusätzlich mit zwei weiteren Konstanten kombiniert werden:

- ◆ `_UTFxBE`

Der UTF-16-Text ist im Big-Endian-Format abgelegt. Standardmäßig wird der Text im Little-Endian-Format verarbeitet.

- ◆ `_UTFxLE`

Der UTF-16-Text ist im Little-Endian-Format abgelegt. Dies ist der Standardwert für UTF-16.

- ◆ `_UTFxBOM`

Kontakt

Beim Umwandeln in einen UTF-16-Text wird eine Byte-Order-Mark vorangestellt, der das verwendete Endian-Format enthält. Die Konstante ist nur in Kombination mit `_UTFxBE` und `_UTFxLE` sinnvoll.

- `_CharsetURI`

Dieser Zeichensatz basiert auf Unicode in UTF-8-Kodierung, wobei alle Bytes außer A-Z, a-z, 0-9, Punkt, Bindestrich, Unterstrich und Tilde in hexadezimaler Notation mit vorangestelltem Prozentzeichen dargestellt werden.

CharsetOS



Zeichensatz des Betriebssystems

Typ int

Siehe Liste, Objekte, SysPropGet()

Bei dem _Sys-Objekt kann über diese Eigenschaft der Zeichensatz des Windows-Betriebssystems ermittelt werden. Der Rückgabewert kann mit den _CharsetWCP...-Konstanten verglichen werden. Wird die aktuelle Codepage des Betriebssystems nicht unterstützt oder der Zeichensatz eines Linux-Betriebssystems abgefragt, gibt die Eigenschaft _CharsetUndefined zurück.

ChartArea



Bereich der Grafik innerhalb des Chart-Objekts

Typ rect

Siehe Liste, Chart

Diese Eigenschaft definiert den Bereich der Grafik innerhalb des bei ChartOpen() angegebenen Ausgabebereichs. Außerhalb der Grafik kann so zum Beispiel Platz für die Legende geschaffen werden (siehe ChartLegendPos).

In dem angegebenen Bereich wird nur die eigentliche Grafik gezeichnet. Alle Bezeichnungen, Achsen-Beschriftungen und Schatten befinden sich nicht zwingend innerhalb des Bereichs.

ChartBorderWidth ☐
 Rahmen des Chart-Objekts
 Typ int
 Siehe Liste, Chart,
ChartColBorder

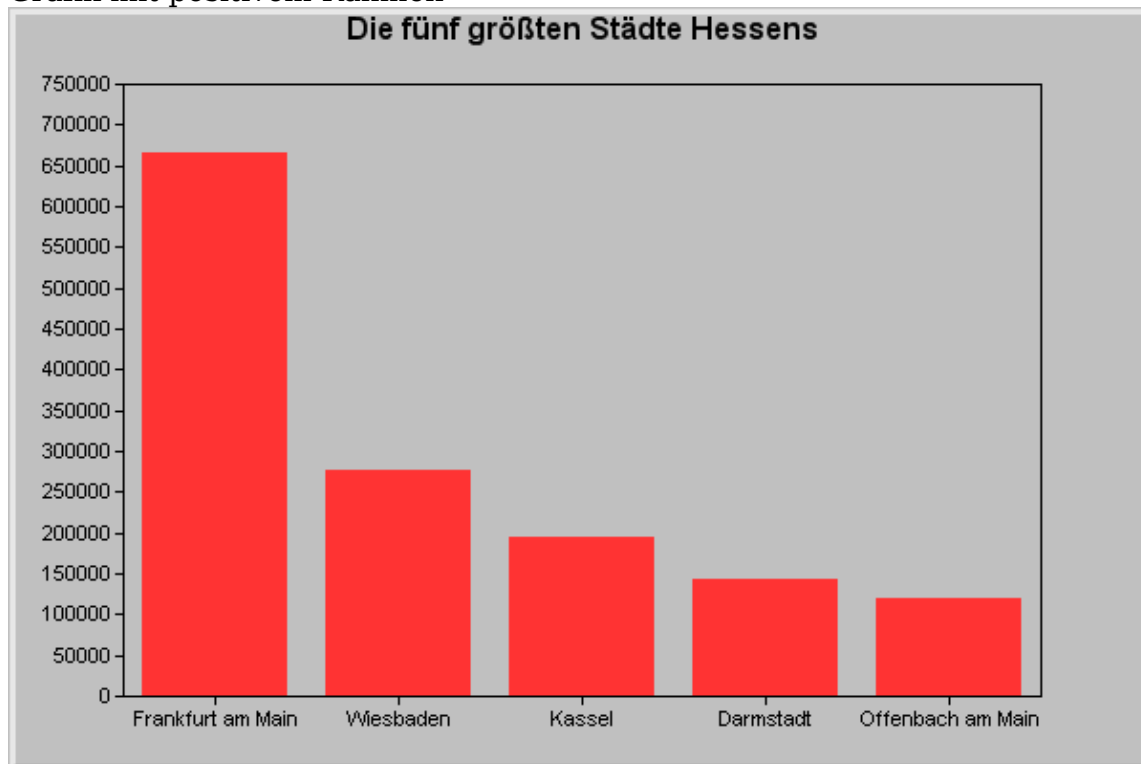
Die Eigenschaft definiert die Breite des Rahmens, der um das Chart-Objekt gezeichnet werden soll. Der Rahmen wird innerhalb des Ausgabebereichs gezeichnet. Wird ein positiver Wert angegeben, wird ein erhabener Rahmen, bei einem negativen Wert, ein eingedrückter Rahmen gezeichnet.

Soll kein Rahmen gezeichnet werden, muss 0 angegeben werden.

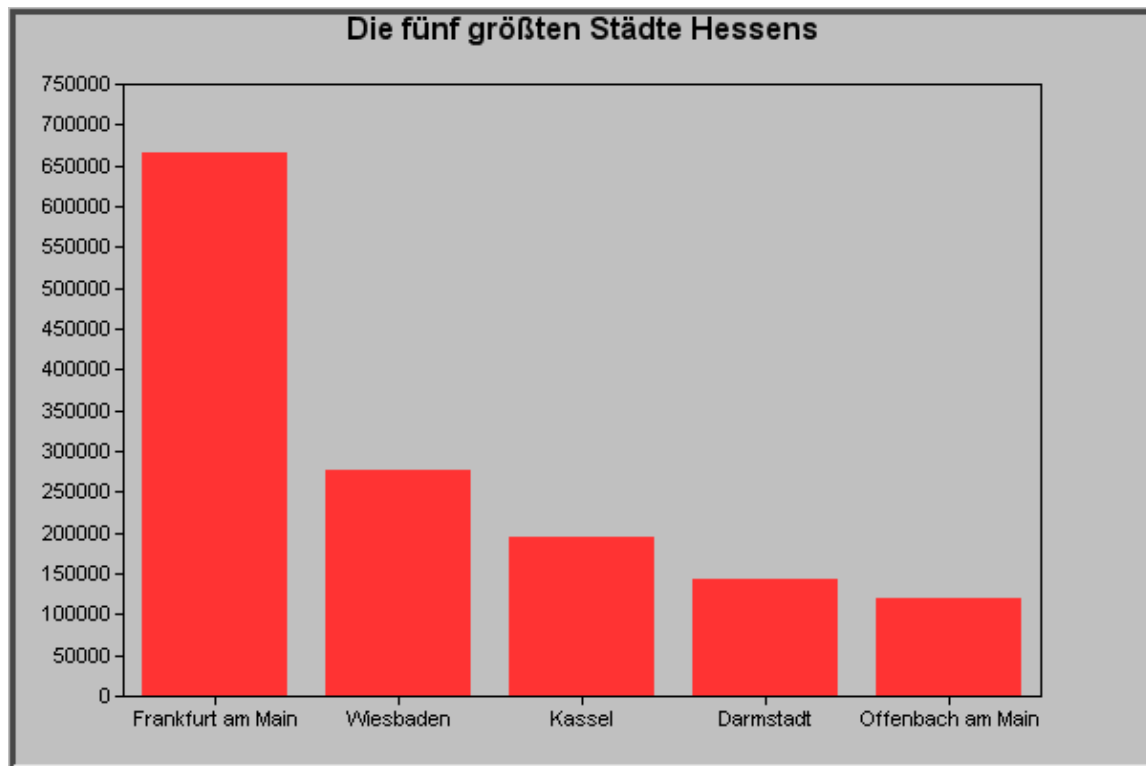
Beispiele:

Die folgenden Grafiken wurden mit einem drei Pixel starken Rand versehen. Zur besseren Unterscheidung zum Hintergrund wurde als Hintergrundfarbe der Grafik grau gewählt.

Grafik mit positivem Rahmen



Grafik mit negativem Rahmen



ChartColBkg



Hintergrundfarbe des Chart-Objekts

Typ color

Siehe Liste, Chart

Mit dieser Eigenschaft kann die Hintergrundfarbe des Chart-Objekts gesetzt und abgefragt werden.

ChartColBorder ☐

Randfarbe des Chart-Objekts

Typ color

Siehe Liste, Chart,
ChartBorderWidth

Mit dieser Eigenschaft kann die Farbe des Randes des Chart-Objekts gesetzt und abgefragt werden. Es wird kein Rand gezeichnet, wenn die Farbe auf ColorMake(_WinTransparent, 0) oder die Eigenschaft ChartBorderWidth auf 0 gesetzt wird.

ChartHeight



Höhe des Ausgabebereiches des Chart-Objekts

Typ int

Siehe Liste, Chart, ChartWidth, ChartOpen()

Mit dieser Eigenschaft kann die Höhe des Ausgabebereiches des Chart-Objekts gesetzt oder ermittelt werden.

ChartLegendColBkg ☐

Hintergrundfarbe der Legende

Typ color

Siehe Liste, Chart

Mit dieser Eigenschaft kann die Hintergrundfarbe der Legende gesetzt oder abgefragt werden. Standardmäßig ist die Hintergrundfarbe transparent.

ChartLegendColBorder ☐

Rahmenfarbe der Legende

Typ color

Siehe Liste, Chart

Mit dieser Eigenschaft kann die Rahmenfarbe der Legende gesetzt und abgefragt werden. Standardmäßig ist die Farbe auf schwarz gesetzt.

ChartLegendKeyGap



Abstand zwischen Quadrat und Text in der Legende

Typ int

Siehe Liste, Chart

Mit dieser Eigenschaft kann der Abstand zwischen den farbigen Quadraten und dem Text in der Legende gesetzt oder ermittelt werden. Die Angabe erfolgt in Pixel. Standardmäßig ist der Abstand auf 8 Pixel gesetzt.

ChartLegendKeySize



Größe der farbigen Quadrate in der Legende

Typ point

Siehe Liste, Chart

Diese Eigenschaft bestimmt die Größe der farbigen Quadrate, die in der Legende gezeichnet werden. Standardmäßig werden Quadrate mit 10 x 10 Pixel gezeichnet. Es wird immer ein Quadrat gezeichnet, auch wenn eine nicht-quadratische Größe (zum Beispiel PointMake(20, 10)) angegeben wurde. Das Quadrat wird dann so groß wie möglich (in dem Beispiel 10 x 10 Pixel) in der Mitte des Bereichs gezeichnet.

Beispiel:

```
tHdlChart->spChartLegendKeySize # PointMake(20, 20);
```


ChartLegendPos ☐

Position der Legende

Typ point

Siehe Liste, Chart

Diese Eigenschaft setzt oder ermittelt die Position der Legende im Chart-Objekt, sofern das Objekt mit einer Legende erstellt wurde (siehe ChartOpen()).

Standardmäßig wird die Legende an die Position (0, 0) im Ausgabebereich gezeichnet.

ChartNumFmtThousandSep□

Tausendertrennzeichen

Typ alpha(1)

Siehe Liste, Chart

Diese Eigenschaft definiert das Tausendertrennzeichen für die Darstellung von Zahlen im Chart-Objekt. Das Zeichen '~' (Tilde) ist reserviert für kein Tausendertrennzeichen.

ChartPieAngle




Winkel des Schattens / Blickwinkel auf das Torten-Diagramm

Typ int

Siehe Liste, Chart

In dieser Eigenschaft kann der Winkel des Schattens (ChartPieShadowMode = true) bzw. der Blickwinkel auf das Torten-Diagramm (ChartPieShadowMode = false) gesetzt und ermittelt werden.

Bei der Darstellung mit Schatten können Winkel bis 360 Grad, bei der Darstellung als Torten-Diagramm bis 90 Grad angegeben werden.

ChartPieColEdge 

Rahmenfarbe der Sektoren

Typ color

Siehe Liste, Chart

Mit dieser Eigenschaft kann die Rahmenfarbe der Sektoren in einem Torten-Diagramm gesetzt und abgefragt werden.

Standardmäßig wird kein Rahmen gezeichnet.

Kontakt

ChartPieColJoin



Farbe der Linien zwischen Sektor und Bezeichner

Typ color

Siehe Liste, Chart

Mit dieser Eigenschaft wird die Farbe der Verbindung zwischen dem Sektor und dem Bezeichner in einem Torten-Diagramm gesetzt und ermittelt.

Standardmäßig wird kein Verbinder gezeichnet.

ChartPieDepth



Tiefe des Schatten / Höhe des Torten-Diagramms

Typ int

Siehe Liste, Chart, ChartPieShadowMode

In dieser Eigenschaft kann die Tiefe des Schattens (ChartPieShadowMode = true) bzw. die Höhe des Torten-Diagramms (ChartPieShadowMode = false) gesetzt und abgefragt werden.

Die Richtung des Schattens bzw. der Blickwinkel auf das Torten-Diagramm werden in der Eigenschaft ChartPieAngle definert.

ChartPieInnerRadius



Innenradius des Torten-Diagramms

Typ int

Siehe Liste, Chart

In dieser Eigenschaft kann der Innenradius des Torten-Diagramms gesetzt und ermittelt werden, sodass das Diagramm als Donut dargestellt wird. Standardmäßig beträgt der Innenradius 0.

ChartPieLabelFont ☐

Schrift der Bezeichner

Typ font

Siehe Liste, Chart

Mit dieser Eigenschaft kann die Schrift der Bezeichner der Sektoren in einem Torten-Diagramm gesetzt oder gelesen werden. Alle Bezeichner werden mit der gleichen Schrift angezeigt. Als Standard wird die Schriftart Arial verwendet.

ChartPieShading



Schattierung des Torten-Diagramms

Typ intSiehe Liste, Chart

Mit dieser Eigenschaft kann die Schattierung des Torten-Diagramms gesetzt oder ermittelt werden. Der Wert kann mit folgenden Konstanten gesetzt werden:

- **_ChartPieShadingDefault**

Keine Schattierung.

- **_ChartPieShadingFlat**

Keine Schattierung.

- **_ChartPieShadingGradientLocal**

Farbverlauf für jeden Sektor von oben nach unten heller werdend.

- **_ChartPieShadingGradientGlobal**

Farbverlauf von oben nach unten heller werdend.

- **_ChartPieShadingConcave**

Eingedellte Torten-Darstellung.

- **_ChartPieShadingRoundedEdge**

Abgerundete Sektoren-Kanten.

- **_ChartPieShadingRadial**

Radialer Farbverlauf von der Mitte des Torten-Diagramms ausgehend.

- **_ChartPieShadingRing**

Ringförmiger Farbverlauf von der Mitte des Torten-Diagramms ausgehend.

Die Konstanten können nicht miteinander kombiniert werden.

ChartPieShadowMode ☐

Torten-Diagramm mit Schatten

Typ logic

Siehe Liste, Chart

Mit dieser Eigenschaft wird bestimmt, ob das Torten-Diagramm mit einem Schatten (true) oder dreidimensional (false) gezeichnet wird. Die Eigenschaften ChartPieDepth und ChartPieAngle geben die Tiefe des Schattens bzw. die Höhe der Torte und den Winkel des Schattens bzw. den Betrachtungswinkel an.

ChartPieStartAngle



Startwinkel und Anzeigereihenfolge des Torten-Diagramms

Typ int

Siehe Liste, Chart

In dieser Eigenschaft können der Startwinkel und die Anzeigereihenfolge des Torten-Diagramms gesetzt und ermittelt werden. Es können Werte zwischen -360 und +360 Grad angegeben werden. Bei negativen Werten startet der erste Abschnitt von oben um die definierte Gradzahl entgegen dem Uhrzeigersinn. Weitere Abschnitte werden der Reihenfolge nach auch entgegen dem Uhrzeigersinn erzeugt. Bei positiven Werten werden die Abschnitte im Uhrzeigersinn angeordnet. Standardmäßig beträgt der Startwinkel 0.

ChartPyramidElevation



Vertikaler Blickwinkel auf das Pyramiden-Diagramm

Typ int

Siehe Liste, Chart

In dieser Eigenschaft kann der vertikale Blickwinkel (Rotation um die X-Achse) auf das Pyramiden-Diagramm gesetzt und ermittelt werden. Standardmäßig beträgt der vertikale Blickwinkel 0 Grad.

ChartPyramidLayerGap ☐

Abstand der Ebenen untereinander

Typ float

Siehe Liste, Chart

Mit dieser Eigenschaft kann der Abstand zwischen den Ebenen bei einem Pyramiden-Diagramm gesetzt werden. In dieser Eigenschaft können Werte zwischen 0.0 und 1.0 angegeben werden, die den prozentualen Anteil des Zwischenraums angeben. Standardmäßig beträgt der Abstand 0.0.



Die Gesamtsumme der Zwischenräume zwischen allen Ebenen kann 1.0 nicht überschreiten.

ChartPyramidRotation



Horizontaler Blickwinkel auf das Pyramiden-Diagramm

Typ int

Siehe Liste, Chart

In dieser Eigenschaft kann der horizontale Blickwinkel (Rotation um die Y-Achse) auf das Pyramiden-Diagramm gesetzt und ermittelt werden. Standardmäßig beträgt der horizontale Blickwinkel 0 Grad.

ChartPyramidStyleData ☐

Stil der Datendarstellung

Typ int

Siehe Liste, Chart

Mit dieser Eigenschaft kann der Stil der Datendarstellung in einem Pyramiden-Diagramm gesetzt und gelesen werden. Folgende Stile stehen zur Verfügung:

- **_ChartPyramidStyleDataDefault / _ChartPyramidStyleDataPyramid**

Darstellung als Pyramide.



- **_ChartPyramidStyleDataCone**

Darstellung als Kegel.



- **_ChartPyramidStyleDataFunnel**

Darstellung als Trichter.



ChartPyramidTubeHeight ☐

Höhe der Röhre in %

Typ float

Siehe Liste, Chart

Mit dieser Eigenschaft kann bei der Trichterdarstellung (ChartPyramidStyleData = _ChartPyramidStyleDataFunnel) in einem Pyramiden-Diagramm die Höhe der Röhre gesetzt und ermittelt werden. In dieser Eigenschaft können Werte zwischen 0.0 und 1.0 angegeben werden, die den prozentualen Anteil der Trichter-Röhre gegenüber der Gesamthöhe angeben. Standardmäßig ist der Wert 0.3.

ChartPyramidTubeRadius ☐

Radius der Röhre in %

Typ float

Siehe Liste, Chart

Mit dieser Eigenschaft kann bei der Trichterdarstellung (ChartPyramidStyleData = ChartPyramidStyleDataFunnel) in einem Pyramiden-Diagramm der Radius der Röhre gesetzt und ermittelt werden. In dieser Eigenschaft können Werte zwischen 0.0 und 1.0 angegeben werden, die den prozentualen Anteil der Trichter-Röhre gegenüber dem Radius der Einfüllöffnung angeben. Standardmäßig ist der Wert 0.2.

ChartPyramidTwist



Blickwinkel der Kamera auf das Pyramiden-Diagramm

Typ int

Siehe Liste, Chart

In dieser Eigenschaft kann der Blickwinkel der Kamera auf das Pyramiden-Diagramm gesetzt und ermittelt werden. Standardmäßig beträgt der Kamerablickwinkel 0 Grad.

ChartSurfaceAmbient



Umgebungsreflektion bei einem Oberflächen-Diagramm

Typ float

Siehe Liste, Chart

Mit dieser Eigenschaft kann die Stärke der Umgebungsreflektion bei einem Oberflächen-Diagramm gesetzt werden. In dieser Eigenschaft können Werte zwischen 0.0 und 10.0 angegeben werden. Standardmäßig beträgt die Umgebungsreflektion 0.5.

ChartSurfaceColContour



Farbe der Z-Achsenlinien auf der Oberfläche

Typ color

Siehe Liste, Chart

Mit dieser Eigenschaft kann die Farbe der Z-Achsenlinien eines Oberflächen-Diagramm gesetzt und gelesen werden. Ist die Eigenschaft gesetzt, werden die Achsenlinien in der gewählten Farbe auf die Oberfläche projiziert. Standardmäßig ist keine Farbe gesetzt.

ChartSurfaceColSurfaceAxis



Farbe der X- und Y-Achsenlinien auf der Oberfläche

Typ color

Siehe Liste, Chart

Mit dieser Eigenschaft kann die Farbe der X- und Y-Achsenlinien eines Oberflächen-Diagramm gesetzt und gelesen werden. Ist die Eigenschaft gesetzt, werden die Achsenlinien in der gewählten Farbe auf die Oberfläche projiziert. Standardmäßig ist keine Farbe gesetzt.

ChartSurfaceColSurfaceData



Farbe der Datenlinien auf der Oberfläche

Typ color

Siehe Liste, Chart

Mit dieser Eigenschaft kann die Farbe der Datenlinien eines Oberflächen-Diagramm gesetzt und gelesen werden. Ist die Eigenschaft gesetzt, werden Verbindungslinien zwischen den Datenpunkten in der gewählten Farbe auf die Oberfläche projiziert. Standardmäßig ist keine Farbe gesetzt.

ChartSurfaceContinuous ☐

Fließende Farbübergänge

Typ logic

Siehe Liste, Chart

Mit dieser Eigenschaft kann bei Oberflächen-Diagrammen gesetzt oder ermittelt werden, ob fließende Farbübergänge verwendet werden sollen, oder nicht. Standardmäßig werden keine fließenden Farbübergänge verwendet.

ChartSurfaceDiffuse



Streureflection bei einem Oberflächen-Diagramm

Typ float

Siehe Liste, Chart

Mit dieser Eigenschaft kann die Stärke der Streureflection bei einem Oberflächen-Diagramm gesetzt werden. In dieser Eigenschaft können Werte zwischen 0.0 und 10.0 angegeben werden. Standardmäßig beträgt die Streureflection 0.5.

ChartSurfaceElevation



Vertikaler Blickwinkel auf das Oberflächen-Diagramm

Typ int

Siehe Liste, Chart

In dieser Eigenschaft kann der vertikale Blickwinkel (Rotation um die X-Achse) auf das Oberflächen-Diagramm gesetzt und ermittelt werden. Standardmäßig beträgt der vertikale Blickwinkel 30 Grad.

ChartSurfaceLegendLength



Länge der Farb-Legende beim Oberflächen-Diagramm

Typ int

Siehe Liste, Chart

Mit dieser Eigenschaft kann die Länge der Farb-Legende bei Oberflächen-Diagrammen gesetzt oder ermittelt werden. Die Angabe erfolgt in Pixel. Standardmäßig ist die Länge auf 200 Pixel gesetzt.

ChartSurfacePerspective ☐

Stärke des Perspektiveneffekts

Typ int

Siehe Liste, Chart

In dieser Eigenschaft kann die Stärke des Perspektiveneffekts des Oberflächen-Diagramm gesetzt und ermittelt werden. Es können Werte zwischen 0 und 100 angegeben werden. Standardmäßig beträgt der Perspektiveneffekt 12.

Je stärker der Perspektiveneffekt ist, desto stärker ist der Größenunterschied zwischen zwei Objekten, die gleich hoch, aber unterschiedlich weit entfernt sind.

ChartSurfacePlotHeight



Höhe der Z-Achse beim Oberflächen-Diagramm

Typ int

Siehe Liste, Chart

Mit dieser Eigenschaft kann die Höhe der Z-Achse bei Oberflächen-Diagrammen gesetzt oder ermittelt werden. Die Angabe erfolgt in Pixel. Standardmäßig ist die Höhe auf 200 Pixel gesetzt.

ChartSurfaceRotation



Horizontaler Blickwinkel auf das Oberflächen-Diagramm

Typ int

Siehe Liste, Chart

In dieser Eigenschaft kann der horizontale Blickwinkel (Rotation um die Y-Achse) auf das Oberflächen-Diagramm gesetzt und ermittelt werden. Standardmäßig beträgt der horizontale Blickwinkel 45 Grad.

ChartSurfaceSamplesX



Datenanzahl der X-Achse für Interpolation

Typ int

Siehe Liste, Chart, ChartSurfaceSamplesY,
ChartSurfaceSmooth

Mit dieser Eigenschaft kann die Datenanzahl der X-Achse für Interpolation bei Oberflächen-Diagrammen gesetzt oder ermittelt werden. Standardmäßig werden keine Daten interpoliert.



Je mehr Punkte interpoliert werden sollen, umso länger dauert die Erstellung des Diagramms, jedoch wird die Darstellung bei mehr Punkten verfeinert. Ist die Eigenschaft ChartSurfaceSmooth auf true gesetzt, genügen meist relativ wenige Punkte.

ChartSurfaceSamplesY



Datenanzahl der Y-Achse für Interpolation

Typ int

Siehe Liste, Chart, ChartSurfaceSamplesX,
ChartSurfaceSmooth

Mit dieser Eigenschaft kann die Datenanzahl der Y-Achse für Interpolation bei Oberflächen-Diagrammen gesetzt oder ermittelt werden. Standardmäßig steht diese Eigenschaft auf dem Wert -1, bei dem die Datenanzahl der X-Achse verwendet wird.



Je mehr Punkte interpoliert werden sollen, umso länger dauert die Erstellung des Diagramms, jedoch wird die Darstellung bei mehr Punkten verfeinert. Ist die Eigenschaft ChartSurfaceSmooth auf true gesetzt, genügen meist relativ wenige Punkte.

ChartSurfaceShadingMode



Untergrund des Oberflächen-Diagramms

Typ int

Siehe Liste, Chart

Mit dieser Eigenschaft kann der Untergrund in einem Oberflächen-Diagramm gesetzt und gelesen werden. Folgende Untergründe stehen zur Verfügung:

- **_ChartSurfaceShadingDefault / _ChartSurfaceShadingSmooth**

Untergrund mit Kantenglättung.

- **_ChartSurfaceShadingTriangle**

Untergrund ist aus Dreiecken zusammengesetzt.

- **_ChartSurfaceShadingRectangle**

Untergrund ist aus Rechtecken zusammengesetzt.

- **_ChartSurfaceShadingTriangleFrame**

Untergrund ist aus Dreiecken zusammengesetzt. Es werden nur die Ränder angezeigt. Die Dicke der Ränder kann in der Eigenschaft ChartSurfaceWireWidth definiert werden.

- **_ChartSurfaceShadingRectangleFrame**

Untergrund ist aus Rechtecken zusammengesetzt. Es werden nur die Ränder angezeigt. Die Dicke der Ränder kann in der Eigenschaft ChartSurfaceWireWidth definiert werden.

ChartSurfaceShine



Glanz bei einem Oberflächen-Diagramm

Typ float

Siehe Liste, Chart

Mit dieser Eigenschaft kann die Stärke des Glanzes bei einem Oberflächen-Diagramm gesetzt werden. In dieser Eigenschaft können Werte zwischen 0.0 und 100.0 angegeben werden. Standardmäßig beträgt die Stärke des Glanzes 8.0.

ChartSurfaceSmooth ☐
Interpolation mit Kantenglättung

Typ logic

Liste, Chart,

Siehe ChartSurfaceSamplesX,
ChartSurfaceSamplesY

Mit dieser Eigenschaft kann bei Oberflächen-Diagrammen gesetzt oder ermittelt werden, ob die Kanten bei einer Interpolation geglättet (true) werden sollen, oder nicht. Standardmäßig werden die Kanten geglättet.

Ist diese Eigenschaft auf true gesetzt, können oftmals mit wenigen Punkten zur Interpolation ähnliche Ergebnisse erreicht werden, wie mit vielen Punkten. Dies ist möglich, da die Punkte mit einer Spline-Linie verbunden werden, statt mit einer geraden.

ChartSurfaceSpecular



Spiegelreflektion bei einem Oberflächen-Diagramm

Typ float

Siehe Liste, Chart

Mit dieser Eigenschaft kann die Stärke der Spiegelreflektion bei einem Oberflächen-Diagramm gesetzt werden. In dieser Eigenschaft können Werte zwischen 0.0 und 20.0 angegeben werden. Standardmäßig beträgt die Spiegelreflektion 1.0.

ChartSurfaceTitleFontX ☐

Schrift der x-Achsen-Beschriftung

Typ font

Siehe Liste, Chart

Mit dieser Eigenschaft kann die Schriftart, mit der der Titel der x-Achse gezeichnet wird, gesetzt und ermittelt werden. Als Standard wird die Schriftart Arial verwendet.