_ErrUnchangeable Objekt kann nicht geändert werden Wert -62

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: Verarbeitungsfehler

Ursache : Es wurde versucht ein Objekt zu ändern, das nicht geändert werden darf.

Dieser Fehlerwert wird zum Beispiel zurückgegeben, wenn versucht wird den

Benutzer "SUPERUSER" oder die Benutzergruppe "everyone" zu löschen.

_ErrUnerasable Objekt kann nicht gelöscht werden Wert -61

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: Verarbeitungsfehler

Ursache : Es wurde versucht ein Objekt zu löschen, das nicht gelöscht werden darf.

Dieser Fehlerwert wird zum Beispiel zurückgegeben, wenn versucht wird den Benutzer "SUPERUSER" oder die Benutzergruppe "_everyone" zu löschen.

_ErrUnknown Objekt ist unbekannt Wert -65

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: : Verarbeitungsfehler

Diese Konstante wird von den Anweisungen JobOpen() und

Ursache: : <u>SvcSessionControl()</u> zurückgegeben, wenn das übergebene <u>Job</u>-Objekt

oder die übergebene Session-Id nicht vorhanden ist.

Konstanten für Netzwerkinformationsfehler

 $Siehe \frac{\underline{Alle}}{\underline{Befehle}}$

<u>ErrNetCreate</u> ICMP-Sockets nicht vorhanden. <u>ErrNetIcmpID</u> Antwortpaket mit der falschen ID.

<u>ErrNetIcmpType</u> Antwortpaket vom falschen Typ.

<u>ErrNetNoHost</u> Hostadresse nicht ermittelbar. <u>ErrNetRead</u> Fehler beim Lesen der Antwort.

<u>ErrNetReadLess</u> Antwortpaket ungültig.

<u>ErrNetSelect</u> Fehler bei der Socketabfrage.

<u>ErrNetWrite</u> Anfrage nicht versendet.

_ErrNetCreate ICMP-Sockets nicht vorhanden Wert -706

Siehe $\frac{\text{NetInfo()}}{\text{ErrGet()}}$

Kategorie: Netzwerk informations fehler

ICMP-Sockets nicht verfügbar. Der systemeigene Befehl "ping" arbeitet

Ursache : mit ICMP-Paketen und falls kein ICMP-Socket vorhanden ist, können die

Pakete nicht verarbeitet werden.

ICMP steht für Internet Control Message Protocol.

_ErrNetIcmpID Antwortpaket mit der falschen ID Wert -720

Siehe $\frac{\text{NetInfo()}}{\text{ErrGet()}}$

Kategorie: Netzwerk informations fehler

Ursache : Antwortpaket mit der falschen ID.

_ErrNetIcmpType Antwortpaket vom falschen Typ Wert -719

Siehe $\frac{\text{NetInfo()}}{\text{ErrGet()}}$

Kategorie : Netzwerkinformationsfehler Ursache : Antwortpaket vom falschen Typ.

ErrNetNoHost Hostadresse nicht ermittelbar Wert -705

Siehe $\frac{\text{NetInfo()}}{\text{ErrGet()}}$

Kategorie: Netzwerkinformationsfehler

 $\label{eq:Ursache} \mbox{Ursache} \quad : \frac{\mbox{Die Hostadresse eines im Netz gesuchten Rechners konnte nicht ermittelt}}{\mbox{werden.}}$

_ErrNetRead Fehler beim Lesen der Antwort Wert -710

Siehe $\frac{\text{NetInfo()}}{\text{ErrGet()}}$

Kategorie : Netzwerkinformationsfehler Ursache : Fehler beim Lesen der Antwort.

_ErrNetReadLess Antwortpaket ungültig Wert -712

Siehe $\frac{\text{NetInfo()}}{\text{ErrGet()}}$

Kategorie: Netzwerk informations fehler

Ursache: Antwortpaket ungültig.

_ErrNetSelect Fehler bei der Socketabfrage Wert -709

Siehe $\frac{\text{NetInfo()}}{\text{ErrGet()}}$

Kategorie : Netzwerkinformationsfehler Ursache : Fehler bei der Socketabfrage.

_ErrNetWrite Anfrage nicht versendet Wert -713

Siehe $\frac{\text{NetInfo()}}{\text{ErrGet()}}$

Kategorie : Netzwerkinformationsfehler Ursache : Anfrage nicht versendet.

Konstanten für Socketfehler

 $Siehe \frac{\underline{Alle}}{\underline{Befehle}}$

<u>ErrSckBindFailed</u> Es konnte keine Verbindung zum Zielport aufgebaut

werden.

<u>ErrSckConnectFailed</u> Es konnte keine Verbindung zum Zielhost aufgebaut

werden.

<u>ErrSckCreate</u> Der Socket konnte nicht angelegt werden.

<u>ErrSckDown</u> Der Socket wurde außerhalb der Applikation

geschlossen.

ErrSckHostUnknown Der Name des Zielhosts konnte nicht in eine IP-Adresse

übersetzt werden.

<u>ErrSckNoLib</u> Das TCP/IP-Protokoll konnte nicht initialisiert werden.

<u>ErrSckProxyAuthFailed</u> Authentifikation beim Proxy fehlgeschlagen.

<u>ErrSckProxyConnectFailed</u> Es konnte keine Verbindung zum Proxy aufgebaut

werden.

<u>ErrSckProxyFailed</u> Kommunikation mit dem Proxy ist fehlgeschlagen.

<u>ErrSckProxyRead</u> Bei der Kommunikation mit dem Proxy ist ein Lesefehler

aufgetreten.

<u>ErrSckProxyRefused</u> Der Proxy hat den Verbindungswunsch abgelehnt.

<u>ErrSckProxyUnknown</u> Der Name des Proxys konnte nicht in eine IP-Adresse

übersetzt werden.

<u>ErrSckProxyWrite</u> Bei der Kommunikation mit dem Proxy ist ein

Schreibfehler aufgetreten.

<u>ErrSckRead</u> Beim Lesen des Sockets ist ein Fehler aufgetreten

(beispielsweise durch Abbruch der Verbindung).

<u>ErrSckReadOverflow</u> Beim Lesen des Sockets mit der Option <u>SckLine</u> ist die

empfangene Zeile länger als die angegebene Variable.

<u>ErrSckSelect</u> Bei der Socketabfrage ist ein Fehler aufgetreten

(beispielsweise durch Abbruch der Verbindung).

<u>ErrSckTlsCertificateVerify</u> Fehler bei der Überprüfung des serverseitigen

Zertifikats.

<u>ErrSckTlsConnect</u> Fehler beim Aufbau einer verschlüsselten Verbindung.

ErrSckWrite Beim Lesen des Sockets ist ein Fehler aufgetreten

(beispielsweise durch Abbruch der Verbindung).

ErrSckBindFailed

Es konnte keine Zuordnung zum angegebenen Port erfolgen

Wert -714

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Socket-Funktionen

Ursache : Möglicherweise ist der angegeben Port bereits in Benutzung.

<u>Verwandte</u>

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Socket-Funktionen

Ursache : Es konnte keine Verbindung zum Zielhost aufgebaut werden.

Die Fehlernummer wird ebenfalls in den <u>Log-Einträgen</u> des Servers ausgegeben, wenn im Hot-Standby Betrieb einer der beteiligten Server seinen Partner-Server nicht

findet.

_ErrSckCreate Der Socket konnte nicht angelegt werden Wert -706

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Socket-Funktionen

Ursache : Der Socket konnte nicht angelegt werden.

ErrSckDown

Der Socket wurde von der anderen Seite geschlossen

Wert -717

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Socket-Funktionen

Ursache : Der Socket wurde außerhalb der Applikation geschlossen.

ErrSckHostUnknown

Der Name des Zielhosts konnte nicht in eine IP-Adresse übersetzt werden Wert -705

Verwandte

Siehe Befehle,

ErrGet()

Kategorie : Socket-Funktionen

Ursache : Der Name des Zielhosts konnte nicht in eine IP-Adresse übersetzt werden. Die Fehlernummer wird ebenfalls in den <u>Log-Einträgen</u> des Servers ausgegeben, wenn im Hot-Standby Betrieb einer der beteiligten Server den Namen seines Partner-Servers nicht in eine IP-Adresse auflösen kann.

ErrSckNoLib

Das TCP/IP-Protokoll konnte nicht initialisiert werden

Wert -701

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Socket-Funktionen

Ursache : Das TCP/IP-Protokoll konnte nicht initialisiert werden.

_ErrSckProxyAuthFailed Authentifikation beim Proxy fehlgeschlagen Wert -733

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Socket-Funktionen

Ursache : Der angegebene Benutzer oder das angegebene Passwort

ist ungültig.

_ErrSckProxyConnectFailed Es konnte keine Verbindung zum Proxy aufgebaut werden Wert -725

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Socket-Funktionen

Ursache : Es konnte keine Verbindung zum Proxy aufgebaut werden.

ErrSckProxyFailed Kommunikation mit dem Proxy ist fehlgeschlagen Wert -732

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: Socket-Funktionen

Ursache : Vom Proxy wurde eine ungültige Antwort geschickt oder die angeforderte Authentifizierungsmethode wird von Proxy nicht unterstützt.

ErrSckProxyRead

Bei der Kommunikation mit dem Proxy ist ein Lesefehler aufgetreten Wert -726

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Socket-Funktionen

Ursache : Bei der Kommunikation mit dem Proxy ist ein Lesefehler

aufgetreten.

ErrSckProxyRefused

Der Proxy hat den Verbindungswunsch abgelehnt.

Wert -724

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Socket-Funktionen

Der Proxy hat den Verbindungswunsch abgelehnt. Dies kann auf Grund

Ursache : von technischen Problemen (keine Netzwerkverbindung) oder auf Grund

von Vorgaben beim Proxy erfolgen.

ErrSckProxyUnknown

Der Name des Proxys konnte nicht in eine IP-Adresse übersetzt werden Wert -722

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Socket-Funktionen

Ursache : Der Name des Proxys konnte nicht in eine IP-Adresse übersetzt werden.

ErrSckProxyWrite

Bei der Kommunikation mit dem Proxy ist ein Schreibfehler aufgetreten Wert -727

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Socket-Funktionen

Ursache : Bei der Kommunikation mit dem Proxy ist ein Schreibfehler

aufgetreten.

ErrSckRead

Beim Lesen des Sockets ist ein Fehler aufgetreten

Wert -710

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: Socket-Funktionen

: Beim Lesen des Sockets ist ein Fehler aufgetreten (beispielsweise durch Abbruch der Verbindung).

Ursache

Die Fehlernummer wird ebenfalls in den Log-Dateien des Servers ausgegeben, wenn beim Lesevorgang auf die Hot-Standby Direktverbindung ein Problem auftritt.

ErrSckReadOverflow empfangene Zeile länger als die angegebene Variable Wert -711

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Socket-Funktionen

Beim Lesen des Sockets mit der Option <u>SckLine</u> ist die empfangene Zeile länger als die angegebene Variable. Ursache

ErrSckSelect

Bei der Socketabfrage ist ein Fehler aufgetreten

Wert -709

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: Socket-Funktionen

: Bei der Socketabfrage ist ein Fehler aufgetreten (beispielsweise durch Abbruch der Verbindung).

Ursache

Die Fehlernummer wird ebenfalls in den Log-Dateien des Servers ausgegeben, wenn beim Lese- oder Schreibvorgang auf die Hot-Standby Direktverbindung ein Problem auftritt.

_ErrSckTlsCertificateVerify Fehler bei der Überprüfung des serverseitigen Zertifikats Wert -745

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: Socket-Funktionen

Es ist ein Fehler bei der Überprüfung des serverseitigen Zertifikats

Ursache : aufgetreten. Der Fehler kann über die Eigenschaft CertificateError des

<u>Sys</u>-Objekts ermittelt werden.

ErrSckTlsConnect

Fehler beim Aufbau einer verschlüsselten Verbindung

Wert -741

Verwandte

Siehe Befehle,

ErrGet()

Kategorie: Socket-Funktionen

Der angegebene Server unterstützt keines der angegebenen

verschlüsselten Verbindungsprotokolle oder der TLS-Handshake ist

Ursache : fehlgeschlagen. Eine Mögliche Ursache könnte sein, dass der Server mehrere Namen hat und der korrekte Server per "Server Name

Indication" (<u>SckTlsSNI</u>) beim Verbindungsaufbau ausgewählt werden

muss.

ErrSckWrite

Beim Schreiben des Sockets ist ein Fehler aufgetreten

Wert -713

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: Socket-Funktionen

Beim Schreiben des Sockets ist ein Fehler aufgetreten (beispielsweise durch Abbruch der Verbindung).

Ursache

Die Fehlernummer wird ebenfalls in den Log-Dateien des Servers ausgegeben, wenn beim Schreiben auf die Hot-Standby Direktverbindung ein Problem auftritt.

Konstanten für Fehler bei der Mehrfachselektion

Siehe $\frac{\text{Alle}}{\text{Befehle}}$

<u>ErrMsdExists</u> Objekt ist bereits selektiert.

<u>ErrMsdNotFound</u> Objekt ist nicht selektiert.

_ErrMsdExists Objekt ist bereits selektiert Wert -1750 Siehe <u>WinMsdInsert()</u>

Dieser Wert wird von der Anweisung <u>WinMsdInsert()</u> zurückgegeben, wenn das übergebene Element bereits in der selektierten Menge vorhanden ist.

_ErrMsdNotFound Objekt nicht selektiert Wert -1751 Siehe <u>WinMsdDelete()</u>

Dieser Wert wird von der Anweisung <u>WinMsdDelete()</u> zurückgegeben, wenn das übergebene Element nicht in der selektierten Menge vorhanden ist.

Konstanten für Fehler bei dynamischen Selektionen

Siehe $\frac{\text{Alle}}{\text{Befehle}}$

<u>ErrParserEndOfText</u> Abfrage unvollständig.
<u>ErrParserIllegalElement</u> Ungültiges Element.
<u>ErrParserInvalidChar</u> Unerlaubtes Zeichen.
ErrParserInvalidConst Ungültige Konstante.

<u>ErrParserMissingComma</u> Komma fehlt.

<u>ErrParserMissingParenthesis</u> Klammer fehlt.

<u>ErrParserOutOfRange</u> Unzulässiger Wert.

<u>ErrParserStringOverflow</u> Zeichenkette zu lang.

<u>ErrParserSyntax</u> Syntaxfehler in der Abfrage.

<u>ErrParserUnknownID</u> Unbekannter Anker. <u>ErrParserWrongType</u> Falscher Typ angegeben.

<u>ErrSelCodeOverflow</u> Interner Fehler bei der Übersetzung und Speicherung

einer Selektion.

<u>ErrSelIllegalOperator</u> Unzulässiger Operator.

<u>ErrSelInvalidField</u> Ungültiges Feld.

<u>ErrSelInvalidKey</u> Vorauswahl mit falschem Schlüssel.

<u>ErrSelNoOuery</u> Verwendung einer undefinierten Abfrage.

<u>ErrSelQueryOverflow</u> Abfrage zu lang.

<u>ErrSelResultSet</u> Fehler bei einer Ergebnismenge.

<u>ErrSelTableOverflow</u> Interner Fehler bei der Übersetzung und Speicherung

einer Selektion.

<u>ErrSelUnknownField</u> Unbekanntes Feld.

<u>ErrSelUnknownOrInvalidLink</u> Unbekannte oder ungültige Verknüpfung.

ErrParserEndOfText Abfrage unvollständig Wert -2100

Verwandte

Siehe $\frac{\text{Befehle}}{\text{SelDefQuery()}}$,

ErrGet()

Kategorie : Selektionsdefinition

Ursache : Die Abfrage ist nicht vollständig definiert.

ErrParserIllegalElement Ungültiges Element Wert -2108

Verwandte

Siehe $\frac{\text{Befehle}}{\text{SelDefQuery()}}$,

ErrGet()

Kategorie : Selektionsdefinition

Ursache : Ein in der Abfrage enthaltenes Element ist nicht gültig.

ErrParserInvalidChar Unerlaubtes Zeichen Wert -2101

Verwandte

Siehe $\frac{\overline{Befehle},}{SelDefQuery()},$

ErrGet()

Kategorie : Selektionsdefinition

Ursache : In der Abfrage ist ein unerlaubtes Zeichen enthalten.

ErrParserInvalidConstUngültige Konstante Wert -2102

Verwandte

Siehe $\frac{\text{Befehle}}{\text{SelDefQuery()}}$,

ErrGet()

Kategorie : Selektionsdefinition

Ursache : In der Abfrage ist eine ungültige Konstante enthalten.

ErrParserMissingComma

Komma fehlt

Wert -2110

Verwandte

Siehe $\frac{\overline{Befehle}}{SelDefQuery()}$,

ErrGet()

Kategorie : Selektionsdefinition

Ursache : In der Abfrage fehlt ein Komma.

ErrParserMissingParenthesis

Klammer fehlt

Wert -2109

Verwandte

Siehe $\frac{\overline{Befehle},}{SelDefOuery()},$

ErrGet()

Kategorie : Selektionsdefinition

Ursache : In der Abfrage fehlt eine oder mehrere Klammern.

ErrParserOutOfRange Unzulässiger Wert Wert -2104

Verwandte

Siehe $\frac{\text{Befehle}}{\text{SelDefQuery()}}$,

ErrGet()

Kategorie: Selektionsdefinition

: In der Abfrage ist ein Wert außerhalb seines Definitionsbereichs

angegeben.

ErrParserStringOverflow Zeichenkette zu lang Wert -2105

Verwandte

Siehe $\frac{\text{Befehle}}{\text{SelDefQuery()}}$,

ErrGet()

Kategorie : Selektionsdefinition

Ursache : Eine Zeichenkette innerhalb der Abfrage ist zu lang.

ErrParserSyntax $\bar{S}yntaxfehle\mathring{r}$ Wert -2107

Verwandte

Siehe $\frac{\text{Befehle}}{\text{SelDefQuery()}}$,

ErrGet()

Kategorie : Selektionsdefinition

Ursache : In der Abfrage befindet sich ein Syntaxfehler.

ErrParserUnknownIDUnbekannter Anker Wert -2106

Verwandte

Siehe $\frac{\overline{Befehle}}{SelDefQuery()}$,

ErrGet()

Kategorie: Selektionsdefinition

Ursache : In der Abfrage wurde ein Anker angegeben, der zuvor nicht mit

<u>SelAddLink()</u> angegeben wurde.

ErrParserWrongType Falscher Typ angegeben Wert -2103

Verwandte

Siehe $\frac{\text{Befehle}}{\text{SelDefQuery()}}$,

ErrGet()

Kategorie : Selektionsdefinition

Ursache : In der Abfrage wurde ein falscher Typ verwendet.

ErrSelCodeOverflow Interner Fehler bei der Übersetzung und Speicherung einer Selektion Wert -2208

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Selektionsdefinition

Ursache : Beim Übersetzen und Speichern einer dynamischen Selektion mit der Anweisung <u>SelStore()</u> ist ein Code länger als 128 KB entstanden.

ErrSelIllegalOperator Ungültiger Operator Wert -2204

Verwandte

Siehe $\frac{\text{Befehle}}{\text{SelDefQuery()}}$,

ErrGet()

Kategorie : Selektionsdefinition

: In der Abfage wurde ein ungültiger Operator (zum Beispiel ><)

angegeben.

ErrSelInvalidField Feld ungültig Wert -2202

Verwandte

Siehe $\frac{\text{Befehle}}{\text{SelDefQuery()}}$,

ErrGet()

Kategorie : Selektionsdefinition

Ursache : In der Abfrage ist ein ungüliges Feld angegeben.

ErrSelInvalidKey Vorauswahl mit falschem Schlüssel Wert -2210

<u>Verwandte</u>

Siehe <u>Befehle</u>, <u>SelDefQuery()</u>,

ErrGet()

Kategorie: Selektionsdefinition

Bei der Anweisung SelDefOuerv() ist für eine Vorauswahl ein Schlüssel

angegeben, der nicht verwendet werden kann. Mindestens ein

Ursache : Schlüsselfeld muss im Abfragekriterium vorhanden sein. Der Fehler tritt

auch dann auf, wenn der angegebene Schlüssel nicht existiert oder eines

der Schlüsselfelder SOUNDEX 1 oder SOUNDEX 2 aktiviert hat.

ErrSelNoQuery Verwendung einer undefinierten Abfrage Wert -2209

Verwandte

Siehe Befehle,

ErrGet()

Kategorie: Selektions definition

 $\label{eq:Ursache} \mbox{Ursache } : \mbox{Innerhalb der dynamischen Selektion wird mit } \mbox{\underline{SelAddLink()}} \mbox{ auf eine nicht } \\ \mbox{vorhandene Abfrage verwiesen.}$

ErrSelQueryOverflow Abfrage zu lang Wert -2205

Verwandte

Siehe $\frac{\text{Befehle}}{\text{SelDefQuery()}}$,

ErrGet()

Kategorie: Selektionsdefinition

: Die Zeichenkette der Abfrage ist zu lang oder es wurden mehr als 100 Ursache

Abfrageelemente angegeben.

_ErrSelResultSet Fehler bei einer Ergebnismenge Wert -2206

Verwandte

Siehe Befehle,

ErrGet()

Kategorie : Selektionsdefinition

Mit der Anweisung <u>SelStore()</u> soll eine Selektion gespeichert werden, die keine Ergebnismenge besitzt. Die Ergebnismenge muss bei <u>SelCreate()</u>

Ursache : oder <u>SelAddResult()</u> angegeben werden. Der Fehler tritt auch auf, wenn

zu einer definierten Ergebnismenge keine Abfrage vorhanden ist oder

notwendige Ergebnismengen nicht definiert sind.

ErrSelTableOverflow Interner Fehler bei der Übersetzung und Speicherung einer Selektion Wert -2207

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Selektionsdefinition

Ursache : Beim Übersetzen und Speichern einer dynamischen Selektion mit der Anweisung <u>SelStore()</u> ist es zum Überlauf interner Tabellen gekommen.

ErrSelUnknownField Feldname unbekannt

Wert -2201

Verwandte

Siehe $\frac{\overline{Befehle}}{SelDefQuery()}$,

ErrGet()

Kategorie : Selektionsdefinition

Ursache : Es wurde ein unbekanntes Feld angegeben.

_ErrSelUnknownOrInvalidLink Verknüpfung unbekannt oder ungültig

Wert -2203

Verwandte

Befehle,

Siehe SelDefQuery(),

SelAddResult(),

ErrGet()

 $Kategorie \ : \ Selektions definition$

Ursache : Der angegebene Anker entspricht keiner Verknüpfung bzw. keiner

gültigen Verknüpfung.

Konstanten für Fehler bei Storage-Objekten

 $Siehe \frac{\underline{Alle}}{\underline{Befehle}}$

<u>ErrStoInvalidFormat</u> Ungültiges Format.

<u>ErrStoLocked</u> Das Storage-Objekt ist gesperrt.

<u>ErrStoNameInvalid</u> Der Name des Storage-Objekts oder -Verzeichnisses ist

ungültig.

<u>ErrStoNoData</u> Keine Daten vorhanden.

<u>ErrStoNoFile</u> Das Storage-Objekt existiert nicht.

<u>ErrStoNoPath</u> Das Storage-Verzeichnis existiert nicht. <u>ErrStoOperation</u> Operation kann nicht ausgeführt werden.

ErrStoInvalidFormat Ungültiges Format Wert -1511

Verwandte

Befehle,

Siehe ErrGet(),

<u>unterstützte</u> Dateiformate

Kategorie : Storage-Befehle

Ursache : Es wurde versucht ein ungültiges Format zu importieren.

_ErrStoLocked Das Storage-Objekt ist gesperrt Wert -1506

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Storage-Befehle

Ursache : Das Storage-Objekt ist von einem anderen Benutzer gesperrt. Die

Operation kann aber nur dann durchgeführt werden, wenn das Objekt

nicht oder vom eigenen Benutzer gesperrt ist.

ErrStoNameInvalid Name des Storage-Objekts/-Verzeichnisses ist ungültig Wert -1501

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: Storage-Befehle

Ursache

: Es ist ein ungültiger Name angegeben worden. Der Name darf nur aus den Buchstaben A...z den Ziffern 0...9 sowie dem "_" und dem "." bestehen, wobei das erste Zeichen ein Buchstabe sein muss. Er darf

maximal 40 Zeichen lang sein.

ErrStoNoData

Keine Daten vorhanden

Wert -1504

Verwandte

Befehle,

Siehe StoImport(),

StoExport(),

ErrGet()

Kategorie : Storage-Befehle

Ursache : Die angegebene externe Datei oder das angegebene Objekt beinhaltet

keine Daten.

_ErrStoNoFile Storage-Objekt existiert nicht Wert -1503

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Storage-Befehle

Ursache : Das angegebene Storage-Objekt existiert nicht.

ErrStoNoPath

Storage-Verzeichnis existiert nicht.

Wert -1502

Verwandte

Siehe Befehle,

ErrGet()

Kategorie : Storage-Befehle

Ursache : Das angegebene Storage-Verzeichnis existiert nicht.

_ErrStoOperation Operation kann nicht ausgeführt werden Wert -1510

Verwandte

Siehe Befehle,

ErrGet()

Kategorie: Storage-Befehle

Ursache : Es wurde versucht ein Storage-Objekt im Wurzelverzeichnis anzulegen.

Es wurde versucht ein Tile in ein falsches Verzeichnis zu importieren.

Es wurde versucht ein Oberflächen-Objekt zu importieren.

Es wurde versucht die Optionen <u>StoCreate</u> und <u>StoDirectory</u> zu

kombinieren.

Konstanten für Fehler der Tapi-Schnittstelle

Siehe $\frac{\text{Alle}}{\text{Befehle}}$

<u>ErrTapiBadAddr</u> Kein Anschluss

<u>ErrTapiBusy</u> Besetzt

<u>ErrTapiCallID</u> Ungültige Call-ID

<u>ErrTapiCallState</u> Operation in diesem Status nicht zulässig

<u>ErrTapiDevName</u> Ungültiges TAPI-Device
<u>ErrTapiDialString</u> Keine wählbare Nummer
<u>ErrTapiDialTimeout</u> TAPI-Timeout überschritten
<u>ErrTapiFailed</u> Operation nicht erfolgreich

<u>ErrTapiInstall</u> TAPI-Schnittstelle nicht korrekt installiert

<u>ErrTapiInUse</u> TAPI-Gerätetreiber in Benutzung <u>ErrTapiMediaMode</u> Nicht unterstützter Medienmodus

<u>ErrTapiMemory</u> Kein Speicher <u>ErrTapiNoConnect</u> Nicht verbunden

<u>ErrTapiNoListen</u> TapiDevice wird nicht überwacht

<u>ErrTapiNotOwner</u> Keine Besitzrechte

<u>ErrTapiReinit</u> TAPI-Konfiguration wurde verändert

<u>ErrTapiReject</u> Anruf abgelehnt

<u>ErrTapiUnavail</u> Operation oder Verbindung nicht verfügbar

<u>ErrTapiUnknown</u> Unbekannter Fehler <u>ErrTapiVersion</u> Falsche TAPI-Version

_ErrTapiBadAddr Kein Anschluss Wert -1814

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Tapi-Fehler

Ursache : Kein Anschluss, die Verbindung wurde zurückgewiesen.

_ErrTapiBusy

Besetzt

Wert -1813

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Tapi-Fehler

Ursache : Die angerufene Gegenstelle ist besetzt.

_ErrTapiCallID Ungültige Call-ID Wert -1819

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Tapi-Fehler

Ursache : Dem Befehl wurde eine ungültige Call-ID übergeben.

_ErrTapiCallState Operation in diesem Status unzulässig Wert -1817

Verwandte

Siehe Befehle,

ErrGet()

Kategorie : Tapi-Fehler

Ursache : Es wurde eine Operation ausgeführt, die in dem Status des Anrufs nicht

zulässig ist.

_ErrTapiDevName Ungültiges TAPI-Device Wert -1803

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Tapi-Fehler

Ursache : Name des Devices existiert nicht.

ErrTapiDialString Keine wählbare Nummer

Wert -1805

Verwandte

Siehe Befehle,

ErrGet()

Kategorie : Tapi-Fehler

Ursache : Es wurde keine Telefonnummer angegeben.

Der Fehler tritt auf, wenn in einem Wählstring illegale Zeichen (zum Beispiel

Buchstaben) enthalten sind.

_ErrTapiDialTimeout TAPI-Timeout überschritten Wert -1806

Verwandte

Siehe Befehle,

ErrGet()

Kategorie : Tapi-Fehler

Ursache : Das TAPI-Device meldet sich nicht.

Das TAPI-Device wurde angesprochen und es hat sich nicht innerhalb der nächsten 60

Sekunden gemeldet.

_ErrTapiFailed Operation nicht erfolgreich Wert -1810

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Tapi-Fehler

Ursache : Operation ist aus nicht näher spezifizierten Gründen fehlgeschlagen.

_ErrTapiInstall

TAPI-Schnittstelle nicht korrekt installiert

Wert -1807

Verwandte

Siehe Befehle,

ErrGet()

Kategorie : Tapi-Fehler

Ursache : Die TAPI-Schnittstelle ist nicht korrekt installiert worden.

_ErrTapiInUse TAPI-Gerätetreiber in Benutzung Wert -1804

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Tapi-Fehler

 $\mbox{ Ursache } \quad : \mbox{ Der TAPI-Ger\"{a}tetreiber wird von einem anderen Programm benutzt. }$

_ErrTapiMediaMode Nicht unterstützter Medienmodus Wert -1812

Verwandte

Siehe Befehle,

ErrGet()

Kategorie : Tapi-Fehler

Ursache : Der beim Modem eingestellte Medienmodus wird nicht unterstützt.

_ErrTapiMemory Kein Speicher Wert -1809

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Tapi-Fehler

Ursache : Es konnte nicht genug Hauptspeicher allokiert werden.

_ErrTapiNoConnect Nicht verbunden Wert -1815

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Tapi-Fehler

Ursache : Die Gegenstelle hebt nicht ab, die Gegenstelle lehnt eine Verbindung ab

oder es gibt kein Freizeichen.

ErrTapiNoListen

<u>TapiDevice</u> wird nicht überwacht

Wert -1821

Verwandte

Siehe Befehle,

ErrGet()

Kategorie : Tapi-Fehler

Ursache : Dem Befehl wurde ein <u>TapiDevice</u> übergeben, das nicht mit <u>TapiListen()</u>

überwacht wird.

_ErrTapiNotOwner Keine Besitzrechte

Wert -1820

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Tapi-Fehler

Ursache : Es wurde versucht eine TAPI-Funktion durchzuführen, die Besitzrechte

an dem TAPI-Gerät voraussetzt. Diese Besitzrechte sind nicht vorhanden.

_ErrTapiReinit TAPI-Konfiguration wurde verändert

Wert -1808

Verwandte

Siehe Befehle,

ErrGet()

Kategorie : Tapi-Fehler

Ursache : Die Konfiguration der TAPI-Schnittstelle wurde verändert.

_ErrTapiReject Anruf abgelehnt Wert -1816

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Tapi-Fehler

Ursache : Der entfernte Gesprächspartner hat den Anruf abgelehnt.

_ErrTapiUnavail Operation oder Verbindung nicht verfügbar Wert -1811

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Tapi-Fehler

Ursache : Es sollte eine Operation durchgeführt werden, die nicht von der

TAPI-Schnittstelle unterstützt wird.

_ErrTapiUnknown Unbekannter Fehler Wert -1801

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Tapi-Fehler

Ursache : Unbekannter oder interner Fehler.

Die TAPI-Schnittstelle liefert einen unbekannten oder internen Fehler zurück.

_ErrTapiVersion Falsche TAPI-Version Wert -1802

Verwandte

Siehe Befehle,

ErrGet()

Kategorie : Tapi-Fehler

Ursache : Die TAPI-Schnittstelle oder das TAPI-Device hat eine Versionsnummer,

die nicht unterstützt wird.

Die TAPI-Schnittstelle muss mindestens die Version 2.0 und der Gerätetreiber die Version 1.4 besitzen, um von CONZEPT 16 angesprochen zu werden.

Die TAPI-Version des Gerätetreibers kann über die Systemeigenschaft <u>Version</u> des Gerätetreibers ermittelt werden.

Konstanten für Dateibearbeitungsfehler (extern)

Siehe $\frac{\text{Alle}}{\text{Befehle}}$

<u>ErrFsiAccessDenied</u> Zugriff verweigert

<u>ErrFsiCurrentDir</u> Aktuelles Verzeichnis kann nicht gelöscht werden

<u>ErrFsiDriveInvalid</u> Ungültige Laufwerksangabe

<u>ErrFsiExists</u> Datei existiert bereits

<u>ErrFsiHdlInvalid</u> Ungültiger Dateideskriptor <u>ErrFsiInvalidFormat</u> Ungültiges Dateiformat

<u>ErrFsiLockViolation</u> Sperrkonflikt in externer Datei <u>ErrFsiNoFile</u> Externe Datei nicht vorhanden

<u>ErrFsiNoPath</u> Pfad nicht vorhanden

<u>ErrFsiOpenFailed</u> Externe Datei konnte nicht geöffnet oder angelegt werden

<u>ErrFsiOpenOverflow</u> Zuviele offene Dateien

<u>ErrFsiOther</u> Unbekannter Fehler beim Arbeiten mit externen Dateien

<u>ErrFsiReadFault</u> Fehler beim Lesen einer externen Datei

<u>ErrFsiSharingViolation</u> Zugriffskonflikt bei externer Datei

<u>ErrFsiWriteFault</u> Fehler beim Schreiben einer externen Datei

ErrFsiAccessDenied Zugriff verweigert

Wert -23

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie **Externe Dateioperation**

Ursache Auf die externe Datei oder das Verzeichnis kann aufgrund :

unzureichender Berechtigung nicht zugegriffen werden.

 $_$ ErrFsiCurrentDir Aktuelles Verzeichnis kann nicht gelöscht werden Wert -26

Verwandte

Siehe Befehle,

ErrGet()

Kategorie : Externe Dateioperation

Ursache : Es wurde versucht, das aktuelle Verzeichnis zu löschen.

_ErrFsiDriveInvalid Ungültige Laufwerksangabe Wert -25

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Externe Dateioperation

Ursache : Auf das angegebene Laufwerk kann nicht zugegriffen

werden.

ErrFsiExists

Datei existiert bereits

Wert -40

Verwandte

Siehe $\frac{\text{Befehle}}{\text{ErrGet()}}$,

<u>FsiCreateNew</u>

Kategorie **Externe Dateioperation**

Ursache Die externe Datei kann nicht angelegt werden, da sie bereits :

existiert.

_ErrFsiHdlInvalid Ungültiger Dateideskriptor Wert -24

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Externe Dateioperation

Ursache : Der benutzte Deskriptor des Betriebssystems ist ungültig.

_ErrFsiInvalidFormat Ungültiges Dateiformat Wert -35

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Externe Dateioperation

Ursache : Das Format der externen Datei kann nicht verwendet werden.

_ErrFsiLockViolation Sperrkonflikt in externer Datei Wert -28

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Externe Dateioperation

Ursache : Bei einer Sperr-, Lese- oder Schreiboperation ist ein Konflikt mit

existierenden Sperren anderer Benutzer aufgetreten.

ErrFsiNoFile

Externe Datei oder Verzeichnis nicht vorhanden

Wert -20

<u>Verwandte</u>

Siehe Befehle,

ErrGet()

Kategorie: Externe Dateioperation

Ursache : Die angegebene Datei oder das angegebene Verzeichnis wurde nicht

gefunden.

Bei dem Befehl <u>FsiPathChange()</u> wird dieser Fehler zurückgegeben, wenn das zuletzt angegebene Verzeichnis nicht vorhanden ist. Existiert der Pfad zu diesem Verzeichnis nicht, wird <u>ErrFsiNoPath</u> zurückgegeben.

_ErrFsiNoPath Pfad nicht vorhanden Wert -21

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Externe Dateioperation

Ursache : Der angegebene Pfad wurde nicht gefunden.

Der Pfad zu der angegebenen Datei oder dem Verzeichnis existiert nicht. Diese

Fehlermeldung trifft keine Aussage über die Existenz der Datei oder des

Verzeichnisses.

ErrFsiOpenFailed

Externe Datei konnte nicht geöffnet oder angelegt werden

Wert -29

Verwandte

Siehe Befehle,

ErrGet()

Kategorie : Externe Dateioperation Ursache : Datei existiert nicht.

Berechtigung unzureichend.

Dateiname ungültig.

_ErrFsiOpenOverflow Zuviele offene Dateien Wert -22

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Externe Dateioperation

Ursache : Die maximale Anzahl von offenen Dateien ist erreicht.

ErrFsiOther

Unbekannter Fehler beim Arbeiten mit externen Dateien

Wert -39

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: Externe Dateioperation

: Es trat ein Fehler beim Arbeiten mit externen Dateien auf, der nicht Ursache

durch andere Fehlerkonstanten abgedeckt ist

Die Fehlermeldung wird ebenfalls zurückgegeben, wenn ein ungültiger Dateiname

(zum Beispiel mit illegalen Zeichen) angegeben wurde.

_ErrFsiReadFault Fehler beim Lesen einer externen Datei Wert -31

Verwandte

Siehe Befehle,

ErrGet()

Kategorie : Externe Dateioperation

Ursache : Die externe Datei konnte nicht ordnungsgemäß gelesen werden

_ErrFsiSharingViolation Zugriffskonflikt bei externer Datei Wert -27

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: Externe Dateioperation

Ursache : Die externe Datei kann aufgrund der aktuellen Zugriffssituation nicht

geöffnet werden. Dieser Fall tritt beispielsweise ein, wenn ein Anwender die Datei im exklusiven Zugriff hat, und ein weiterer Benutzer diese

Datei öffnen möchte.

_ErrFsiWriteFault Fehler beim Schreiben einer externen Datei Wert -32

Verwandte

Siehe Befehle,

ErrGet()

Kategorie : Externe Dateioperation

Ursache : Die externe Datei konnte nicht ordnungsgemäß geschrieben werden

Konstanten für Fehler des OEM-Kits

 $Siehe \frac{\underline{Alle}}{\underline{Befehle}}$

<u>ErrOemDbaLock</u> Datenbank ist im Mehrbenutzerbetrieb <u>ErrOemInvalidFormat</u> Definitionsdatei im falschen Format

<u>ErrOemOpenDesigner</u> Prozedur wurde aus dem Designer gestartet
<u>ErrOemOpenFailed</u> Öffnen der Definitionsdatei fehlgeschlagen
<u>ErrOemOpenFrame</u> Ein oder mehrere Dialoge sind noch offen

<u>ErrOemOutOfSpace</u> Speicherplatz unzureichend

<u>ErrOemPassword</u> Kennwort ungültig oder OEM-Kit nicht vorhanden

ErrOemDbaLock Datenbank ist im Mehrbenutzerbetrieb Wert -1502

Verwandte

Siehe $\frac{\overline{\text{Befehle}},}{\overline{\text{OEMLoad()}},}$

ErrGet()

Kategorie : OEM-Kit

Ursache : Datenbank ist im Mehrbenutzerbetrieb.

ErrOemInvalidFormat Definitions date im falschen Format

Wert -1504

Verwandte

Siehe $\frac{\overline{\text{Befehle}},}{\overline{\text{OEMLoad()}},}$

ErrGet()

Kategorie: OEM-Kit

Ursache : Definitionsdatei hat nicht das korrekte Format oder wurde von einer

anderen CONZEPT 16-Clientversion erstellt.

_ErrOemOpenDesigner

Prozedur wurde aus dem Designer gestartet

Wert -1506

Verwandte

Siehe $\frac{\overline{\text{Befehle}},}{\overline{\text{OEMLoad()}},}$

ErrGet()

Kategorie : OEM-Kit

Ursache : Die Prozedur wurde aus dem Designer gestartet.

ErrOemOpenFailed

Öffnen der Definitionsdatei fehlgeschlagen

Wert -1503

<u>Verwandte</u>

Befehle,

Siehe OEMLoad(),

OEMSave(),

ErrGet()

Kategorie : OEM-Kit

Ursache : Öffnen der Definitionsdatei fehlgeschlagen.

ErrOemOpenFrame

Ein oder mehrere Dialoge sind noch offen

Wert -1507

Verwandte

Siehe $\frac{\overline{\text{Befehle}},}{\overline{\text{OEMLoad()}},}$

ErrGet()

Kategorie : OEM-Kit

Ursache : Ein oder mehrere Dialoge sind noch offen.

_ErrOemOutOfSpace Speicherplatz unzureichend Wert -1505

Verwandte

Siehe $\frac{\overline{\text{Befehle}},}{\overline{\text{OEMLoad()}},}$

ErrGet()

Kategorie : OEM-Kit

Ursache : Der Speicherplatz ist unzureichend.

ErrOemPassword

Kennwort ungültig oder OEM-Kit nicht vorhanden

Wert -1501

<u>Verwandte</u>

Befehle,

Siehe OEMLoad(),

OEMSave(),

ErrGet()

Kategorie : OEM-Kit

Ursache : OEM-Kit ist nicht vorhanden.

Kennwort des OEM-Kits ist ungültig.

Konstanten für Fehler der binären Objekte

 $Siehe \frac{\underline{Alle}}{\underline{Befehle}}$

<u>ErrBinData</u> Falsches Codewort angegeben <u>ErrBinDecryption</u> Falsches Codewort angegeben

ErrBinDirNotEmpty Verzeichnis ist nicht leer

<u>ErrBinExists</u> Binäres Objekt existiert bereits <u>ErrBinLocked</u> Das binäre Objekt ist gesperrt

<u>ErrBinNameInvalid</u> Ungültiger Name

<u>ErrBinNoData</u> Keine Daten vorhanden

<u>ErrBinNoFile</u> Binäres Objekt nicht vorhanden <u>ErrBinNoLock</u> Binäres Objekt nicht gesperrt

<u>ErrBinNoPath</u> Pfad nicht vorhanden

<u>ErrBinOperation</u> Operation kann nicht ausgeführt werden

ErrBinData

Falsches Codewort angegeben oder allgemeiner Fehler aufgetreten Wert -1509

Verwandte Befehle,

Siehe ErrGet(),

 $\underline{\underline{\hspace{0.1cm}} ErrBinDecryption}$

Kategorie: Binäre Objekte

Ursache : Es ist kein oder der falsche Verschlüsselungscode angegeben worden

oder ein allgemeiner Fehler aufgetreten.

ErrBinDecryption Falsches Codewort angegeben Wert -1512

Verwandte

Siehe $\frac{\text{Befehle}}{\text{ErrGet()}}$,

<u>ErrBinData</u>

Kategorie: Binäre Objekte

: Es ist kein oder der falsche Verschlüsselungscode angegeben worden. Ursache

Dieser Fehler tritt nur auf, wenn bei BinExport() oder BinReadMem() die

Option BinErrorDecryption angegeben wurde.

_ErrBinDirNotEmpty Verzeichnis ist nicht leer Wert -1508

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Binäre Objekte

Ursache : Die verwendete Operation kann nur bei einem leeren Verzeichnis

durchgeführt werden.

ErrBinExists

Binäres Objekt existiert bereits

Wert -1507

Verwandte

Siehe $\frac{\text{Befehle}}{\text{BinOpen()}}$,

ErrGet()

Kategorie: Binäre Objekte

Ursache : Es existiert bereits ein binäres Objekt mit dem gleichen Namen.

_ErrBinLocked Das binäre Objekt ist gesperrt Wert -1506

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: Binäre Objekte

Ursache : Das binäre Objekt ist von einem anderen Benutzer gesperrt. Die

Operation kann aber nur dann durchgeführt werden, wenn das Objekt

nicht oder vom eigenen Benutzer gesperrt ist.

ErrBinNameInvalid Ungültiger Name Wert -1501

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: Binäre Objekte

Ursache

: Der angegebene Name ist ungültig. In Namen von binären Objekten dürfen keine Steuerzeichen oder die Zeichen "*" und "?" vorkommen. Beim Anlegen eines binären Verzeichnisses, darf der Name nicht auf "\"

enden.

ErrBinNoData

Keine Daten vorhanden

Wert -1504

<u>Verwandte</u>

Befehle,

Siehe BinImport(),

BinExport(),

ErrGet()

Kategorie: Binäre Objekte

Ursache : Die angegebene externe Datei oder das angegebene Objekt beinhaltet

keine Daten.

_ErrBinNoFile Binäres Objekt nicht vorhanden Wert -1503

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: Binäre Objekte

Ursache : Das angegebene binäre Objekt konnte nicht gefunden werden.

Der Fehlerwert wird auch dann zurückgegeben, wenn ein binäres Verzeichnis nicht geöffnet werden konnte. Wird ein Problem schon in der Pfadangabe zu dem binären Objekt festgestellt, wird <u>ErrBinNoPath</u> zurückgegeben.

_ErrBinNoLock Binäres Objekt nicht gesperrt Wert -1505

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: Binäre Objekte

Ursache : Das binäre Objekt wurde nicht gesperrt. Die Operation kann aber nur bei

einem gesperrten Objekt durchgeführt werden.

_ErrBinNoPath Pfad nicht vorhanden Wert -1502

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: Binäre Objekte

Ursache : Der angegebene Pfad zu dem binären Objekt existiert nicht.

_ErrBinOperation Operation kann nicht ausgeführt werden Wert -1510

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Binäre Objekte

Ursache : Es wurde versucht ein binäres Objekt im Wurzelverzeichnis anzulegen.

Es wurde versucht ein binäres Objekt zwischen zwei Datenbanken zu

kopieren oder zu verschieben.

Die maximale Anzahl von 60 Ebenen wurde überschritten.

Konstanten für Fehler des Druckprozessors

 $Siehe \frac{\underline{Alle}}{\underline{Befehle}}$

<u>ErrPpcAcrobat</u> Fehler beim Erzeugen eines PDF-Dokuments

<u>ErrPpcArgument</u> Argumente fehlerhaft

<u>ErrPpcDriver</u> Falscher Treiber

<u>ErrPpcFileCreate</u> Fehler beim Erzeugen der Datei <u>ErrPpcFileOpen</u> Fehler beim Öffnen der Datei

<u>ErrPpcFileRead</u> Fehler beim Lesen <u>ErrPpcFileWrite</u> Fehler beim Schreiben

<u>ErrPpcGhostscript</u> Fehler beim Erzeugen eines PDF-Dokuments

<u>ErrPpcPrint</u> Fehler beim Drucken

_ErrPpcAcrobat Fehler beim Erzeugen eines PDF-Dokuments Wert -9074

<u>Verwandte Befehle,</u> Siehe <u>PpcMakeAcrobatPdf(),</u> <u>ErrGet()</u>

Kategorie: Druckprozessor

Ursache : Beim Erzeugen eines PDF-Dokuments über den Acrobat Distiller ist ein

Fehler aufgetreten. Der Distiller schreibt ein eigenes Fehlerprotokoll,

dass über die Ursache aufschluss geben kann.

ErrPpcArgument Argumente fehlerhaft

Wert -9034

Verwandte Befehle,

PpcMakePreviewBmp(),

Siehe PreviewJpg(), PpcMakePreviewPng(),

PpcMakePreviewTif(),

ErrGet()

Kategorie: Druckprozessor

Ursache : Ein oder mehrere der angegebenen Argumente sind nicht korrekt.

ErrPpcDriver Falscher Treiber Wert -9070

Verwandte Befehle,

Siehe $\frac{\text{PpcMakePdf()}}{\text{PpcMakeAcrobatPdf()}}$,

ErrGet()

Kategorie: Druckprozessor

: Es wurde der Druckjob eines falschen Druckers verwendet, um ein Ursache

PDF-Dokument zu erzeugen. Ein PDF-Dokument kann nur von einem Postscript-Druckjob, wie ihn der CONZEPT 16-PDF-Drucker erzeugt,

generiert werden.

ErrPpcFileCreate

Fehler beim Erzeugen der Datei

Wert -9054

Verwandte Befehle,

PpcMakePreviewBmp(),

PpcMakePreviewJpg(),

Siehe PpcMakePreviewPng(),

PpcMakePreviewTif(),

PpcMakeTif(),

ErrGet()

Kategorie: Druckprozessor

Ursache : Beim Erzeugen der externen Vorschau-Datei ist ein Fehler aufgetreten.

_ErrPpcFileOpen

Fehler beim Öffnen der Datei

Wert -9051

Verwandte Befehle,

PpcMakePreviewBmp(),

PpcMakePreviewJpg(),

Siehe PpcMakePreviewPng(),

PpcMakePreviewTif(),

PpcMakeTif(),

ErrGet()

Kategorie : Druckprozessor

Ursache : Beim Öffnen der externen Vorschau-Datei ist ein Fehler aufgetreten.

_ErrPpcFileRead Fehler beim Lesen

Wert -9052

<u>Verwandte Befehle</u>, <u>PpcMakePreviewBmp()</u>, <u>PpcMakePreviewIpg()</u>,

Siehe PpcMakePreviewPng(),

PpcMakePreviewTif(),

PpcMakeTif(),

ErrGet()

Kategorie : Druckprozessor

Ursache : Beim Lesen der externen Vorschau-Datei ist ein Fehler aufgetreten.

_ErrPpcFileWrite Fehler beim Schreiben

Wert -9053

Verwandte Befehle,

PpcMakePreviewBmp(),

PpcMakePreviewJpg(),

Siehe PpcMakePreviewPng(),

PpcMakePreviewTif(),

PpcMakeTif(),

ErrGet()

Kategorie : Druckprozessor

Ursache : Beim Schreiben der externen Vorschau-Datei ist ein Fehler aufgetreten.

ErrPpcGhostscript Fehler beim Erzeugen eines PDF-Dokuments Wert -9015

<u>Verwandte</u>

Siehe $\frac{\text{Befehle}}{\text{PpcMakePdf()}}$,

ErrGet()

Kategorie: Druckprozessor

: Beim Erzeugen eines PDF-Dokuments über Ghostscript ist ein Fehler

aufgetreten.

Diese Fehlermeldung wird zum Beispiel dann zurückgegeben, wenn bei den PpcMake...-Befehlen eine Datei auf einem Netzwerkpfad erzeugt werden soll. Da der Druckprozessor als Dienst läuft, hat er in der Regel keinen Zugriff auf verbundene Laufwerke.

_ErrPpcPrint Fehler beim Drucken Wert -9081

Verwandte

Siehe $\frac{\overline{Befehle}}{PpcPrint()}$,

ErrGet()

Kategorie: Druckprozessor

Ursache : Beim Drucken des Druckjobs ist ein Fehler aufgetreten.

Konstanten für Fehler bei der Verarbeitung von PDF-Dokumenten

Siehe $\frac{\text{Alle}}{\text{Befehle}}$

<u>ErrPdfImageFormat</u> Ungültiges Format

<u>ErrPdfInsertMetaFile</u> Fehler beim Einfügen von Informationen

<u>ErrPdfNotLicensed</u> Erstellung eines PDF-Dokuments mit einer Standard- oder

Advanced-Edition

<u>ErrPdfNotPdfA</u> Fehler bei der Erzeugung eines PDF/A-Dokuments

<u>ErrPdfPageAppend</u> Seite konnte nicht hinzugefügt werden <u>ErrPdfPageClosed</u> Keine Seite zum Bearbeiten geöffnet

<u>ErrPdfPageNotExisting</u> Seite nicht vorhanden <u>ErrPdfPassword</u> Ungültiges Kennwort

_ErrPdfImageFormat Ungültiges Format Wert -2501

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: PDF-Verarbeitung

Ursache : Das angegebene Bild hat ein ungültiges Format.

ErrPdfInsertMetaFile

Fehler beim Einfügen von Informationen

Wert -2506

Verwandte

Siehe $\frac{\text{Befehle}}{\text{PrtJobClose()}}$,

PdfInsertMeta()

Kategorie: PDF-Verarbeitung

: Es ist ein Fehler beim Einfügen der Informationen in ein PDF-Dokument

aufgetreten.

ErrPdfNotLicensed

Erstellung eines PDF-Dokuments mit einer Standard- oder Advanced-Edition Wert -2508

Verwandte

Siehe Befehle,

PrtJobClose()

Kategorie: PDF-Verarbeitung

Ursache : Es wurde ein PDF-Dokument mit einer CONZEPT 16-Standard- oder

Advance-Edition erstellt. Das Dokument wurde gedruckt, aber mit dem Wasserzeichen "DRAFT" versehen. Zum Erstellen eines PDF-Dokuments wird bis zur Version 5.7.06 eine Enterprise-Edition benötigt (siehe auch

Lizenztypen).

Ab der Version 5.7.06 tritt dieser Fehler nicht mehr auf.

_ErrPdfNotPdfA

Fehler bei der Erzeugung eines PDF/A-Dokuments

Wert -2507

Verwandte

Siehe Befehle,

PrtIobClose()

Kategorie : PDF-Verarbeitung

Ursache : Es konnte kein PDF/A-Dokument erzeugt werden, weil nicht alle

Informationen in das Dokument eingebettet werden konnten.

_ErrPdfPageAppend Seite konnte nicht hinzugefügt werden Wert -2505

Verwandte

Siehe <u>Befehle</u>,

PrtIobClose()

Kategorie: PDF-Verarbeitung

Ursache : Bei der Erstellung eines PDF-Dokuments konnte eine Seite nicht erzeugt

werden.

_ErrPdfPageClosed Keine Seite zum Bearbeiten

Keine Seite zum Bearbeiten geöffnet

Wert -2503

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: PDF-Verarbeitung

Ursache : Die Anweisung erwartet eine zum Bearbeiten geöffete Seite eines

PDF-Dokuments.

_ErrPdfPageNotExisting Seite nicht vorhanden Wert -2504

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : PDF-Verarbeitung

Ursache : Die bei der Anweisung angegebene Seite des PDF-Dokuments ist nicht

vorhanden.

ErrPdfPassword Ungültiges Kennwort Wert -2502

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: PDF-Verarbeitung

Ursache : Beim Öffnen eines PDF-Dokuments wurde ein falsches Kennwort

angegeben.

Konstanten für Fehler bei regulären Ausdrücken

Siehe $\frac{\text{Alle}}{\text{Befehle}}$

<u>ErrRegExBadEscapeSequence</u> Nicht aufgelöste Escape-Sequenz im Ausdruck

<u>ErrRegExBadInterval</u> Fehler im {min,max} Intervall

<u>ErrRegExInvalidBackRef</u> Rückbezug auf eine nicht vorhandene Referenz

<u>ErrRegExInvalidFlag</u> Ungültiger Modus

<u>ErrRegExInvalidRange</u> In einer Zeichenmenge [x-y] ist x größer als y

<u>ErrRegExLookBehindLimit</u> Rückschau Ausdrücke müssen eine beschränkte

maximale Länge haben

<u>ErrRegExMaxLtMin</u> Im Intervall {min,max} ist max kleiner als min ErrRegExMismatchedParentheses Falsch verschachtelte Klammern im regulären

Ausdruck

<u>ErrRegExMissingCloseBracket</u> Fehlende schließende Klammer in einem

Klammerausdruck

<u>ErrRegExNotSupported</u> Verwendung einer Funktion, die nicht unterstützt

wird

<u>ErrRegExNumberTooBig</u> Dezimalzahl zu groß

<u>ErrRegExOctalTooBig</u> Oktale Zahl muss kleiner oder gleich 0377 sein

<u>ErrRegExPropertySyntax</u> Ungültige Unicode-Eigenschaft

<u>ErrRegExRuleSyntax</u> Syntaxfehler im regulären Ausdruck

<u>ErrRegExSetContainsString</u> Reguläre Ausdrücke können keine UnicodeSets

mit Zeichenketten beinhalten

<u>ErrRegExStackOverflow</u> Stapelüberlauf in der Ablaufverfolgung des

regulären Ausdrucks

<u>ErrRegExTimeout</u> Maximale Suchzeit überschritten

_ErrRegExBadEscapeSequence Nicht aufgelöste Escape-Sequenz im Ausdruck Wert -2703

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Reguläre Ausdrücke

Bedeutung : Nicht aufgelöste Escape-Sequenz im Ausdruck

_ErrRegExBadInterval Fehler im {min,max} Intervall Wert -2708

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Reguläre Ausdrücke

Bedeutung : Fehler im {min,max} Intervall

_ErrRegExInvalidBackRef Rückbezug auf eine nicht vorhandene Referenz Wert -2710

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Reguläre Ausdrücke

Bedeutung: Rückbezug auf eine nicht vorhandene Referenz

_ErrRegExInvalidFlag Ungültiger Modus Wert -2711

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Reguläre Ausdrücke Bedeutung : Ungültiger Modus

_ErrRegExInvalidRange In einer Zeichenmenge [x-y] ist x größer als y Wert -2716

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Reguläre Ausdrücke

Bedeutung : In einer Zeichenmenge [x-y] ist x größer als y

_ErrRegExLookBehindLimit

Rückschau Ausdrücke müssen eine beschränkte maximale Länge haben Wert -2712

Veit -2/12

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Reguläre Ausdrücke

Bedeutung : Rückschau Ausdrücke müssen eine beschränkte maximale Länge haben

_ErrRegExMaxLtMin Im Intervall $\{min,max\}$ ist max kleiner als min Wert -2709

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Reguläre Ausdrücke

Bedeutung: Im Intervall {min,max} ist max kleiner als min

_ErrRegExMismatchedParentheses Falsch verschachtelte Klammern im regulären Ausdruck Wert -2706

<u>Verwandte</u>

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Reguläre Ausdrücke

Bedeutung : Falsch verschachtelte Klammern im regulären Ausdruck

_ErrRegExMissingCloseBracket Fehlende schließende Klammer in einem Klammerausdruck Wert -2715

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Reguläre Ausdrücke

Bedeutung : Fehlende schließende Klammer in einem Klammerausdruck

_ErrRegExNotSupported Verwendung einer Funktion, die nicht unterstützt wird Wert -2705

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Reguläre Ausdrücke

Bedeutung: Verwendung einer Funktion, die nicht unterstützt wird

_ErrRegExNumberTooBig Dezimalzahl zu groß Wert -2707

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Reguläre Ausdrücke Bedeutung : Dezimalzahl zu groß

_ErrRegExOctalTooBig Oktale Zahl muss kleiner oder gleich 0377 sein Wert -2714

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Reguläre Ausdrücke

Bedeutung: Oktale Zahl muss kleiner oder gleich 0377 sein

_ErrRegExPropertySyntax Ungültige Unicode-Eigenschaft Wert -2704

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Reguläre Ausdrücke

Bedeutung : Ungültige Unicode-Eigenschaft

_ErrRegExRuleSyntax Syntaxfehler im regulären Ausdruck Wert -2702

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Reguläre Ausdrücke

Bedeutung : Syntaxfehler im regulären Ausdruck

 $_ErrRegExSetContainsString$

Reguläre Ausdrücke können keine UnicodeSets mit Zeichenketten beinhalten Wert -2713

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Reguläre Ausdrücke

Bedeutung : Reguläre Ausdrücke können keine UnicodeSets mit Zeichenketten

beinhalten

_ErrRegExStackOverflow Stapelüberlauf in der Ablaufverfolgung des regulären Ausdrucks Wert -2717

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Reguläre Ausdrücke

Bedeutung : Stapelüberlauf in der Ablaufverfolgung des regulären Ausdrucks

_ErrRegExTimeout Maximale Suchzeit überschritten Wert -2718

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Reguläre Ausdrücke

Bedeutung: Maximale Suchzeit überschritten

Konstanten für Fehler der ODBC-Schnittstelle

 $Siehe \frac{\underline{Alle}}{\underline{Befehle}}$

<u>ErrOdbcEnvironment</u> ODBC-Umgebung konnte nicht initialisiert werden

<u>ErrOdbcError</u> Ausführung eines Statements fehlgeschlagen

 $\underline{\underline{ErrOdbcFunctionFailed}}\ Funktions aufruf\ fehlgeschlagen$

<u>ErrOdbcIncomplete</u> Bibliothek unvollständig

<u>ErrOdbcNoData</u> Keine (weiteren) Daten vorhanden

<u>ErrOdbcNotFound</u> Bibliothek nicht geladen

<u>ErrOdbcWarning</u> Warnung bei der Ausführung eines Statements

_ErrOdbcEnvironment ODBC-Umgebung konnte nicht initialisiert werden Wert -553

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: ODBC-Schnittstelle

Ursache : Die ODBC-Umgebung konnte nicht initialisiert werden.

 $_$ ErrOdbcError Ausführung eines Statements fehlgeschlagen Wert $\mbox{-}555$

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: ODBC-Schnittstelle

Dieser Fehler wird zurückgegeben, wenn bei der Ausführung eines ODBC-Statements ein Fehler aufgetreten ist. Genauere Informationen

Ursache : Über den Fehler können über die OdbcErr...-Eigenschaften des

entsprechenden Objekts ermittelt werden.

ErrOdbcFunctionFailed Funktionsaufruf fehlgeschlagen Wert -554

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: ODBC-Schnittstelle

Ursache : Dieser Fehlerwert wird von ODBC-Befehle zurückgegeben, wenn ein Objekt nicht erzeugt werden konnte.

ErrOdbcIncomplete Bibliothek unvollständig Wert -552

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: ODBC-Schnittstelle

Ursache : Es sind nicht alle notwendigen Funktionen in der dynamischen Link-Bibliothek (odbc32.dll) vorhanden.

ErrOdbcNoData Keine (weiteren) Daten vorhanden Wert -557

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: ODBC-Schnittstelle

Ursache : Dieser Fehlerwert wird zurückgegeben, wenn keine oder keine weiteren Daten mehr vorhanden sind.

_ErrOdbcNotFound Bibliothek nicht geladen Wert -551

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: ODBC-Schnittstelle

Ursache : Die dynamische Link-Bibliothek (odbc32.dll) konnte nicht geladen werden.

ErrOdbcWarning Warnung bei der Ausführung eines Statements Wert -556

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: ODBC-Schnittstelle

Dieser Fehlerwert wird zurückgegeben, wenn bei der Ausführung eines

ODBC-Statements eine Warnung aufgetreten ist. Genauere Informationen

Ursache : können über die OdbcErr...-Eigenschaften des entsprechenden Objekts

ermittelt werden.

Konstanten für Fehler der Benutzerverwaltung Siehe $\frac{\text{Alle}}{\text{Befehle}}$

<u>ErrUrmObjectNotFound</u> Objekt nicht gefunden

<u>ErrUrmParentNotFound</u> Elternobjekt nicht gefunden

_ErrUrmObjectNotFound Objekt nicht gefunden Wert -2301

Verwandte

Siehe Befehle,

ErrGet()

Kategorie: Benutzerverwaltung

Ursache : Das angegebene Objekt wurde nicht gefunden.

_ErrUrmParentNotFound Elternobjekt nicht gefunden Wert -2302

Verwandte

Siehe Befehle,

ErrGet()

Kategorie: Benutzerverwaltung

Ursache : Das angegebene Eltern-Objekt ist nicht mehr vorhanden.

Dieser Fehler kann nur auftreten, wenn zwischen dem Öffnen des Eltern-Objekts und der Verarbeitung eines anderen Befehls, bei dem das Eltern-Objekt als Referenz angegeben ist, das Eltern-Objekt gelöscht wurde. Die Löschung kann durch eine gleichzeitig verarbeitete Prozedur auf dem gleichen Client oder durch die Prozedur eines anderen Clients erfolgen. Der Befehl kann durch das Fehlen des Eltern-Objekts nicht mehr durchgeführt werden.

Konstanten für Fehler bei XML-Befehlen

 $Siehe\,\frac{\underline{Alle}}{\underline{Befehle}}$

<u>ErrXmlFatal</u> Kritischer Fehler <u>ErrXmlNotValid</u> Schema ungültig <u>ErrXmlRecoverable</u> Behebbarer Fehler

<u>ErrXmlWarning</u> Warnung

_ErrXmlWarning

Warnung

Wert -2401

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: XML-Verarbeitung

Bei der Verarbeitung der XML-Datei ist eine Warnung aufgetreten.

Ursache : Weitere Informationen über die Warnung können mit der Anweisung

XmlError() ermittelt werden.

_ErrXmlRecoverable Behebbarer Fehler Wert -2402

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: XML-Verarbeitung

Bei der Verarbeitung der XML-Datei ist ein behebbarer Fehler

Ursache : aufgetreten. Weitere Informationen über den Fehler können mit der

Anweisung XmlError() ermittelt werden.

_ErrXmlFatal Kritischer Fehler Wert -2403

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: XML-Verarbeitung

Bei der Verarbeitung der XML-Datei ist ein kritischer Fehler aufgetreten.

Ursache : Weitere Informationen über den Fehler können mit der Anweisung

XmlError() ermittelt werden.

_ErrXmlNotValid Schemadatei ungültig Wert -2404

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: XML-Verarbeitung

Bei der Verarbeitung der XML-Datei ist die Schemadatei ungültig.

Ursache : Weitere Informationen über den Fehler können mit der Anweisung

XmlError() ermittelt werden.

Konstanten für Fehler der Validierungselemente

 $Siehe \frac{\underline{Alle}}{\underline{Befehle}}$

<u>ErrVldExists</u> <u>Validierungselement</u> existiert bereits <u>ErrVldLocked</u> Das <u>Validierungselement</u> ist gesperrt

_ErrVldNameInvalid Ungültiger Name

<u>ErrVldNoFile</u> <u>Validierungselement</u> nicht vorhanden <u>ErrVldNoLock</u> <u>Validierungselement</u> nicht gesperrt

ErrVldExists Validierungselement existiert bereits Wert -1507

Verwandte

Siehe $\frac{\text{Befehle}}{\text{VldOpen()}}$,

ErrGet()

Kategorie: Validierungselemente

Ursache : Es existiert bereits ein Validierungselement mit dem gleichen Namen.

_ErrVldLocked Das Validierungselement ist gesperrt Wert -1506

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Validierungselemente

Ursache : Das <u>Validierungselement</u> ist von einem anderen Benutzer gesperrt. Die

Operation kann aber nur dann durchgeführt werden, wenn das Element

nicht oder vom eigenen Benutzer gesperrt ist.

_ErrVldNameInvalid Ungültiger Name Wert -1501

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Validierungselemente

Ursache : Der angegebene Name ist ungültig. In Namen von

Validierungselementen dürfen keine Steuerzeichen oder die Zeichen "*"

und "?" vorkommen.

_ErrVldNoFile Validierungselement nicht vorhanden Wert -1503

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Validierungselemente

Ursache : Das angegebene <u>Validierungselement</u> konnte nicht gefunden werden.

_ErrVldNoLock Validierungselement nicht gesperrt Wert -1505

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie : Validierungselemente

Ursache : Das <u>Validierungselement</u> wurde nicht gesperrt. Die Operation kann aber

nur bei einem gesperrten Element durchgeführt werden.

Weitere Fehlerkonstanten Siehe $\frac{\text{Alle}}{\text{Befehle}}$

<u>ErrPrtPaperFormat</u> Falsches Papierformat

<u>ErrRtfSyntaxError</u> Syntax Fehler beim Lesen vom RTF-Format

ErrPrtPaperFormat Falsches Papierformat

Wert -1601

Verwandte

Siehe $\frac{\text{Befehle}}{\text{PrtJobClose()}}$,

ErrGet()

Kategorie : Drucken

Ursache : Das im Druckjob angegebene Papierformat wird durch die

Ausgabeeinheit nicht unterstützt.

_ErrRtfSyntaxError Syntax Fehler beim Lesen vom RTF-Format Wert -1701

Verwandte

Siehe <u>Befehle</u>,

ErrGet()

Kategorie: RTF-Verarbeitung

Ursache : Es existiert ein syntaktischer Fehler in der RTF-Datei.

Der Fehler kann beim Laden oder Speichern einer Datei im RTF-Format auftreten.

Liste aller Prozedurbefehle Liste aller Prozedurbefehle Siehe Befehlsgruppen

A - B - C - D - E - F - G - H - I - I - K - L - M - N - O - P - R - S - T - U - V - W - X

- Abs
- Acos
- <u>and</u>
- Asin
- Atan
- BinClose
- BinCopy
- BinDelete
- BinDirDelete
- BinDirMove
- BinDirOpen
- BinDirRead
- BinExport
- BinImport
- BinMove
- BinOpen
- BinReadMem
- BinRename
- BinUpdate
- BinWriteMem
- break
- Call
- CallOld
- Ceil
- ChartClose
- ChartDataAdd
- ChartDataClose
- ChartDataOpen
- ChartDataSort
- ChartOpen
- ChartSave
- ClipBoardRead
- ClipBoardWrite
- CnvAB
- CnvAD
- CnvAF
- CnvAI
- CnvAM
- CnvAT
- CnvBA
- CnvBC

- CnvBF
- CnvBI
- CnvBM
- CnvCB
- CnvDA
- CnvDI
- CnvFA
- CnvFB
- CnvFI
- CnvFM
- CnvIA
- CnvIB
- CnvID
- CnvIF
- CnvIL
- CnvIM
- CnvIT
- CnvLI
- CnvMA
- CnvMB
- CnvMF
- CnvMI
- CnvTA
- CnvTI
- ColorMake
- ColorRgbMake
- ComCall
- ComCallResult
- ComClose
- ComEvtProcessGet
- ComEvtProcessSet
- ComInfo
- ComOpen
- ComPropGet
- ComPropGetText
- ComPropSet
- ComPropSetText
- Cos
- CteClear
- CteClose
- CteDelete
- CteInfo
- CteInsert
- CteInsertItem
- CteInsertNode
- CteNodeValueAlpha
- CteOpen
- CteRead
- cvcle

- DateDay
- <u>DateDayOfWeek</u>
- DateMake
- DateMonth
- DateYear
- DbaConnect
- <u>DbaControl</u>
- DbaDisconnect
- <u>DbaInfo</u>
- DbaKeyRebuild
- **DbaLicense**
- DbaLog
- DbaName
- DbgConnect
- <u>DbgControl</u>
- **DbgDisconnect**
- <u>DbqDump</u>
- DbqStop
- DbgTrace
- <u>DdeCommand</u>
- <u>DdeConnect</u>
- DdeDisconnect
- DdeGetData
- <u>DdeGetDataInfo</u>
- DdeInit
- DdeServiceClose
- DdeServiceData
- <u>DdeServiceOpen</u>
- DdeServiceRead
- DdeSetData
- DdeTerm
- Dec
- declare
- define
- div
- DllCall
- DllLoad
- DllUnload
- do...while
- <u>DtaBegin</u>
- DtaCommit
- DtaRollback
- ErrCall
- ErrGet
- ErrIgnore
- ErrMapText
- ErrPos
- ErrSet
- ErrThrowProc

- ErrTryCatch
- ErrTryIgnore
- exit
- <u>Exp</u>
- FileInfo
- FileInfoByName
- FileName
- <u>Fld...</u>
- Fld...ByName
- FldAttributes
- FldCompare
- FldCopy
- FldDef
- FldDefBvName
- FldInfo
- FldInfoByName
- FldName
- Floor
- FontMake
- for...loop...while/until
- Fract
- FsiAttributes
- FsiClose
- FsiDate
- FsiDelete
- FsiDirClose
- FsiDirOpen
- FsiDirRead
- FsiDiskInfo
- FsiFileCompress
- FsiFileInfo
- FsiFileProcess
- FsiFileUncompress
- FsiLock
- FsiMark
- FsiMonitorAdd
- FsiMonitorClose
- FsiMonitorControl
- FsiMonitorOpen
- FsiMonitorRemove
- FsiOpen
- FsiPath
- FsiPathChange
- FsiPathCreate
- FsiPathDelete
- FsiRead
- FsiReadMem
- FsiRename
- FsiSeek

- FsiSeek64
- FsiSize
- FsiSize64
- FsiSplitName
- FsiTime
- FsiWrite
- FsiWriteMem
- global
- handle
- HdlEnum
- <u>HdlInfo</u>
- HdlLink
- HttpClose
- <u>HttpGetData</u>
- HttpOpen
- if...else
- <u>Inc</u>
- <u>IobClose</u>
- <u>IobControl</u>
- <u>JobEvent</u>
- <u>IobOpen</u>
- <u>IobSleep</u>
- <u>IobStart</u>
- IsonLoad
- IsonSave
- KeyFldInfo
- <u>KeyInfo</u>
- KeyInfoByName
- KevName
- LinkFldInfo
- LinkInfo
- LinkInfoBvName
- LinkName
- local
- LocaleLoad
- LocaleSelect
- LocaleUnload
- <u>Log10</u>
- <u>Log2</u>

- LoaN
- MailClose
- MailData
- MailOpen
- main
- Max
- MemAllocate
- MemCnv
- MemCompress
- MemCopy
- MemDecrypt
- MemEncrypt
- MemFindByte
- MemFindStr
- MemFree
- MemGenKeyPair
- MemHash
- MemHMAC
- MemReadByte
- MemReadStr
- MemSign
- MemUncompress
- MemVerify
- MemWriteByte
- MemWriteStr
- Min
- Modulo-Operator
- MsxClose
- MsxOpen
- MsxRead
- MsxReadMem
- MsxWrite
- MsxWriteMem
- NetInfo
- OdbcCatalogClm
- OdbcCatalogTbl
- OdbcClmData
- OdbcClmInfo
- OdbcClose
- OdbcConnect
- OdbcConnectDriver
- OdbcDataSources
- OdbcExecute
- OdbcExecuteDirect
- OdbcFetch

- OdbcOpen
- OdbcParamAdd
- OdbcParamSet
- OdbcPrepare
- OEMLoad
- OEMSave
- or
- PdfAttachExportFile
- PdfAttachExportMem
- PdfAttachFile
- PdfAttachInfoGet
- PdfAttachMem
- PdfClose
- PdfInsertImage
- PdfInsertMeta
- PdfNew
- PdfOpen
- PdfPageClose
- PdfPageOpen
- PdfSavePage
- PdfTextColor
- PdfTextExtractMem
- PdfTextFont
- PdfTextWrite
- PicExifInfo
- PointMake
- Pow
- PpcMakeAcrobatPdf
- PpcMakeEps
- PpcMakePdf
- PpcMakePreviewBmp
- PpcMakePreviewIpq
- PpcMakePreviewPng
- PpcMakePreviewTif
- PpcMakeTif
- PpcPrint
- ProcAdvertise
- ProcCacheClear
- ProcCompile
- PrtAdd
- PrtAddByName
- PrtDeviceClose
- PrtDeviceOpen
- PrtDeviceRefresh
- PrtFormClose
- PrtFormOpen
- PrtInfo
- PrtInfoStr
- PrtIobClose

- PrtIobOpen
- PrtJobWrite
- PrtPrinterRefresh
- PrtPropGet
- PrtPropSet
- PrtRtfSearch
- PrtSearch
- PrtUnit
- PrtUnitLog
- Random
- RangeMake
- RecBufClear
- RecBufCompare
- RecBufCompareFld
- RecBufCopy
- RecBufCreate
- RecBufDefault
- RecBufDestroy
- RecDelete
- RecDeleteAll
- RecFilterAdd
- RecFilterCreate
- RecFilterDestroy
- RecInfo
- RecInsert
- RecLink
- RecLinkInfo
- RecRead
- RecReplace
- RectMake
- Rescale
- return
- RmtCall
- RmtDataRead
- RmtDataSearch
- RmtDataWrite
- Rnd
- RtfTabMake
- SbrClear
- SbrCompare
- SbrCopy
- SbrInfo
- SbrInfoByName
- SbrName
- SbrStatus
- SckClose
- SckConnect

- SckInfo
- SckListen
- SckRead
- SckReadMem
- SckStartTls
- SckWrite
- SckWriteMem
- SelAddLink
- SelAddSortFld
- SelClear
- SelClose
- SelCopy
- SelCreate
- SelDefOuery
- <u>SelDelete</u>
- SelIgnore
- SelInfo
- SelInfoAlpha
- SelInfoDate
- SelInfoTime
- SelOpen
- SelRead
- <u>SelRecDelete</u>
- SelRecInsert
- SelRun
- SelStore
- SelValue
- <u>Sqn</u>
- Sin
- Sart
- StoClose
- StoDirOpen
- StoDirRead
- StoExport
- StoImport
- StoImportTile
- StoOpen
- StoReadMem
- StoWriteMem
- StoWriteTileMem
- StrAdi
- StrChar
- StrCnv
- StrCut
- StrDecrypt
- StrDel
- <u>StrEncrypt</u>
- StrFind
- StrFmt
- StrIns
- StrLen

- StrToChar
- <u>sub</u>
- SvcSessionControl
- switch...case...default
- SysBeep
- SysDate
- SysExecute
- SysGetArg
- SysGetEnv
- SysOS
- SysPropGet
- SysPropSet
- SysSleep
- SysTics
- SysTime
- SysTimerClose
- SysTimerCreate
- Tan
- TapiCall
- TapiClose
- <u>TapiConference</u>
- <u>TapiConferenceCommit</u>
- <u>TapiConferenceDial</u>
- TapiDevice
- TapiDial
- TapiForward
- <u>TapiInfo</u>
- TapiListen
- TapiOpen
- TapiPickup
- TextClear
- TextClose
- TextCopy
- TextCreate
- TextDelete
- TextEdit
- TextInfo
- TextInfoAlpha
- TextInfoDate
- TextInfoTime
- TextLineRead
- TextLineWrite
- TextOpen
- TextRead
- TextRename
- TextSearch
- TextWrite
- TimeHour
- TimeHSec

- TimeMake
- TimeMin
- TimeSec
- <u>Trn</u>
- <u>trv</u>
- trysub
- <u>UrmClose</u>
- UrmCreate
- <u>UrmDelete</u>
- UrmOpen
- UrmPermGet
- <u>UrmPermGetRaw</u>
- <u>UrmPermSet</u>
- <u>UrmPropGet</u>
- <u>UrmPropSet</u>
- <u>UrmPropType</u>
- UrmRead
- UserClear
- <u>UserID</u>
- UserInfo
- <u>UserName</u>
- <u>UserNumber</u>
- UserPassword
- VarAllocate
- VarCopy
- VarFree
- VarInfo
- VarInstance
- VarName
- VldClose
- VldDelete
- VldDirOpen
- VldDirRead
- VldOpen
- VldUpdate
- while
- WinAdd
- WinAddByName
- WinBarcodeSaveImage
- WinBeep
- WinBoxScrollVisible
- WinClose
- WinColorOpacityGet
- WinColorOpacitySet
- WinCopy

- WinCreate
- WinCroNavigate
- WinCroPrint
- WinCroReload
- WinCroSelection
- WinDestroy
- WinDialog
- WinDialogBox
- WinDialogResult
- WinDialogRun
- WinDocLoadBin
- WinDocLoadName
- WinDocSaveBin
- WinDocSaveName
- WinDocUserDictAddName
- WinDocUserDictRemoveName
- WinEmfProcess
- WinEvtProcessGet
- WinEvtProcessSet
- WinEvtProcNameGet
- WinEvtProcNameSet
- WinFlash
- WinFocusGet
- WinFocusSet
- WinGanttBoxAdd
- WinGanttCellInfo
- WinGanttIvlAdd
- WinGanttIvlRemove
- WinGanttLineAdd
- WinHalt
- WinIconPreload
- WinInfo
- WinLayer
- WinLstCellGet
- WinLstCellSet
- WinLstDatLineAdd
- WinLstDatLineInfo
- WinLstDatLineRemove
- WinLstEdit
- WinMenuContext
- WinMenuItemAdd
- WinMenuItemRemove
- WinMsdDelete
- WinMsdDeleteName
- WinMsdInsert
- WinMsdInsertName
- WinMsdRead
- WinMsdReadName
- WinMsdUpdate
- WinOpen
- WinPropGet

- WinPropSet
- WinRemove
- WinRtfLoad
- WinRtfLoadBin
- WinRtfLoadName
- WinRtfPicInsertMem
- WinRtfPicInsertName
- WinRtfSave
- WinRtfSaveBin
- WinRtfSaveName
- WinRtfSearch
- WinRtfTabGet
- WinRtfTabSet
- WinRvwColumn
- WinRvwEdit
- WinRvwUpdate
- WinSave
- WinSearch
- WinSearchClear
- WinSearchPath
- WinSearchPathGet
- WinShutdownBlock
- WinSleep
- WinThemeClose
- WinThemeDelete
- WinThemeOpen
- WinThemeSetDelete
- WinThemeSetOpen
- WinTreeNodeAdd
- WinTreeNodeRemove
- WinTreeNodeSearch
- WinUpdate
- WinUrmDialog
- WinUserEvent
- with
- WseArq
- WseInfo
- WseReturn
- WseStatus
- WseStrCnv
- XmlClose
- XmlError
- XmlGetValueAlpha
- XmlGetValueInt
- XmlLoad
- XmlOpenReader
- XmlOpenWriter
- XmlRead
- XmlSave

- XmlWrite
- <u>xor</u>

Deskriptoren

Verwendung von Deskriptoren

Alle Objekte werden in CONZEPT 16 über einen Deskriptor identifiziert. Die Deskriptoren der Objekte werden durch die entsprechenden Anweisungen zurückgegeben. Die Deskriptoren der untergeordneten Objekte können über die Anweisung <u>WinInfo()</u> bzw. <u>PrtInfo()</u> ermittelt werden.

Alle Deskriptoren mit Ausnahme von <u>Storage</u>- und <u>Binären Objekten</u> verfügen über folgende Eigenschaften:

- Name
- ID
- Custom

Die Eigenschaften können vom Programmierer verwendet werden. Insbesondere können die entsprechenden Deskriptoren in <u>dynamischen Strukturen</u> organisiert werden.

Bei allen Prozedurbefehlen die einen Deskriptor als erstes Argument erwarten, kann der Deskriptor auch vor den Befehl geschrieben und mit dem Operator -> an den Befehl übergeben werden.

Deskriptoren für Oberflächen- und Druckobjekte können über den Namen des Objektes ermittelt werden. Dazu wird dem Objektnamen das \$-Zeichen vorangestellt.

Zur Ermittlung des Deskriptors zu dem Namen wird der komplette Objekt-Baum verwendet. Die Suche wird dabei von einem bestimmten Objekt innerhalb des Suchbaums gestartet. Das Objekt ist entweder das zuletzt geladenen Fenster-Objekt oder das Objekt, das das Ereignis ausgelöst hat. Zunächst wird das Start-Objekt überprüft und anschließend alle untergeordneten Objekte. Konnte das Objekt nicht gefunden werden, wird das Fenster-Objekt und dessen untergeordnete Objekte durchsucht. Die Suche wird beendet, sobald das erste Objekt mit dem entsprechenden Namen gefunden wird.

Das Startobjekt bestimmt, wie lange die Suche dauert und welche Objekte durchsucht werden. Der Suchpfad wird automatisch gesetzt, wenn ein Ereignis ausgelöst oder ein <u>Frame</u>-Objekt geladen wird. Alle Namensreferenzen werden dann innerhalb des Fenster-Objekts aufgelöst.

Mit dem Befehl <u>WinSearchPath()</u> wird das Startobjekt der Suche definiert. Wurde innerhalb eines Ereignisses ein Objekt mit dem Befehl <u>WinOpen()</u> mehrfach geladen, ist der Suchpfad zunächst auf das zuletzt geladene Fenster gesetzt. Da die Namen der Objekte nicht mehr eindeutig sind, muss, um ein Objekt aus dem zuerst geladenen Fenster anzusprechen, der Suchpfad auf dieses Fenster gesetzt werden.



Die Suche wird bei jeder Namensreferenz erneut durchgeführt. Wird ein Deskriptor innerhalb einer Funktion mehrfach verwendet (zum Beispiel in einer Schleife), sollte der Deskriptor in einer Variablen gespeichert und diese als Objektreferenz genutzt werden.

Beispiele:

// Eingabeobjekt "edName" inkativ setzen\$edName->wpDisabled # true;// Selektion ausführentHdlSel

Deskriptoren von Fenster- bzw. Druckobjekten lassen sich ebenfalls über folgende Befehle ermitteln:

- WinInfo()
- WinSearch()
- PrtInfo()
- PrtSearch()

In jedem <u>Ereignis</u> kann der Deskriptor des ereignisauslösenden Objektes über das Argument aEvt ermittelt werden.

Mit dem Befehl <u>HdlInfo()</u> können über einen beliebigen Deskriptor Informationen eingeholt werden. Über <u>HdlLink()</u> lassen sich Deskriptoren miteinander verknüpfen. Durch <u>HdlEnum()</u> lassen sich alle Deskriptoren enumerieren.

Oberflächen-Objekte

Ein Dialog besteht aus ein oder mehrerer Objekte.

Alphabetische Liste

Siehe aller

Oberflächen-Objekte

Die Objekte untergliedern sich in folgende Bereiche:

- Form-Objekte
- Eingabe-Objekte
- Schaltflächen-Objekte
- Ansicht-Objekte
- Ausgabe-Objekte
- Toolbar-Objekte
- Anordnung-Objekte
- COM-Objekte
- Weitere Objekte

Application

Application

Siehe App, Liste, Eigenschaften

Die Eigenschaften des Application-Objekts wirken sich auf die Oberfläche der gesamten Applikation aus. Das Objekt wird vom System generiert. Es wird nicht in der Entwicklungsumgebung angelegt. Über den Befehl <u>WinInfo()</u> mit dem Parameter <u>WinApplication</u> oder das Schlüsselwort <u>App</u> wird das Application-Objekt ermittelt.

Über die Eigenschaften dieses Objekts können unter anderem die installierten Systemdrucker ermittelt und applikationsweite Eigenschaften gesetzt werden.

Die Einstellungen in diesem Objekt bleiben bis zum Ende der Applikation erhalten, können aber zur Laufzeit geändert werden. Werden die Einstellungen in einer Prozedur geändert, die aus dem Designer über den Menüpunkt "Datei / Testen" aufgerufen werden, bleiben sie bis zum Ende der Prozedur erhalten.

Das Application-Objekt steht nur Prozeduren und Funktionen zur Verfügung, die vom CONZEPT 16-Client (Standard oder Advanced) durchgeführt werden. Der Zugriff auf das Objekt zum Beispiel durch eine Prozedur, die durch die <u>CONZEPT</u> 16-Programmierschnittstelle oder den <u>SOA-Service</u> ausgeführt wird ist nicht möglich.

Form-Objekte
Siehe Alle
Oberflächen-Objekte
Frame
AppFrame
MdiFrame
TravFrame

TrayFrame
PrintDoc
PrintDocRecord
PrintForm

<u>PrintFormList</u>

Frame

Anwendungsfenster

Liste,

Siehe $\frac{\text{Eigenschaften}}{\text{Ereignisse}}$,

Dialogerstellung

Das Anwendungsfenster ist die Benutzeroberfläche für den Anwender.

Für die Erstellung einer MDI-Anwendung stehen die Fenstertypen Applikationsfenster und MDI-Anwendungsfenster zur Verfügung.

Damit bestehende MdiFrame-Objekte auch als Frame-Objekte benutzt werden können, besteht die Möglichkeit ein MdiFrame-Objekt in ein Frame-Objekt zu konvertieren. Hierzu existiert der Menüeintrag Konvertieren zu Frame im Kontextmenü des MdiFrame-Objekts, sowie der Menüeintrag Bearbeiten / Konvertieren zu Frame.

Der Titel des Anwendungsfensters kann mit der Eigenschaft Caption gesetzt werden.

AppFrame

Applikationsfenster

Liste,

Siehe Eigenschaften,

Ereignisse

Das Applikationsfenster ist das Rahmenfenster einer MDI (multiple document interface)-Anwendung. Es kann nur Fenster vom Typ <u>MdiFrame</u> aufnehmen.

Ein Applikationsfenster besitzt im Arbeitsbereich keine Interaktionselemente. Ein Applikationsfenster sollte immer ein Menü besitzen. Wird ein neues Applikationsfenster erstellt und der Menüeditor gestartet, wird ein Standardmenü vorgeschlagen, welches bereits Menüeinträge für die Darstellung und die Organisation der untergeordneten MdiFrame-Objekte enthält.

Der Titel des Anwendungsfensters kann mit der Eigenschaft Caption gesetzt werden.

Wird die Eigenschaft <u>StyleTheme</u> auf <u>WinStyleThemeModern</u> gesetzt, sollte dem AppFrame ein <u>ToolbarMenu</u>-Objekt hinzugefügt werden. Sonst werden die Minimieren- und Schließen-Schaltflächen der MDI-Fenster nicht angezeigt.

MdiFrame

MDI-Anwendungsfenster

Liste.

Eigenschaften,

Ereignisse,

Siehe $\frac{\text{Dialogerstellung}}{\text{Ereignisabläufe}}$

MdiFrame,

Datensatzpuffer

(Blog)

Ein Applikationsfenster kann ein oder mehrere untergeordnete MdiFrame-Objekte aufnehmen und darstellen (siehe WinAdd() und WinAddByName()). Zwar werden MdiFrame-Objekte im Designer unabhängig vom AppFrame-Objekt entworfen, brauchen jedoch zur Laufzeit immer ein AppFrame-Objekt.

Jedes Unterfenster kann innerhalb der Anwendung mehrfach mit unterschiedlichen oder auch identischen Daten erscheinen. Das MDI-Fenster kann den Arbeitsbereich des Applikationsfensters belegen und ist gleichzeitig durch dessen Größe begrenzt.

Wird die maximale Größe des Unterfensters gewählt, wird der gesamte Arbeitsbereich des Applikationsfensters genutzt. Als Name der Anwendung erscheint der Titel des Applikationsfensters und Titel des jeweiligen Unterfensters getrennt durch einen Bindestrich in der Titelzeile des Applikationsfensters.

MdiFrame-Objekte können alle Objekte der Objektpalette aufnehmen. Jedoch kann ihnen kein Menü zugewiesen werden. Lediglich das Applikationsfenster besitzt ein Menü. Das Menü muss ebenfalls die entsprechenden Steuerungselemente für das MDI-Fenster enthalten.

Zum Testen der MdiFrame-Objekte existiert wie gewohnt der Testmodus. Da die Objekte immer innerhalb eines Applikationsfensters laufen, wird für den Testmodus ein Standard-Applikationsfenster erzeugt.

Damit bestehende Frame-Objekte auch als MdiFrame-Objekte benutzt werden können, besteht die Möglichkeit ein Frame-Objekt in ein MdiFrame-Objekt zu konvertieren. Hierzu existiert der Menüeintrag Konvertieren zu MdiFrame im Kontextmenü des Frame-Objekts, sowie der Menüeintrag Bearbeiten / Konvertieren zu MdiFrame.

Wird der Menüeintrag ausgewählt, wird ein neues MdiFrame-Objekt erzeugt und die Objekte des Frame-Objekts übertragen. Hier bleibt zu beachten, dass ein evtl. vorhandenes Menü nicht mit übertragen werden kann, da MdiFrame-Objekte kein Menü besitzen können.

Der Titel des Anwendungsfensters kann mit der Eigenschaft <u>Caption</u> gesetzt werden.

Da in einer MDI-Anwendung ein Fenster mehrfach dargestellt werden kann (z. B. Dialog "Kunden" wurde mehrmals mit unterschiedlichen Daten geladen) muss für jeden Dialog ein eigener Datensatzpuffer angelegt werden. Über die Eigenschaft <u>DbRecBuf</u> kann einem MdiFrame ein Datensatzpuffer zugeordnet werden. Die Verwaltung des Puffers geschieht automatisch.

Wird ein MdiFrame-Objekt mehrfach geladen, kann es bei der Referenzierung auf Objekte innerhalb des Fensters über den Namen des Objekts zu Schwierigkeiten kommen, da der Name nicht mehr eindeutig ist. Die Namensauflösung funktioniert korrekt, solange innerhalb der Ereignisse des MDI-Fensters auf Objekte des gleichen Fensters referenziert wird. Sollen Objekte des anderen MDI-Fensters angesprochen werden, muss der Suchpfad entsprechend gesetzt werden (siehe <u>WinSearchPath()</u>).

Die Auflösung von Objekt-Referenzen (\$Name) in Prozeduren kann auf das aktive MDI-Fenster beschränkt werden. Dazu muss die Ausprägung <u>WinAppSearchMdiFrame</u> der <u>Flags</u>-Eigenschaft des <u>App</u>-Objektes gesetzt werden. Die Laufzeiten für die Referenzauflösung werden damit verkürzt, besonders bei Referenzen, die nicht aufgelöst werden können.

Wird die Eigenschaft <u>StyleTheme</u> auf <u>WinStyleThemeModern</u> gesetzt, sollte dem AppFrame ein <u>ToolbarMenu</u>-Objekt hinzugefügt werden. Sonst werden die Minimieren- und Schließen-Schaltflächen der MDI-Fenster nicht angezeigt.

TrayFrame

Tray-Anwendungsfenster

Liste,

Siehe $\frac{Eigenschaften}{Ereignisse}$,

Dialogerstellung

Das TrayFrame-Objekt dient der Erstellung von Tray-Anwendungen. Solche Anwendungen werden im SysTray-Bereich der Windows-Taskbar dargestellt. Folgende Komponenten sind für eine TrayFrame-Anwendung mindestens notwendig:

• Icon

Über die Eigenschaft muss ein Icon zugewiesen werden, welches nach dem Starten der TrayFrame-Anwendung im SysTray-Bereich dargestellt wird.

• MenuNameCntxt

Über die Eigenschaft MenuNameCntxt muss ein Menü zugewiesen werden. Das Menü wird angezeigt, wenn der Benutzer mit der rechten Maustaste auf das Icon klickt. Ein Doppelklick auf das Icon führt den fett hervorgehobenen Menüeintrag aus, der durch die Eigenschaft <u>Default</u> = <u>true</u> des <u>MenuItems</u> konfiguriert ist.

Die Tray-Anwendung wird mit der Anweisung WinDialog() oder WinDialogRun() gestartet.

PrintDoc



PrintDoc

Liste,

Siehe $\frac{Eigenschaften}{PrtInfo()}$,

Blog

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

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

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

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

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

PrintDocRecord

PrintDocRecord

Liste,

Siehe Eigenschaften,

Blog

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

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

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

Die angegebene Anzahl von Spalten werden beim Erzeugen gleichmäßig über die gesamte Seitenbreite angeordnet. Die Breite der Spalten kann entweder mit der Maus oder über die Eigenschaft Width verändert werden. Wird beim Ändern der Breite einer Spalte mit der Maus die -Taste gedrückt, werden alle übereinander liegenden Spalten mit verändert.

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

PrintForm



PrintForm

Liste,

Eigenschaften,

Siehe Beispiel,

PrintFormList,

Blog

Das PrintForm-Objekt dient dem dynamischen Aufbau von Druckdokumenten (PrintDoc-Objekt). Hierzu gibt es spezielle Prozedurbefehle, die die im PrintForm-Objekt enthaltenen Druckobjekte einem PrintDoc-Objekt hinzufügen.

Das PrintForm-Objekt wird dann verwendet, wenn ein Druckjob aus kleineren oder sehr unterschiedlichen Elementen aufgebaut werden soll. Zum Beispiel eine Rechnung oder eine Liste, die auf Daten aus mehreren Tabellen zugreift.

Wird ein PrintForm-Objekt im Designer erstellt, ist in der Objektpalette nur der Registerreiter "Druck" vorhanden, und nur die dort enthaltenen Objekte können dem Objekt hinzugefügt werden.

PrintFormList



Liste,

 $Siehe \frac{Eigenschaften}{PrintForm},$

Blog

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

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

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

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

Eingabe-Objekte Objekte der Eingabe-Palette Siehe <u>Alle</u> <u>Oberflächen-Objekte</u>

Edit • I

<u>IntEdit</u>

BigIntEdit 1227

<u>FloatEdit</u>

<u>DecimalEdit</u>

TimeEdit 236

<u>DateEdit</u>

<u>ColorEdit</u>

FontNameEdit A.S.

FontSizeEdit ALD

<u>TextEdit</u>

RtfEdit

<u>CodeEdit</u>

Edit

Eingabeobjekt für einzeilige Zeichenketten

Liste,

 $Siehe \frac{Eigenschaften}{Ereignisse},$

<u>alpha</u>

Der Inhalt des Eingabeobjekts kann mit der Eigenschaft Caption gesetzt und gelesen werden.

In der Eigenschaft <u>DbFieldName</u> kann ein Datenbankfeld von beliebigem Typ mit dem Eingabeobjekt verknüpft werden. Ist in den Einstellungen die Option "Setzen von DbFieldName setzt auch Eingabelimit" aktiviert, wird die Eigenschaft LengthMax auf die Länge des Datenbankfeldes gesetzt, wenn zuvor in der Eigenschaft kein kleinerer Wert eingetragen war.

Für eine mehrzeilige Eingabe von Zeichenketten stehen die Objekte TextEdit und RtfEdit zur Verfügung.

IntEdit

Eingabeobjekt für ganzzahlige Werte

Liste,

Siehe Eigenschaften,

Ereignisse, int

Der Inhalt des Eingabeobjekts kann mit der Eigenschaft <u>CaptionInt</u> gesetzt und gelesen werden. Die zulässigen Werte liegen im Bereich von -2.147.483.647 und +2.147.483.647.

In der Eigenschaft <u>DbFieldName</u> kann ein Datenbankfeld mit dem Eingabeobjekt verknüpft werden. Einem IntEdit-Objekt kann auch ein Datenbankfeld vom Typ Ganzzahlig kurz (<u>word</u>) zugewiesen werden. Ist in den <u>Einstellungen</u> des Designers die Option "Setzen von DbFieldName setzt auch Eingabelimit" aktiviert, werden die Eigenschaften <u>MinInt</u> und <u>MaxInt</u> auf die Beschränkungen des Datenbankfeldes gesetzt, wenn zuvor in der Eigenschaft keine engeren Grenzen eingetragen waren.

Die Eigenschaft InputCtrl bestimmt, ob die Eingabehilfe aktiv ist.

BigIntEdit

Eingabeobjekt für große ganze Zahlen

Liste,

 $Siehe \frac{Eigenschaften}{Ereignisse},$

bigint

Der Inhalt des Eingabeobjekts kann mit der Eigenschaft CaptionBigInt gesetzt und gelesen werden. Die zulässigen Werte liegen im Bereich von -9.223.372.036.854.775.807 und +9.223.372.036.854.775.807.

In der Eigenschaft **DbFieldName** kann ein Datenbankfeld mit dem Eingabeobjekt verknüpft werden. Einem BigIntEdit-Objekt können auch Datenbankfelder der Datentypen Ganzzahlig kurz (word) und Ganzzahlig lang (int) zugewiesen werden. Ist in den Einstellungen des Designers die Option "Setzen von DbFieldName setzt auch Eingabelimit" aktiviert, werden die Eigenschaften MinBigInt und MaxBigInt auf die Beschränkungen des Datenbankfeldes gesetzt, wenn zuvor in der Eigenschaft keine engeren Grenzen eingetragen waren.

FloatEdit

Eingabeobjekt für Gleitkommazahlen

Liste,

 $Siehe \frac{\underline{Eigenschaften},}{\underline{Ereignisse},}$

float

In diesem Objekt können Werte vom Typ <u>float</u> eingegeben werden. Der Inhalt des Eingabeobjekts kann mit der Eigenschaft CaptionFloat gesetzt und gelesen werden.

In der Eigenschaft <u>DbFieldName</u> kann ein Datenbankfeld vom Typ <u>float</u> (Gleitkomma) mit dem Eingabeobjekt verknüpft werden.

Die Eigenschaft InputCtrl bestimmt, ob die Eingabehilfe aktiv ist.

DecimalEdit

Eingabeobjekt für Dezimalzahlen

Liste,

Siehe $\frac{Eigenschaften}{Ereignisse}$,

decimal

Der Inhalt des Eingabeobjekts kann mit der Eigenschaft CaptionDecimal gesetzt und gelesen werden.

In der Eigenschaft <u>DbFieldName</u> kann ein Datenbankfeld vom Typ <u>decimal</u> (dezimal) mit dem Eingabeobjekt verknüpft werden.

Wird die Eigenschaft FmtDecimalFlags in der Ausprägung FmtNumNoZero gesetzt, können die Inhalte "0" und "undefiniert" nicht mehr unterschieden werden. In beiden Fällen wird das Eingabeobjekt leer dargestellt.

Verliert das Eingabeobjekt den Fokus und ist das Eingabeobjekt leer, wird je nach Ausprägung der Eigenschaft DecEditFlags der Wert des Eingabefeldes auf "undefiniert" (<u>WinDecEditDecimalUndef</u> ist gesetzt) oder "0" (<u>WinDecEditDecimalUndef</u> ist nicht gesetzt) gesetzt.

TimeEdit

Eingabeobjekt für Zeitwerte

Liste,

Siehe $\frac{\text{Eigenschaften}}{\text{Ereignisse}}$,

Der Inhalt des Eingabeobjekts kann mit der Eigenschaft CaptionTime gesetzt und gelesen werden.

In der Eigenschaft <u>DbFieldName</u> kann ein Datenbankfeld vom Typ <u>time</u> (Zeit) mit dem Eingabeobjekt verknüpft werden.

Die Eigenschaft InputCtrl bestimmt, ob die Eingabehilfe aktiv ist.

Der Nullwert für TimeEdit ist 24:00:00.

DateEdit

Eingabeobjekt für Datumswerte

Liste,

 $Siehe \frac{\underline{Eigenschaften}}{\underline{Ereignisse}},$

date

Der Inhalt des Eingabeobjekts kann mit der Eigenschaft CaptionDate gesetzt und gelesen werden. In der Eigenschaft DbFieldName kann ein Datenbankfeld vom Typ date (Datum) mit dem Eingabeobjekt verknüpft werden.

Die Eigenschaft InputCtrl bestimmt, ob die Eingabehilfe aktiv ist.

Standardmäßig wird bei einer zweistelligen Eingabe des Jahres 1900 zu dem Jahreswert addiert. Die Eingabe 01.01.01 wird demnach als 01.01.1901 interpretiert. Über die Eigenschaft DateWindow lässt sich steuern, ob 1900 oder 2000 zu dem Jahreswert addiert werden soll.

Das eingegebene Datum wird auf Gültigkeit überprüft. Über die Eigenschaft InputCheck wird festgelegt, ob die Überprüfung bereits während der Eingabe oder nach Verlassen des Objektes stattfindet.

Eine Schnelleingabe des Datums kann durch das Setzen der Eigenschaft FormatDate auf den Wert WinFmtDateDDMM erreicht werden. In diesem Fall reicht bereits die Angabe des Tages in zweistelliger Schreibweise. Monat und Jahr werden aus der Eigenschaft DefaultDate ergänzt.

Soll ein bestehender Wert in dem DateEdit-Objekt nur teilweise überschrieben werden, kann in der Eigenschaft Flags die Ausprägung WinAppEditDateFast angegeben werden. Bekommt das Objekt den Eingabefokus, wird nicht der gesamte Inhalt des Objektes selektiert und er kann Zeichenweis überschrieben werden.

Der Nullwert für DateEdit ist 0.0.0.

Durch die Eigenschaft StyleTheme wird beeinflusst, ob die Kalenderanzeige des Objekts in Theme-Darstellung WinStyleThemeSystem oder in den angegebenen Farben WinStyleThemeNone gezeichnet wird. Die Eigenschaft wirkt sich bei dem Objekt nur aus, wenn die gleiche Eigenschaft beim App-Objekt auf <u>WinStyleThemeSystem</u> gesetzt ist.

Das Datum kann mit Hilfe der folgenden Tastaturkommandos relativ zum eingetragenen Datum verändert werden:

Tastenkombination Veränderung relativ zum aktuellen Datum

Gleicher Wochentag der vorherigen Woche 0 Gleicher Wochentag der nächsten Woche

Vorheriger Tag Nächster Tag

bilda Gleicher Tag des vorherigen Monats

bild	Gleicher Tag des nächsten Monats
strg + bilda	Gleiches Datum des vorhergehenden Jahres
strg + bildv	Gleiches Datum des nächsten Jahres
strg + pos1	1. Tag des Monats des aktuellen Datums
strg] + endel	Letzter Tag des Monats des aktuellen Datums

Ist das Datum der 29.2. eines Schaltjahres und wird und bilde oder gedrückt, wird auf den 28.2. des vorhergehenden bzw. nachfolgenden Jahres positioniert.

Die Tastaturkombinationen und können nur verwendet werden, wenn die Eigenschaft EmulateKeys des übergeordneten Frame-Objekts auf <u>false</u> gesetzt ist.

ColorEdit

Eingabeobjekt für Farbwerte

Liste,

Siehe Eigenschaften,

Ereignisse

Der Inhalt des Eingabeobjekts kann in der Eigenschaft <u>CaptionColor</u> gesetzt und gelesen werden.

In diesem Objekt können Werte der Datentypen <u>word</u> (ganzzahlig kurz) und <u>int</u> (ganzzahlig lang) dargestellt werden.

Über die Eigenschaft <u>ModeColor</u> kann eingestellt werden, ob nur Standardfarben oder Standard- / System- und benutzerdefnierte Farben auswählbar sind.

Die Eigenschaft InputCtrl bestimmt, ob die Eingabehilfe aktiv ist.

FontNameEdit

Mai Objekt zur Auswahl der Schriftart

Liste,

Siehe Eigenschaften,

Ereignisse

Über dieses Objekt kann eine Auswahl der auf dem System installierten Zeichensätze getroffen werden. Der ausgewählte Zeichensatz kann in der Eigenschaft <u>Caption</u> ausgelesen werden.

Mit der gleichen Eigenschaft kann ebenfalls eine Vorbelegung gesetzt werden.

Für die Größe des Zeichensatzes kann das Objekt <u>FontSizeEdit</u> verwendet werden. Die Verbindung zwischen dem FontNameEdit-Objekt und dem <u>FontSizeEdit</u>-Objekt wird durch die Eigenschaft <u>ObjLink</u> des FontNameEdit-Objekts hergestellt. Die Eigenschaft wird automatisch beim Einfügen der Objekte eingetragen.

Das Objekt FontNameEdit ist auch Bestandteil der Werkzeugleiste vom Typ ToolbarRtf.

FontSizeEdit

MISObjekt zur Auswahl des Schriftgrades

Liste,

Siehe Eigenschaften,

Ereignisse

Mit diesem Objekt kann der Schriftgrad einer Schriftart ausgewählt werden. Die Schriftart wird in einem <u>FontNameEdit</u>-Objekt ausgewählt, das mit diesem Objekt über die Eigenschaft <u>ObjLink</u> verbunden ist. Die Auswahl der Schriftart setzt automatisch die Eigenschaft <u>FontName</u> des Objekts. Das FontSizeEdit-Objekt stellt nur die Schriftgrößen zur Verfügung, die für diese Schriftart definiert sind.

Das Objekt FontSizeEdit ist auch Bestandteil der Werkzeugleister vom Typ ToolbarRtf.

Der Schriftgrad kann in der Eigenschaft <u>CaptionInt</u> gesetzt und gelesen werden.

TextEdit

Eingabeobjekt für mehrzeilige Zeichenketten

 $Siehe \frac{Eigenschaften}{Ereignisse},$

RtfEdit

Der Inhalt des TextEdit-Objekts kann über die Eigenschaften Caption, DbFieldName oder <u>DbTextBuf</u> gesetzt werden. Bei Verwendung eines Textpuffers (<u>DbTextBuf</u>) bleibt das Setzen der Eigenschaften Caption und DbFieldName ohne Wirkung und das Auslesen liefert einen leeren alpha-String.

In der Eigenschaft **DbFieldName** kann ein Datenbankfeld mit beliebigem Typ angegeben werden.

Befinden sich innerhalb des internen Textes Linienzeichen, kann zur korrekten Darstellung des Textes der Zeichensatz "C16 Courier" oder "C16 Terminal" in der Eigenschaft Font eingetragen werden.

Die Länge des Textes ist nicht <u>limitiert</u>.

```
RtfEdit
```

Eingabeobjekt für mehrzeiligen formatierten Text

Liste.

Eigenschaften,

Ereignisse,

Siehe $\frac{\text{TextEdit}}{\text{Befehle}}$, $\frac{\text{Text}}{\text{Text}}$

und Daten

mischen,

PrtRtf

In diesem Objekt kann unformatierter Text und Text im Rich Text Format (RTF) angezeigt und editiert werden. Der Text kann aus unterschiedlichen Ouellen stammen. Die Quelle wird in der Eigenschaft StreamSource angegeben.

In den Eigenschaften des Objekts kann eine generelle Formatierung in Bezug auf die Schriftart (FontName und RtfEffect), Schriftgröße (FontSize und FontSizeTwips) und der Textausrichtung (RtfAlign) vorgenommen werden.

Soll innerhalb des Textes die Formatierung verändert werden, wird eine Toolbar vom Typ ToolbarRtf benötigt. Wird die Toolbar eingefügt, verbindet sie sich automatisch mit einem bereits existierenden RtfEdit-Objekt. Genauso verbindet sich das RtfEdit-Objekt mit einem bereits bestehenden ToobarRtf-Objekt. Bestehen mehrere Möglichkeiten eine Verbindung herzustellen, wird immer eine Verbindung zum ersten Objekt hergestellt.

Die Verbindung wird über die Eigenschaft ObiLink hergestellt.

Die Länge des Textes wird durch seine Quelle limitiert. Wird der Text in der Eigenschaft Caption angegeben, können maximal 8192 Zeichen geschrieben bzw. dargestellt werden.

Wurde eine andere Ouelle angegeben, bezieht sich der Inhalt der Eigenschaft Caption auf den selektierten Bereich. Eine Selektion innerhalb des Textes kann über die Eigenschaft Range angegeben werden. Hiermit kann ebenfalls die Einfügemarke innerhalb des Textes positioniert werden, wenn der minimale und maximale Wert der Eigenschaft gleich ist. Wird die Eigenschaft Range auf den Wert RangeMake(-1, -1) gesetzt, wird das Ende des Textes angesprungen.

Ein markierter Text kann mit Hilfe von Formatierungsanweisungen formatiert werden.

Im RtfEdit-Objekt werden folgende Tastaturkommandos ausgewertet:

Navigation

- ♦ Der Cursor kann mit den Pfeiltasten im Text bewegt werden.
- ◆ strg + → und strg + → positioniert den Cursor an den Anfang des folgenden oder des vorigen Wortes.
- strg + bidv und strg + bida scrollt den Text seitenweise nach unten und oben.
- post und positioniert innerhalb einer Zeile auf die erste Spalte, oder auf das letzte Zeichen.

- und blättert eine Bildschirmseite.

 | Strg | + | Post | und | und | strg | + | ende | positioniert auf den Anfang bzw. auf das Ende | des gesamten Textes.
- strg] + bilda und strg] + bildv positioniert den Cursor an den Anfang bzw. an das Ende des angezeigten Textes.

Markieren

- ♦ Zum Markieren können die Tastenkombinationen zur Navigation bei gedrückter _____-Taste verwendet werden.
- strg] + A markiert den gesamten Text.
- + entf löscht den markierten Text oder von der Cursorposition bis zum Ende des Wortes.
- zum Anfang des Wortes.

• Ausschneiden / Kopieren / Einfügen

- strg + oder + entf schneidet den markierten Textbereich aus.
- strg + C oder strg + einfg kopiert den markierten Textbereich.
- strg + Oder + einfg fügt den ausgeschnittenen/kopierten Textbereich ein.
- strg + macht die letzte Änderung rückgängig.
- stellt die letzt rückgängig gemachte Änderung wieder her.

• Formatierung und Eingabe von Sonderzeichen

- + Absatz zentriert.
- strg] + 🗓 Darstellung im Blocksatz.
- Absatz rechtsbündig.
- Absatz linksbündig.
- tie<u>f</u> gestellt.
-]+∰ hoch gestellt.
- einfacher Zeilenabstand.
- doppelter Zeilenabstand.
- ha<u>lbe</u>r Zeilenabstand.
- strg] + 📤 + 🖺 Wechsel des Aufzählungszeichen.
- strg + A wandelt den markierten Bereich in Großbuchstaben oder Kleinbuchstaben.
- Umwandlung des links neben der Einfügemarke stehenden Hexadezimalcodes in das entsprechende Unicode-Zeichen.
- ◆ at + Umwandlung des links neben der Einfügemarke stehenden Unicode-Zeichens in den entsprechenden Hexadezimalcode.

Beispiele:

tRange:min # 1; tRange:max # 30;// alternativtRange # RangeMake(1, 30); \$RtfEdit->wpRange # tRange

Text und Daten mischen

Texte mit Datensätzen verbinden

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

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

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

• Beim Lesen eines Textes in ein RtfEdit-Objekt

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

• Beim Speichern eines Textes aus einem RtfEdit-Objekt

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

• Beim Lesen eines Textes in ein CtxDocEdit-Objekt

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

• Beim Speichern eines Textes aus einem CtxDocEdit-Objekt

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

• Beim Drucken eines Textes mit dem PrtRtf-Objekt

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

Folgende Markierungen werden beim Mischen von Text und Daten ersetzt:

• ~F:<Feldname>~

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

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

		Über diese Option wird die Anzahl der Nachkommastellen definiert. Soll die Anzahl der Nachkommastellen definiert werden, muss entweder zuvor mindestens einer der bereits genannten Formatoptionen oder ein "/" ohne Optionen angegeben werden. Werden keine Nachkommastellen angegeben, werden alle Nachkommastellen angezeigt.
Datum (<u>date</u>)	'/L'	Mit dieser Option wird das lange Datumsformat eingestellt. Wird kein langes Datumsformat gewählt, erscheint das Datum in der kurzen Anzeige.
Zeit (<u>time</u>)	'/S'	Die Uhrzeit wird mit Sekunden angezeigt.
	'/H'	Die Uhrzeit wird mit Sekunden und hundertstel Sekunden angezeigt.
Logisch (<u>logic</u>)	'/ <true>/<false>'</false></true>	Logische Felder und Variablen müssen mit Werten versehen werden, die in Abhängigkeit des Wertes angezeigt werden sollen. Die erste Zeichenkette wird angezeigt, wenn der Wert true ist, die zweite Zeichenkette wird angezeigt, wenn der Wert false ist.

• ~G:<Variablenbereich>:<Variablenname>~

Für Variablen gelten die gleichen Formatierungsmöglichkeiten, wie für Felder.

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

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

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

Beispiele:

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



🔟 Editor für Prozeduren und Texte

Siehe Eigenschaften.

Ereignisse,

Befehle

Das CodeEdit-Objekt dient der Eingabe von Prozeduren und Texten und ist Basis für den Editor. Der Inhalt wird über die Eigenschaft FileName definiert. Hierbei können sowohl interne, als auch externe Dokumente angezeigt werden. Das Objekt unterstützt Syntax-Highlighting für CONZEPT 16, XML- und ISON-Dokumente. Der Typ des angezeigten Textes wird über die Eigenschaft EditorTextType definiert. Ist die Eigenschaft EditorFold aktiv, kann der Inhalt des CodeEdit-Objektes zwischen zusammengehörenden Klammern zusammengeklappt werden.



Der Zugriff auf Eigenschaften und Funktionen des CodeEdit-Objektes ist erst ab dem EvtCreated des enthaltenden Elternfensters möglich.

Im CodeEdit-Objekt können folgende Tastenkombinationen verwendet werden:

Navigation

- Der Cursor kann mit den Pfeiltasten im Text bewegt werden.
- strg + und strg + positioniert den Cursor an den Anfang des folgenden oder des vorigen Wortes.
- positioniert innerhalb einer Zeile auf das erste oder letzte Zeichen, das kein Leerzeichen ist. Beim erneuten Drücken der gleichen Taste wird auf das erste oder letzte Zeichen der Zeile positioniert.
- bildv und blättert eine Bildschirmseite.
- strg + post und strg + ende positioniert auf den Anfang bzw. auf das Ende des gesamten Textes.
- strg + , strg + ida und strg + bilde scrollt den Inhalt des Editors zeilen- bzw. spaltenweise, ohne den Cursor zu verschieben.
- F2 versieht die aktuelle Zeile mit einem Lesezeichen zu dem zu einem späteren Zeitpunkt mit strg + F2 bzw. + F2 gesprungen werden kann.
- latt + post und latt + lender wechselt zwischen der öffnenden und schließenden Klammer, wenn der Cursor auf einer Klammer steht (unabhängig davon, welche Klammern verwendet werden).
- ett + 1 ... ett + 4 wechselt zwischen den Bereichen, wenn ein Text in verschiedene Bereiche aufgeteilt wurde.

Markieren

- Zum Markieren können die Tastenkombinationen zur Navigation bei gedrückter Taste verwendet werden.
- Ein Bereich kann durch Ziehen mit der Maus markiert werden. Soll ein rechteckiger Bereich (nicht zeilenweise) markiert werden, kann die Maus mit gedrückten + - bzw. strg + - Tasten verwendet werden.
- strg + A markiert den gesamten Text.
- at + a markiert einen rechteckigen Bereich.

- Mit oder kann eine rechteckige Markierung links oder rechts verschoben werden.
- strg] + att + post] oder strg] + att + endel markiert zwischen der öffnenden und schließenden Klammer, wenn der Cursor auf einer Klammer steht (unabhängig davon, welche Klammern verwendet werden).

Ausschneiden / Kopieren / Einfügen

- strg] + X oder + entf schneidet den markierten Textbereich aus.
- strg] + Oder strg] + einfg kopiert den markierten Textbereich.
- oder + einfg fügt den ausgeschnittenen/kopierten Textbereich ein.
- + löscht die aktuelle Zeile.

Folding

- + einzelnen Bereich zuklappen.
- strg + einzelnen Bereich aufklappen.
- + alle Bereiche zuklappen.
- alle Bereiche aufklappen.

Zoom

- strg + + oder strg + Ansicht vergrößern.
 strg + oder strg + Ansicht verkleinern.
- strg] + O Ansicht zurücksetzen.

Sonstiges

- strg] + entf löscht den Text bis zum Ende des Wortes.
- strg + at + blendet Leerzeichen ein/aus
- strg + + + wandelt den markierten Bereich in Großbuchstaben.
- strg + U wandelt den markierten Bereich in Kleinbuchstaben.
- setzt/löscht ein Lesezeichen.
- springt zum nächsten Lesezeichen.
- springt zum vorherigen Lesezeichen.
- strg + B aktiviert und deaktiviert die Anzeige der zusammengehörenden Klammern.
- strg + fügt das aktuelle Datum an der Cursorposition ein.
- strg + blendet die Anzeige der Zeilennummern ein bzw. aus.
- strg + kommentiert die markierten Zeilen aus oder entfernt die Kommentarzeichen. Ist mindestens eine Zeile nicht auskommentiert, werden alle Zeilen auskommentiert.

• strg + fügt unabhängig vom automatischen Klammerungsmodus einen Zeilenumbruch ein.

Schaltflächen-Objekte Siehe <u>Alle</u> <u>Oberflächen-Objekte</u>

Button

MenuButton ==

1 <u>CheckBox</u>

Button

OK

Button

Liste,

Siehe Eigenschaften,

Ereignisse

Durch das Betätigen einer Schaltfläche werden unterschiedliche Aktionen ausgelöst.

Folgende Aktionen sind möglich:

• Ereignis EvtClicked

Die hier angegebene Prozedur wird ausgeführt.

Über das Kontextmenü der Schaltfläche lassen sich Aktionen definieren, deren Ausführung keiner Prozedur bedürfen:

Schließen

Die Eigenschaft <u>TypeButton</u> wird auf den Wert <u>WinBtnClose</u> gesetzt. Die Eigenschaft <u>ID</u> erhält den Wert <u>WinIdClose</u>. Das Drücken der Schaltfläche führt zu einem Schließen des Fensters.

Abbrechen

Die Eigenschaft <u>TypeButton</u> wird auf den Wert <u>WinBtnClose</u> gesetzt. Die Eigenschaft <u>ID</u> erhält den Wert <u>WinIdCancel</u>. Das Drücken der Schaltfläche führt zu einem Schließen des Fensters.

• Ok

Die Eigenschaft <u>ID</u> erhält den Wert <u>WinIdOk</u>. Das Drücken der Schaltfläche führt zu einem Schließen des Fensters.

• Hilfe

Die Eigenschaft <u>TypeButton</u> wird auf den Wert <u>WinBtnHelp</u> gesetzt. Die Eigenschaft <u>ID</u> erhält den Wert <u>WinIdHelp</u>. Das Drücken der Schaltfläche ruft die in der Eigenschaft in <u>Help</u> angegebene Hilfe auf.

• RTF-Toolbar

Für die Verwendung des Buttons in einer <u>Toolbar</u> für das <u>RtfEdit-Objekt</u>, stehen verschiedene Funktionen zur Auswahl:

Fett

Kursiv

Unterstrichen

Linksbündig

Rechtsbündig

Zentriert

Blocksatz

Aufzählungszeichen

Textfarbe

Hintergrundfarbe

Das Auslösen einer Schaltfläche kann mit der Maus oder über die Tastatur erfolgen. Bei Verwendung der Tastatur wird über die Leertaste die Schaltfläche ausgelöst. Soll ein Drücken auch über die Return-Taste möglich sein, muss entweder die Eigenschaft EmulateKeys des Fensters auf false gesetzt sein, andernfalls findet ein Wechsel des Eingabefokus auf das nächste Objekt statt, oder die Eigenschaft ReturnKeyClick hat den Wert true. In diesem Fall kann der Button immer mit der Return-Taste aktiviert werden, sofern er den Fokus besitzt.

Eine Schaltfläche kann ebenfalls durch eine mnemonische Auswahl aktiviert werden. Dazu wird in der Eigenschaft <u>Caption</u> dem gewünschte Auswahlzeichen das &-Zeichen vorangestellt. Das Auswahlzeichen wird dann mit einem Unterstrich dargestellt und der Button kann über die Kombination + Auswahlzeichen ausgelöst werden.

Neben der Beschriftung kann ein Button auch über ein Symbol verfügen. In der Eigenschaft ImageTile kann aus einer Reihe von vordefinierten Symbolen gewählt werden. Das Verwenden eigener Symbole ist ebenfalls möglich. Dazu muss eine entsprechende Grafik in die Datenbank importiert werden. In der Eigenschaft ImageTileUser wird dann die Nummer der zu verwendenden Kachel angegeben.

Um eine Drag & Drop-Operation auf einem Button auszulösen, muss zusätzlich die strg -Taste gedrückt werden.

MenuButton

MenuButton

Liste,

Siehe Eigenschaften,

Ereignisse

Das Objekt MenuButton funktioniert identisch zum <u>Button</u>-Objekt. Lediglich beim Anklicken der Pfeilschaltfläche oder Drücken der Tastenkombination bzw. strg + wird ein Menü angezeigt. Das Menü wird in der Eigenschaft <u>MenuName</u> eingetragen.

Vor dem Laden des Menüs wird das Ereignis <u>EvtMenuPopup</u> aufgerufen. Hier kann das zu ladende Menü geändert werden. In dem anschließend aufgerufenen Ereignis <u>EvtMenuInitPopup</u> können noch Änderungen an den einzelnen Menüeinträgen vorgenommen werden, bevor das Menü angezeigt wird.

Wird ein Menüpunkt ausgewählt, wird das Ereignis <u>EvtMenuCommand</u> aufgerufen. Das Ereignis wird sowohl bei der Auswahl eines Menüeintrages aus dem Kontextmenü als auch bei der Auswahl eines Eintrages aus dem Popupmenü aufgerufen. Der Deskriptor des geladenen Menüs kann mit der Anweisung <u>WinInfo()</u> ermittelt werden.

ColorButton

Eingabeobjekt für Farbwerte

Liste,

Siehe <u>Eigenschaften</u>,

<u>Ereignisse</u>

Über dieses Objekt kann der Benutzer Farben auswählen. Die ausgewählte Farbe kann über die Eigenschaft <u>CaptionColor</u> gesetzt und gelesen werden.

Radiobutton Einfachauswahlknopf

Liste,

Siehe Eigenschaften,

Ereignisse

Auswahlknöpfe sind geeignet um den Benutzer zwischen verschiedenen festgelegten Eingaben wählen zu lassen. Von allen Auswahlknöpfen die dasselbe Vaterobjekt haben kann jeweils nur ein Knopf ausgewählt sein.

Der Status des Knopfes kann mit der Eigenschaft <u>CheckState</u> gelesen und gesetzt werden.

Wird ein Dialog aufgerufen, bekommt das erste Objekt in dem Dialog den Fokus. Im Falle eines Radiobuttons bedeutet das automatisch, dass der erste Radiobutton ausgewählt wird. Soll der Fokus auf den aktivierten Radiobutton gesetzt werden, muss die Ausprägung <u>WinAppRadioFocusChecked</u> der Eigenschaft <u>Flags</u> des <u>App</u>-Objekts gesetzt sein.

Checkbox

✓ Mehrfachauswahlknopf

Liste,

Siehe $\frac{\text{Eigenschaften}}{\text{Ereignisse}}$,

<u>logic</u>

Dieses Objekt kann dazu verwendet werden, Einstellungen von Parametern, die entweder ein oder aus sein müssen, anzuzeigen. Im Gegensatz zu dem Radiobutton handelt es sich um einen Mehrfachauswahlknopf.

In diesem Objekt kann nur der Datentyp logic dargestellt werden.

Der Status des Häkchens kann mit der Eigenschaft CheckState gelesen und gesetzt werden.

Objekte zur Datenansicht Siehe <u>Alle</u> <u>Oberflächen-Objekte</u>

<u>DataList</u>

<u>StoList</u>

RecList

<u>RecView</u>

TreeView [

DataList



DataList

Liste,

Eigenschaften,

Siehe $\frac{\text{Ereignisse}}{\text{Befehle}}$,

Fragen zur

DataList

Mit diesem Objekt können Informationen zeilenweise angezeigt werden. Das DataList-Objekt ähnelt im Aufbau dem RecList-Objekt. In dem DataList-Objekt werden allerdings keine Datensätze aus der Datenbank sondern Daten, die sich im Hauptspeicher befinden, dargestellt. Der Datentyp der darzustellenden Informationen wird über die Eigenschaft ClmTvpe spezifiziert. Die Eigenschaft kann zur Laufzeit nur gelesen werden. Der Typ der Spalte wird somit beim Erstellen des DataList-Objekts festgelegt. Über die Eigenschaften <u>TileNameUser</u> und <u>ClmTypeImage</u> können Grafiken in einer DataList angezeigt werden.

Die darzustellenden Informationen werden zur Laufzeit der Liste zugeordnet. Einzelne Zeilen werden mit dem Befehl WinLstDatLineAdd() der Liste hinzugefügt oder mit WinLstDatLineRemove() entfernt.

Die Informationen innerhalb einer Zeile werden mit WinLstCellSet() gesetzt und mit <u>WinLstCellGet()</u> gelesen.

Zum Bearbeiten der Liste im Designer steht der DataList-Editor zur Verfügung.

Bei diesem Objekt ist eine Mehrfachselektion möglich. Zur Verwendung der Mehrfachselektion, muss die Eigenschaft MultiSelect gesetzt werden. Weitere Informationen befinden sich im Abschnitt SelectionData-Objekt.

Column

Spalteneintrag

Liste,

Eigenschaften

Spalte /

RecList,

Eigenschaften

Spalte /

Siehe <u>RecView</u>,

Eigenschaften

Spalte /

DataList,

Eigenschaften

Spalte /

<u>StoList</u>

Die Objekte <u>RecList</u>, <u>RecView</u>, <u>DataList</u> und <u>StoList</u> bestehen aus einem oder mehreren Spalteneinträgen.

DataListPopup

DataListPopup

Liste,

Siehe Eigenschaften,

Ereignisse

Mit diesem Objekt wird das <u>DataList</u>-Objekt als Combobox dargestellt. In diesem Fall ist die DataList ein Unterobjekt des <u>PopupList</u>-Objekts.

Mit diesem Objekt können Informationen zeilenweise angezeigt werden. Das DataListPopup-Objekt ähnelt im Aufbau dem <u>RecListPopup</u>-Objekt. In dem DataListPopup-Objekt werden allerdings keine Datensätze aus der Datenbank sondern Daten, die sich im Hauptspeicher befinden, dargestellt. Der Datentyp der darzustellenden Informationen wird über die Eigenschaft <u>ClmType</u> spezifiziert. Die Eigenschaft kann zur Laufzeit nur gelesen werden. Der Typ der Spalte wird somit beim Erstellen des DataList-Objekts festgelegt. Über die Eigenschaften <u>TileNameUser</u> und <u>ClmTypeImage</u> können Grafiken in einem DataListPopup-Objekt angezeigt werden.

Eine DataListPopup wird standardmäßig mit maximal 10 Zeilen dargestellt. Über die Eigenschaft MaxLines kann eine Anzahl zwischen 1 und 100 Zeilen gesetzt werden.

Neben der Maus kann die Liste über die Tastenkombination strug + oder per Prozedur über die Eigenschaft PopupOpen aufgeklappt werden.

Für die Bearbeitung einer DataListPopup stehen entsprechende <u>Prozedurbefehle</u> zur Verfügung.

StoList

Liste mit Datenbankelementen

Liste,

Siehe $\frac{\text{Eigenschaften}}{\text{Ereignisse}}$,

Storage-Objekte

In diesem Objekt können alle vom Programmierer angelegten Ressourcen der Applikation angezeigt werden. Innerhalb einer Liste kann immer nur eine Art von Ressource angezeigt werden. Dies wird in der Eigenschaft StoView bestimmt.

Ausgehend von einem <u>Benutzer</u>, einer <u>Benutzergruppe</u> oder einer <u>Elementgruppe</u> können auch alle Mitgliedschaften bzw. alle zugeordneten Elemente in der Liste angezeigt werden. Dazu muss der Name des Benutzers, der Benutzergruppe oder der Elementgruppe in der Eigenschaft UrmParentName angegeben werden. Es werden dann die Benutzergruppen des Benutzers, die Mitglieder einer Benutzergruppe bzw. die Elemente einer Elementgruppe angezeigt. Zur Laufzeit kann auch der Deskriptor des entsprechenden Objekts in die Eigenschaft <u>UrmParentHandle</u> eingetragen werden.

Die Spalten dieses Objekts sind fest vorgegeben und können auch nicht aus der Liste entfernt werden. Sollen bestimmte Informationen nicht angezeigt werden, müssen diese über die Eigenschaft Visible unsichtbar gesetzt werden. Die Reihenfolge der Spalten kann beliebig verändert werden.

Folgende Spalten werden erzeugt:

- ClmIcon Symbol des Typs
- ClmName Name des Objekts
- ClmSize Größe des Objekts in Bytes
- ClmLines Anzahl der Zeilen (bei Prozeduren)
- ClmDate Datum der letzten Änderung
- ClmTime Uhrzeit der letzten Änderung
- ClmType Typ des Objekts
- ClmUserCreated Erstellender Benutzer
- ClmUserModified Benutzer der letzten Änderung
- ClmPerm Effektiven Rechte des eigenen Benutzers

Column

Spalteneintrag

Liste,

Eigenschaften

Spalte /

RecList,

Eigenschaften

Spalte /

Siehe <u>RecView</u>,

Eigenschaften

Spalte /

DataList,

Eigenschaften

Spalte /

<u>StoList</u>

Die Objekte <u>RecList</u>, <u>RecView</u>, <u>DataList</u> und <u>StoList</u> bestehen aus einem oder mehreren Spalteneinträgen.

StoListPopup

StoListPopup

Liste,

Siehe Eigenschaften,

Ereignisse

das Objekt StoListPopup entspricht dem Objekt <u>StoList</u>. Das Objekt wird lediglich in einem <u>PopupList</u>-Objekt angelegt.

Das Objekt besitzt zusätzlich die Eigenschaft <u>ClmPopup</u>, womit die Spalte definiert wird, deren Inhalt an das Eingabe-Objekt zurückgegeben wird.

RecList



RecList

Liste,

Eigenschaften,

Ereignisse,

Befehle.

Fragen zur

RecList,

Siehe Ereignisabläufe

RecList,

<u>Aktualisierung</u>

der Inhalte

(Blog),

Editieren einer

Zelle (Blog)

Mit diesem Objekt lassen sich Datensätze einer Datei oder Selektionsmenge in Form einer Liste anzeigen. In Abhängigkeit der darzustellenden Datensätze ist folgendes zu beachten:

• Anzeige Datensätze einer Datei

In der Eigenschaft <u>DbFileNo</u> wird die Dateinummer der gewünschten Datei angegeben. Über die Eigenschaft <u>DbKeyNo</u> wird bestimmt, über welchen Schlüssel die Liste sortiert werden soll.

• Anzeige Datensätze einer verknüpften Datei

Bei einer Verknüpfung wird die Zieldatei in der Eigenschaft <u>DbLinkFileNo</u> und die Nummer der <u>Verknüpfung</u> in der Eigenschaft <u>DbKeyNo</u> angegeben.

Anzeige Datensätze einer Selektionsmenge

Sollen Datensätze einer Selektionsmenge dargestellt werden, wird in der Eigenschaft <u>DbSelection</u> der Deskriptor der Selektion (<u>SelOpen()</u>) angegeben. Da der Deskriptor der Selektion erst zur Laufzeit bekannt ist, kann diese Eigenschaft nicht im Designer gesetzt werden.

• Anzeige verknüpfter Datensätze einer Selektionsmenge

Sollen verknüpfte Datensätze einer Selektionsmenge dargestellt werden, wird in der Eigenschaft <u>DbSelection</u> der Deskriptor der Selektion (<u>SelOpen()</u>) angegeben. Da der Deskriptor der Selektion erst zur Laufzeit bekannt ist, kann diese Eigenschaft nicht im Designer gesetzt werden. In der Selektionsmenge müssen die verknüpften Datensätze als Ergebnismenge enthalten sein.

• Anzeige Datensätze über Filter

Sollen Datensätze angezeigt werden, die einem bestimmten Filterkriterium entsprechen, wird in der Eigenschaft <u>DbFilter</u> der Deskriptor des gesetzten Filters (<u>RecFilterCreate()</u>) angegeben. Da der Deskriptor des Filters erst zur Laufzeit bekannt ist, kann diese Eigenschaft nicht im Designer gesetzt werden.

Anzeige von Grafiken

Über die Eigenschaften <u>TileNameUser</u> und <u>ClmTypeImage</u> können Grafiken in

einer RecList angezeigt werden.

• Anzeige von Feldern aus einer anderen Datei

In dem RecList-Objekt können auch Spalten angegeben werden, die Felder aus einer anderen Datei ausgeben, als in der Eigenschaft <u>DbFileNo</u> angegeben ist. Der Inhalt der Felder muss in dem Ereignis <u>EvtLstDataInit</u> gesetzt werden. Auf die gleiche Weise können auch Abkürzungen in der Liste ausgeschrieben werden.

Die Feldpuffer werden durch die RecList nur dann mit dem selektierten Datensatz gefüllt, wenn beim Ereignis <u>EvtLstSelect</u> eine Funktion angesprungen wird. Der selektierte Datensatz wird erst beim Fokuswechsel in ein anderes Objekt in die Feldpuffer übertragen. Es werden nur die Feldpuffer gesetzt, die in den Spalten der RecList eingetragen sind. Das Übertragen der Feldpuffer beim Fokuswechsel kann durch das Deaktivieren des Flags <u>WinLstRecFocusTermReset</u> in der Eigenschaft <u>LstFlags</u> unterbunden werden.

Wird beim Ereignis <u>EvtLstSelect</u> eine Funktion aufgerufen, werden die Feldpuffer, die in den Spalten der RecList enthalten sind, mit den Werten des selektierten Datensatzes gefüllt. Diese Übertragung der Feldinhalte kann durch das Flag <u>WinLstRecSelectBuffer</u> in der Eigenschaft <u>LstFlags</u> aktiviert und deaktiviert werden.

Über das Ereignis <u>EvtLstRecControl</u> der RecList können bestimmte Sätze von der Anzeige ausgenommen werden.

Bei diesem Objekt ist eine Mehrfachselektion möglich. Zur Verwendung der Mehrfachselektion, muss die Eigenschaft <u>MultiSelect</u> gesetzt werden. Weitere Informationen befinden sich im Abschnitt <u>SelectionData</u>-Objekt.

Im Designer ist die Übernahme der Datenbankfelder in die RecList wie folgt möglich:

- Ziehen eines Feldes aus der Datenstruktur in die Liste.
- Ziehen eines Teildatensatzes aus der Datenstruktur in die Liste.
- Mit Betätigung der Etral Taste beim Ziehen wird automatisch eine RecList erstellt.
- Über den RecList-Editor.

Die Breite des RecList-Objekts ist durch Beschränkungen auf der Ebene des Betriebssystems auf ca. 16.000 Pixel begrenzt. Um ein flexibel verwendbares RecList-Objekt zu erstellen, kann entweder eine RecList mit vielen Spalten angelegt werden, wobei alle Spalten, die nicht benötigt werden, nicht angezeigt werden (Eigenschaft $\underline{\text{Visible}} = \underline{\text{false}}$), oder es werden die Spalten einer RecList an die darzustellenden Felder (Eigenschaft $\underline{\text{DbFieldName}}$) angepasst.

Column

Spalteneintrag

Liste,

Eigenschaften

Spalte /

RecList,

Eigenschaften

Spalte /

Siehe <u>RecView</u>,

Eigenschaften

Spalte /

DataList,

Eigenschaften

Spalte /

<u>StoList</u>

Die Objekte <u>RecList</u>, <u>RecView</u>, <u>DataList</u> und <u>StoList</u> bestehen aus einem oder mehreren Spalteneinträgen.

RecListPopup

RecListPopup

Liste,

Siehe Eigenschaften,

Ereignisse

Mit diesem Objekt wird das <u>RecList</u>-Objekt als Combobox dargestellt. In diesem Fall ist die Liste ein Unterobjekt des <u>PopupList</u>-Objekts. Es lassen sich Datensätze einer Datei oder Selektionsmenge in Form einer Liste anzeigen.

In Abhängigkeit der darzustellenden Datensätze ist folgendes zu beachten:

• Anzeige Datensätze einer Datei

In der Eigenschaft <u>DbFileNo</u> wird die Dateinummer der gewünschten Datei angegeben. Über die Eigenschaft <u>DbKeyNo</u> wird bestimmt, über welchen Schlüssel die Liste sortiert werden soll.

• Anzeige Datensätze einer verknüpften Datei

Bei einer Verknüpfung wird die Zieldatei in der Eigenschaft <u>DbLinkFileNo</u> und die Nummer der Verknüpfung in der Eigenschaft <u>DbKeyNo</u> angegeben.

• Anzeige Datensätze einer Selektionsmenge

Sollen Datensätze einer Selektionsmenge dargestellt werden, wird in der Eigenschaft <u>DbSelection</u> der Deskriptor der Selektion (<u>SelOpen()</u>) angegeben. Da der Deskriptor der Selektion erst zur Laufzeit bekannt ist, kann diese Eigenschaft nicht im Designer gesetzt werden.

• Anzeige verknüpfter Datensätze einer Selektionsmenge

Sollen verknüpfte Datensätze einer Selektionsmenge dargestellt werden, wird in der Eigenschaft <u>DbSelection</u> der Deskriptor der Selektion (<u>SelOpen()</u>) angegeben. Da der Deskriptor der Selektion erst zur Laufzeit bekannt ist, kann diese Eigenschaft nicht im Designer gesetzt werden. In der Selektionsmenge müssen die verknüpften Datensätze als Ergebnismenge enthalten sein.

Anzeige Datensätze über Filter

Sollen Datensätze angezeigt werden die einem bestimmten Filterktiterium entsprechen, wird in der Eigenschaft <u>DbFilter</u> der Deskriptor des gesetzten Filters (<u>RecFilterCreate()</u>) angegeben. Da der Deskriptor des Filters erst zur Laufzeit bekannt ist, kann diese Eigenschaft nicht im Designer gesetzt werden.

Anzeige von Grafiken

Über die Eigenschaften <u>TileNameUser</u> und <u>ClmTypeImage</u> können Grafiken in einem RecListPopup-Objekt angezeigt werden.

• Anzeige von Feldern aus einer anderen Datei

In dem RecListPopup-Objekt können auch Spalten angegeben werden, die Felder aus einer anderen Datei ausgeben, als in der Eigenschaft <u>DbFileNo</u> angegeben ist. Der Inhalt der Felder muss in dem Ereignis <u>EvtLstDataInit</u> gesetzt werden. Auf die gleiche Weise können auch Abkürzungen in der Liste ausgeschrieben werden.

Über das Ereignis <u>EvtLstRecControl</u> der RecListPopup können bestimmte Sätze von der Anzeige ausgenommen werden.

Neben der Maus kann die Liste über die Tastenkombination et per Prozedur über die Eigenschaft PopupOpen aufgeklappt werden.

Die Übernahme der Datenbankfelder in die Liste ist wie folgt möglich:

- Ziehen eines Feldes aus der Datenstruktur in die Liste.
- Ziehen eines Teildatensatzes aus der Datenstruktur in die Liste.
- Über den RecListPopup-Editor

RecView



RecView

Liste,

Eigenschaften,

Ereignisse,

Befehle, Aufbau

eines RecView,

<u>Ereignisabläufe</u>

RecView,

Grundlagen

(Blog)

Siehe Gleichbleibende

Zeilenhöhe

(Blog),

Editieren eines

Items (Blog),

Dynamische

Werte (Blog),

<u>Verknüpfte</u>

<u>Datensätze</u>

(Blog)

Das RecView-Objekt eignet sich für die Darstellung von Inhalten aus der Datenbank mit variabler Zeilenhöhe. Beispielsweise ist die Anzeige von Bildern, Text- und RTF-Inhalten möglich. Die anzuzeigenden Inhalte (Elemente) bestimmen sich aus der Spaltendefinition, die im RecView-Editor konfiguriert wird.

Elemente können mit unbegrenzter Höhe (<u>ContentHeightMax</u> = 0) dargestellt werden oder auf eine Höhe begrenzt werden. Ist die Eigenschaft <u>ContentHeightMax</u> < 0, hat das Element eine feste Höhe, die dem Betrag der Eigenschaft entspricht. Enthält die Eigenschaft einer Wert > 0, dann besitzt das Element eine Maximalhöhe des angegebenen Wertes.

Die Höhe einer Zeile leitet sich aus der Höhe des größten Elementes einer Zeile ab.

In Abhängigkeit der darzustellenden Datensätze ist folgendes zu beachten:

• Anzeige Datensätze einer Datei

In der Eigenschaft <u>DbFileNo</u> wird die Dateinummer der gewünschten Datei angegeben. Über die Eigenschaft <u>DbKeyNo</u> wird bestimmt, über welchen Schlüssel die Liste sortiert werden soll.

• Anzeige Datensätze einer verknüpften Datei

Bei einer Verknüpfung wird die Zieldatei in der Eigenschaft <u>DbLinkFileNo</u> und die Nummer der <u>Verknüpfung</u> in der Eigenschaft <u>DbKeyNo</u> angegeben.

• Anzeige Datensätze einer Selektionsmenge

Sollen Datensätze einer Selektionsmenge dargestellt werden, wird in der Eigenschaft <u>DbSelection</u> der Deskriptor der Selektion (<u>SelOpen()</u>) angegeben. Da der Deskriptor der Selektion erst zur Laufzeit bekannt ist, kann diese

Eigenschaft nicht im Designer gesetzt werden.

• Anzeige verknüpfter Datensätze einer Selektionsmenge

Sollen verknüpfte Datensätze einer Selektionsmenge dargestellt werden, wird in der Eigenschaft <u>DbSelection</u> der Deskriptor der Selektion (<u>SelOpen()</u>) angegeben. Da der Deskriptor der Selektion erst zur Laufzeit bekannt ist, kann diese Eigenschaft nicht im Designer gesetzt werden. In der Selektionsmenge müssen die verknüpften Datensätze als Ergebnismenge enthalten sein.

• Anzeige Datensätze über Filter

Sollen Datensätze angezeigt werden, die einem bestimmten Filterkriterium entsprechen, wird in der Eigenschaft <u>DbFilter</u> der Deskriptor des gesetzten Filters (<u>RecFilterCreate()</u>) angegeben. Da der Deskriptor des Filters erst zur Laufzeit bekannt ist, kann diese Eigenschaft nicht im Designer gesetzt werden.

• Anzeige von Text- und RTF-Inhalten

Wird die Eigenschaft <u>ContentType</u> auf den Wert <u>WinContentTypeText</u> bzw. <u>WinContentTypeRTF</u> gesetzt, können Text- und RTF-Inhalte aus Datenbankfeldern, internen Texten, binären Objekten und externen Dateien angezeigt werden. Die Quelle des Inhaltes wird über die Eigenschaft <u>DbFieldName</u> für Datenbankfelder, oder <u>FileName</u> für andere Quellen definiert.

• Anzeige von Bildern

Über die Eigenschaften <u>ContentType</u> mit dem Wert <u>WinContentTypeImage</u> können Bilder in einem RecView angezeigt werden. Der Pfad des Bildes wird in der Eigenschaft <u>FileName</u> definiert.

• Anzeige von Feldern aus einer anderen Datei

In dem RecView-Objekt können auch Spalten angegeben werden, die Felder aus einer anderen Datei ausgeben, als in der Eigenschaft <u>DbFileNo</u> angegeben ist. Die <u>Caption...</u>-Eigenschaft des jeweiligen <u>(Sub-)Items</u> muss im Ereignis <u>EvtLstGroupInit</u> auf den Inhalt des Felder der anderen Datei gesetzt werden. Auf die gleiche Weise können auch Abkürzungen in der Liste ausgeschrieben werden.

• Anzeige von beliebigen Werten

Zur Laufzeit kann im Ereignis <u>EvtLstGroupInit</u> die Eigenschaft <u>Caption...</u> mit dem jeweiligen Typ direkt gesetzt werden. Somit werden keine Hilfsfelder mehr benötigt.

Über das Ereignis <u>EvtLstRecControl</u> des RecViews können bestimmte Sätze von der Anzeige ausgenommen werden.

Im Designer ist die Übernahme der Datenbankfelder in das RecView wie folgt möglich:

- Ziehen eines Feldes aus der Datenstruktur in das View.
- Ziehen eines Teildatensatzes aus der Datenstruktur in das View.
- Mit Betätigung der und der Taste beim Ziehen wird automatisch ein RecView erstellt.
- Über den RecView-Editor.

Die Breite des RecView-Objekts ist durch Beschränkungen auf der Ebene des Betriebssystems auf ca. 16.000 Pixel begrenzt. Um ein flexibel verwendbares RecView-Objekt zu erstellen, kann entweder ein RecView mit vielen Spalten angelegt werden, wobei alle Spalten, die nicht benötigt werden, nicht angezeigt werden (Eigenschaft <u>Visible</u> = <u>false</u>), oder es werden die Spalten eines RecView an die darzustellenden Felder (Eigenschaft <u>DbFieldName</u>) angepasst.

Column

Spalteneintrag

Liste,

Eigenschaften

Spalte /

RecList,

Eigenschaften

Spalte /

Siehe <u>RecView</u>,

Eigenschaften

Spalte /

DataList,

Eigenschaften

Spalte /

<u>StoList</u>

Die Objekte <u>RecList</u>, <u>RecView</u>, <u>DataList</u> und <u>StoList</u> bestehen aus einem oder mehreren Spalteneinträgen.

RecView-Aufbau Aufbau eines RecView-Objektes Ein RecView-Objekt kann aus folgenden Elementen bestehen:

- Column
- SubColumn
- View
- Group
- Item
- SubItem

Column

Ein <u>RecView</u> kann mehrere Spalten (Columns) enthalten. Diese können im <u>RecView-Editor</u> angelegt und deren <u>Eigenschaften</u> gesetzt werden.

SubColumn

View

Ein RecView enthält 1 bis 4 Anzeigebereiche (Views), in denen die Datensätze dargestellt werden. In jedem View kann die Darstellung variiert werden.

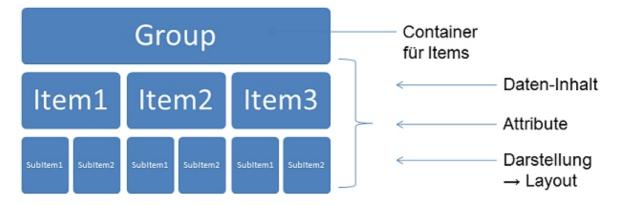
Group

Zur Laufzeit wird für jeden angezeigten Datensatz eine Gruppe (Group) angelegt. Sie ist der Container, der die Items und SubItems enthält. Folgende Eigenschaften stehen bereit:

ExpireTime Zeit in Sekunden, wann frühestens das nächste mal das Ereignis

<u>EvtLstGroupInit</u> für die Gruppe aufgerufen wird.

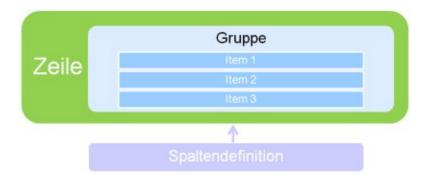
SelectorItemItem eines RecView-ObjektesSelectorSubItemSubItem eines RecView-ObjektesGroupColFgVordergrundfarbe der GruppeGroupColBkgHintergrundfarbe der GruppeGroupDrawSelectSelektionsrahmen zeichnen



Wurde ein Item oder SubItem durch setzen der Eigenschaften <u>SelectorItem</u> und <u>SelectorSubItem</u> gesetzt, können die Eigenschaften des jeweiligen Items bei der Gruppe verändert werden.

Item

Jede Column wird zur Laufzeit bei der Erstellung der Group als Item kopiert. Sie besitzt <u>Eigenschaften</u> die von gleichnamigen <u>Eigenschaften der Spalte</u> kopiert werden. Im Ereignis <u>EvtLstGroupInit</u> können die Eigenschaften des Items geändert werden. Zuvor muss das Item mit der Eigenschaft <u>SelectorItem</u> der Gruppe ausgewählt werden. Die Nummer unter der ein Item angesprochen wird, ist identisch mit der Nummer aus der Spaltendefinition im <u>RecView-Editor</u>. Diese Nummer ist unabhängig von der Anzeigereihenfolge.



SubItem

Jede SubColumn wird zur Laufzeit als SubItem kopiert. Um die <u>Eigenschaften</u> eines SubItems im Ereignis <u>EvtLstGroupInit</u> zu ändern, muss die Eigenschaft <u>SelectorSubItem</u> der Gruppe auf die Nummer des SubItems gesetzt werden.

TreeView

TreeView

Liste,

Siehe $\frac{Eigenschaften}{Ereignisse}$,

Befehle

Mit diesem Objekt werden hierarchische Strukturen dargestellt. Das TreeView-Objekt besteht aus einem oder mehreren Knoten. Die Knoten werden mit dem TreeView-Editor erstellt und geändert.

Über die Eigenschaft NodeStyle kann aus einer Reihe von vordefinierten Symbolen ein Symbol für den jeweiligen Knoten ausgewählt werden. Das Verwenden eigener Symbole ist ebenfalls möglich. Dazu muss eine entsprechende TreeView-Grafik in die Datenbank importiert werden. In der Eigenschaft ImageTileUser wird dann die Nummer der zu verwendenden Kachel angegeben.

Das Anlegen und Entfernen der Knoten während der Laufzeit erfolgt über die Befehle WinTreeNodeAdd() und WinTreeNodeRemove(). Besitzt ein Knoten untergeordnete angezeigt, wenn bei dem Knoten die Eigenschaft NodeDynamic gesetzt ist.

Bei diesem Objekt ist eine Mehrfachselektion möglich. Zur Verwendung der Mehrfachselektion, muss die Eigenschaft MultiSelect gesetzt werden. Weitere Informationen befinden sich im Abschnitt SelectionData-Objekt.

Im TreeView kann eine Suche durchgeführt werden. Dazu muss die Eigenschaft SearchEnabled auf true gesetzt sein. Mit den, in den Eigenschaften SearchKevStart, SearchKeyNext und SearchKeyPrey definieren, Tasten wird die Suche gestartet und fortgesetzt. Die Suchoptionen werden mit der Eigenschaft SearchFlags festgelegt. Standardmäßig wird mit einem Wildcard-Vergleich geprüft, ob die Eigenschaft Caption den Suchtext enthält. Die Groß-/Kleinschreibung wird dabei ignoriert. Die Suche kann mit der Funktion WinTreeNodeSearch() auch prozedural gestartet werden.

TreeNode

TreeNode

Siehe $\frac{\text{Liste}}{\text{Eigenschaften}}$

Das Objekt <u>TreeView</u> besteht aus mehreren Knoten. Das Anlegen und Entfernen der Knoten während der Laufzeit erfolgt über die Befehle WinTreeNodeAdd() und WinTreeNodeRemove().

Ausgabe-Objekte

Siehe Alle Oberflächen-Objekte

<u>Label</u>

abo

<u>Icon</u>



Picture



 $\underline{MetaPicture}$



Animation



DocView



WebNavigator (



<u>PrtJobPreview</u>



<u>Barcode</u>



Chromium



Label

Bezeichner

Liste,

Siehe Eigenschaften,

Ereignisse

Mit einem Label kann ein kurzer, in der Regel statischer Text ausgegeben werden. Der Text des Labels wird in der Eigenschaft <u>Caption</u> gesetzt werden. Das Zeichen & wird verwendet, um einen "Hot-Spot" zu definieren. Der nachfolgende Buchstabe wird unterstrichen. Ist in der Eigenschaft <u>ObjLink</u> ein Eingabe-Objekt eingetragen, wird beim Drücken der Tastenkombination <u>at text in der Eigenschaft</u> + <unterstrichener Buchstabe > der Fokus in das Eingabe-Objekt versetzt. Um das Zeichen & auszugeben, muss es zwei mal geschrieben werden.

In der Eigenschaft <u>DbFieldName</u> kann ein Datenbankfeld von beliebigem Typ mit dem Eingabeobjekt verknüpft werden. Der Inhalt des Feldes wird dann anstelle der <u>Caption</u> angezeigt. Eine Auswertung des & findet dann nicht statt.

Icon



Symbol

Liste,

Siehe Eigenschaften,

Ereignisse

Über das Objekt Icon lassen sich Piktogramme vom Typ ICO anzeigen. Die ICO-Datei kann dabei extern oder in der Datenbank vorliegen.

In der Eigenschaft <u>FileName</u> wird der Name der darzustellenden Datei angegeben. Wird eine Bilddatei bestehend aus mehreren Icons verwendet, ist in der Eigenschaft <u>NumIcon</u> die Bildnummer der Icon-Grafik anzugeben. Wird die Eigenschaft auf -1 gesetzt, wird die Eigenschaft <u>TileSize</u> verwendet, um die Größe zu definieren. Handelt es sich bei dem Symbol um eine externe Datei muss in der Eigenschaft <u>TypeIcon</u> die Einstellung <u>WinIcoExternFile</u> verwendet werden.

Über diese Eigenschaft kann auch aus einer Reihe von vorgegebenen Standard-Symbolen (Warnsymbol, Fehlersymbol ...) gewählt werden.

Picture

Marstellen von Grafiken 🚵

Liste,

Eigenschaften,

Ereignisse,

PrtPicture,

<u>Darstellung</u>

Siehe von

Kennzahlen

(Blog),

Grafiken in

CONZEPT 16

(Blog)

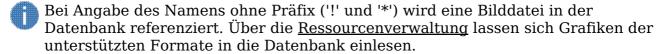
In diesem Objekt können über die Eigenschaft <u>Caption</u> Grafiken vom Format BMP, JPG, GIF, PNG, TIFF (ein und mehrseitig) und ICO dargestellt werden. Die anzuzeigende Grafik kann extern oder in der Datenbank vorliegen. Die Angabe einer externen Datei kann mit einem absoluten oder relativen Pfad erfolgen. Bei der Angabe eines absoluten Pfades wird ein '*', bei der eines relativen Pfades (relativ zum Startverzeichnis des CONZEPT 16-Clients) wird ein '!' vorangestellt. Bei mehrseitigen TIFF- und ICO-Bildern, wird die anzuzeigende Seite in der Eigenschaft <u>CurrentInt</u> angegeben.

In einem Picture-Objekt können weitere Picture- oder <u>MetaPicture</u>-Objekte eingefügt werden (Overlays). Dies erfolgt im Designer, indem ein Objekt selektiert wird und anschliessend über die <u>Palette</u> ein entsprechendes Objekt eingefügt wird. Untergeordneten Objekten können keine weiteren Objekte hinzugefügt werden.

Durch Setzen der Eigenschaft <u>ColBkg</u> eines untergeordneten Objekts auf <u>WinColTransparent</u> wird der Inhalt des übergeordneten Objekts im untergeordneten Objekt gezeichnet. Die Eigenschaft <u>Opacity</u> des untergeordneten Objekts bestimmt in diesem Fall die Deckfähigkeit zum Hintergrund (dem Bild des übergeordneten Bildes).

Auf diese Weise können z. B. einfach Stempel oder Wasserzeichen für die Anzeige eines Bildes generiert werden.

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



Beispiele:

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

Wird ein Bild dargestellt, dass größer als das Objekt ist, erscheinen automatisch

Scrollbalken auf der rechten und der unteren Seite des Objektes (die Rollbalken können mit der Eigenschaft <u>ScrollbarVisible</u> deaktiviert werden). Bei Angabe von 0 bei der Eigenschaft <u>ZoomFactor</u> wird das Bild so skaliert, dass es komplett im Picture-Objekt dargestellt wird.

Neben der üblichen Funktionalität mit der linken Maustaste, kann das Bild mit dem Rad einer Wheelmaus gescrollt werden: Durch Drehen des Rades wird vertikal gescrollt. Erfolgt das Drehen bei gedrückter —-Taste wird horizontal gescrollt. Das Scrolling erfolgt pixelweise. Um ein schnelleres Scrolling zu erreichen, muss zusätzlich die —-Taste gedrückt gehalten werden.

Zusätzlich werden noch folgende Tastenkommandos akzeptiert:

Taste	Wirkung		
+	Bild eine Stufe heranzoomen		
	Bild eine Stufe herauszoomen		
strg) + +	Höchste Zoomstufe		
strg) + -	niedrigste Zoomstufe		
①	Bild nach oben scrollen		
•	Bild nach unten scrollen		
P	Bild nach links scrollen		
P	Bild nach rechts scrollen		
pos1	Bild an den Anfang scrollen		
ende	Bild an das Ende scrollen		
bilda	Bild schneller nach oben scrollen		
bild⊽	Bild schneller nach unten scrollen		
scrollens.	ändert in Kombination mit den Pfeiltasten die Geschwindigkeit des Bei den Tasten post, ende , bilder und kann die Richtung der vegung durch Drücken der Taste umgekehrt werden.		
Ist das Ereignis EvtKevItem beim Picture-Objekt eingetragen, entscheidet der			

MetaPicture

💐 Objekt zur Darstellung von Vektorgrafiken

Liste,

Eigenschaften,

Siehe $\frac{\text{Ereignisse}}{\text{Grafiken in}}$

CONZEPT 16

(Blog)

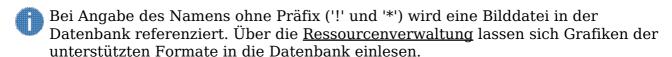
In diesem Objekt können über die Eigenschaft Caption Vektorgrafiken im Metafile-Format (EMF, WMF) dargestellt werden. Die anzuzeigende Grafik kann extern oder in der Datenbank vorliegen. Die Angabe einer externen Datei kann mit einem absoluten oder relativen Pfad erfolgen. Bei der Angabe eines absoluten Pfades wird ein '*', bei der eines relativen Pfades (relativ zum Startverzeichnis des CONZEPT 16-Clients) wird ein '!' vorangestellt.

In einem MetaPicture-Objekt können weitere Picture- oder MetaPicture-Objekte eingefügt werden (Overlays). Dies erfolgt im Designer, indem ein Objekt selektiert wird und anschliessend über die Palette ein entsprechendes Objekt eingefügt wird. Untergeordneten Objekten können keine weiteren Objekte hinzugefügt werden.

Durch Setzen der Eigenschaft ColBkg eines untergeordneten Objekts auf <u>WinColTransparent</u> wird der Inhalt des übergeordneten Objekts im untergeordneten Objekt gezeichnet. Die Eigenschaft Opacity des untergeordneten Objekts bestimmt in diesem Fall die Deckfähigkeit zum Hintergrund (dem Bild des übergeordneten Bildes).

Auf diese Weise können z. B. einfach Stempel oder Wasserzeichen für die Anzeige eines Bildes generiert werden.

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



Beispiele:

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

Wird ein Bild dargestellt, dass größer als das Objekt ist, erscheinen automatisch Rollbalken auf der rechten und der unteren Seite des Objektes (die Rollbalken können mit der Eigenschaft ScrollbarVisible deaktiviert werden). Bei Angabe von 0 bei der Eigenschaft ZoomFactor wird das Bild so skaliert, dass es komplett im MetaPicture-Objekt dargestellt wird.

Neben der üblichen Funktionalität mit der linken Maustaste, kann das Bild mit dem Rad einer Wheelmaus gescrollt werden: Durch Drehen des Rades wird vertikal

gescrollt. Erfolgt das Drehen bei gedrückter —-Taste wird horizontal gescrollt. Das Scrolling erfolgt pixelweise. Um ein schnelleres Scrolling zu erreichen, muss zusätzlich die strg -Taste gedrückt gehalten werden.

Zusätzlich werden noch folgende Tastenkommandos akzeptiert:

Taste	Wirkung
+	Bild eine Stufe heranzoomen
	Bild eine Stufe herauszoomen
strg) + +	Höchste Zoomstufe
strg] + -	niedrigste Zoomstufe
Û	Bild nach oben scrollen
4	Bild nach unten scrollen
D	Bild nach links scrollen
\$	Bild nach rechts scrollen
pos1	Bild an den Anfang scrollen
ende	Bild an das Ende scrollen
bilda	Bild schneller nach oben scrollen
[bildv]	Bild schneller nach unten scrollen
scrollens.	ändert in Kombination mit den Pfeiltasten die Geschwindigkeit des Bei den Tasten post, ende, bidd und bidd kann die Richtung der vegung durch Drücken der Taste umgekehrt werden.

Ist das Ereignis <u>EvtKeyItem</u> beim MetaPicture-Objekt eingetragen, entscheidet der Rückgabewert darüber, ob die Scrolltasten verarbeitet werden (<u>true</u>) oder nicht (<u>false</u>). Die Zoomfunktion der — / — Tasten wird immer durchgeführt.

Animation

.AVI Datei-Objekt

Liste,

Siehe Eigenschaften,

Ereignisse

In diesem Objekt können AVI-Dateien (Audio Video Interleave) abgespielt werden. Der Film ist in der Eigenschaft <u>FileName</u> anzugeben. Über die Eigenschaft <u>TypeFile</u> steht eine Liste verschiedener Animationen des Betriebssystems zur Verfügung.

DocView

Anzeige-Objekt für Bilder und PDF-Dokumente

Siehe $\frac{Eigenschaften}{Ereignisse}$,

PrtPdf

Das DocView-Objekt ermöglicht die Anzeige von PDF-Dokumenten ohne dass hierzu entsprechende Software eines anderen Anbieters auf dem Rechner installiert sein muss. Neben der Anzeige von PDF ist auch die Anzeige von TIFF, IPEG, PNG, BMP und GIF möglich.

Der Pfad und der Name des Dokuments wird in der Eigenschaft FileName angegeben.

Das DocView-Objekt erlaubt eine Galerie-Anzeige mehrerer PDF- bzw. TIFF-Seiten. Es werden dabei mehrere Seiten dargestellt. Die Anzahl der Seiten, die nebeneinander dargestellt werden, wird über die Eigenschaft PageNumClm definiert.

Die Hintergrundfarbe der Seiten kann über die Eigenschaft ColBkg gesetzt werden; die Farbe des Seitenrandes über die Eigenschaft ColBoundary.

Die Eigenschaft PageZoom gibt an, wie die Seite oder die Seiten dargestellt werden:

Benutzerdefinierter Zoom PrtPageZoomUser <u>PrtPageZoomPage</u> Zoom auf ganze Seite Zoom auf Seitenbreite <u>PrtPageZoomPageWidth</u> PrtPageZoomPageAll Zoom auf alle Seiten

PrtPageZoomPageWidthAll Zoom auf Seitenbreite aller Seiten

Innerhalb des Objekts können zusätzliche Picture- oder MetaPicture-Objekte eingefügt werden (Overlay). Dies erfolgt im Designer, indem das DocView-Objekt selektiert wird und anschließend über die <u>Palette</u> ein Objekt eingefügt wird. In den Eigenschaften ViewId und ViewType kann angegeben werden, unter welchen Bedingungen das Overlay angezeigt wird.

WebNavigator

Anzeige registrierter Dateitypen

Liste

Eigenschaften,

Ereignisse,

Siehe Kompatibilität

(Blog),

Steuerung

(Blog)

In diesem Objekt können beliebige registrierte Dateitypen angezeigt werden. Die Datei wird innerhalb des WebNavigators angezeigt und kann je nach verwendetem Programm auch geändert werden.

In der Eigenschaft <u>Caption</u> wird der Dateiname der anzuzeigenden Datei oder eine Web-Adresse (URL) angegeben.

Ist bei der Initialisierung des Objekts ein Fehler aufgetreten, kann dieser mit der Anweisung <u>ComInfo()</u> ermittelt werden.

Mit dem Befehl <u>WinWbnExecCommand()</u> können Kommandos für die Navigation und die Zwischenablage an den WebNavigator gesendet werden.

PrtJobPreview PrtJobPreview

Liste.

Siehe Eigenschaften,

Ereignisse

In diesem Objekt können über die Eigenschaft Caption Druckjob-Dateien und in der Datenbank gespeicherte PrintDoc-Objekte angezeigt werden. Der Druckjob muss zu diesem Zeitpunkt geschlossen (PrtIobClose()) sein.

Die Angabe einer extern vorliegenden Druckjob-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. Liegt die Datei als binäres Objekt vor, können diese mit dem Präfix '>' angesprochen werden (siehe Caption).

Bei Angabe des Namens ohne Präfix ('!', '*' und '>') wird ein PrintDoc-Objekt aus der Datenbank geladen.

Mit der Eigenschaft CryptKey kann ein Passwort gesetzt werden, welches benötigt wird, wenn ein verschlüsselter Druckjob gelesen werden soll.

Die Eigenschaft CurrentInt definiert die Nummer der anzuzeigenden Seite. MaxInt kann nur gelesen werden und liefert die Anzahl der Seiten. Diese Eigenschaften können nur zur Laufzeit gesetzt werden.

Wird in dem Druckjob ein PrtRtf-Objekt ausgegeben, kann es in der Vorschau zu Abweichungen gegenüber dem Druckergebnis kommen, da noch nicht bekannt ist, über welchen Drucker der Ausdruck erfolgt. Ist zum Zeitpunkt der Anzeige des Druckjobs bereits bekannt, auf welchem Drucker das Dokument gedruckt werden soll, können die Abweichungen zwischen Vorschau und Druckergebnis minimiert werden, in dem der Deskriptor des PrintDevice-Objekts der Eigenschaft PrtDevice zugewiesen wird.

Bei Verwendung einer Wheel-Maus werden folgende zusätzliche Funktionen unterstützt:

In dem PrtJobPreview-Objekt kann die Anzahl der Vorschauseiten variiert werden. Durch Angabe der Eigenschaft PageCount kann definiert werden, wie viele Seiten ab der, durch die Eigenschaft <u>CurrentInt</u> gesetzten, Startseite angezeigt werden sollen. Durch die Eigenschaft PageNumClm kann angegeben werden, wie viele Seiten maximal nebeneinander dargestellt werden.

Barcode

Barcode anzeigen

Liste,

Siehe $\frac{Eigenschaften}{Ereignisse}$,

Befehle

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

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

Beispiele:

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

Chromium

Web-Browser-Objekt

Liste, Eigenschaften,

Siehe Ereignisse, CodeLibrary-Beispiel

"ChromeBrowser"

Das Chromium-Objekt ermöglicht die Anzeige Web-basierter Inhalte.

Das Objekt stellt eine moderne Alternative zum <u>WebNavigator</u>-Objekt dar und basiert auf dem quelloffenen Chromium-Projekt auf dem auch der Browser Chrome basiert. Das Chromium-Projekt ist über das Chromium Embedded Framework in CONZEPT 16 integriert.

Das Objekt ist optional und kann über die Setup-Routine als eigentständiges Paket installiert werden.

Die Anzeige einer Web-Seite geschieht durch setzen der Eigenschaft Caption. Mit dem Befehl WinCroNavigate kann zu einer zuvor angezeigten Seite navigiert werden. Mit WinCroReload kann die aktuelle Seite erneut geladen werden.

Die Entwickler-Werkzeuge können mit den Eigenschaften CroDevToolsSide und CroDevToolsSize aktiviert werden.

Eigenschaften eines Chromium-Objekts

Das Objekt Chromium unterstützt folgende Eigenschaften.

Chromium,

<u>Liste der</u>

Siehe Objekte, Liste

<u>der</u>

Eigenschaften

- AlignGrouping
- AlignHeight
- <u>AlignMarginBottom</u>
- AlignMarginLeft
- AlignMarginRight
- AlignMarginTop
- AlignWidth
- AreaBottom
- AreaHeight
- AreaLeft
- AreaRight
- <u>AreaTop</u>
- AreaWidth
- Caption
- ColDesign
- CroDevToolsSide
- CroDevToolsSize
- Custom
- Design
- DesignFlags
- Disabled
- Name
- StyleBorder
- TabPos
- Url
- Visible
- **ZoomFactor**

Ereignisse eines Chromium-Objekts

Das Objekt <u>Chromium</u> unterstützt folgende Ereignisse.

Chromium,

<u>Liste der</u>

Siehe Objekte,

Liste der

Ereignisse

- EvtChanged
- EvtChangedDesign
- EvtCroNavigate
- EvtKeyItem
- EvtMouse

Toolbar-Objekte
Siehe Alle
Oberflächen-Objekte
FrameClient
ToolbarDock
Toolbar
Toolbar für Copy Paste
Toolbar für MDI
ToolbarRtf
ToolbarMenu
Windowbar

<u>Statusbar</u>

FrameClient

FrameClient

Liste,

Siehe Eigenschaften,

Ereignisse

Das FrameClient-Objekt kann nur in ein <u>Frame-</u> oder <u>MdiFrame-Objekt</u> eingefügt werden.

Es dient dazu, das Frame-Objekt so aufzuteilen, dass ein oder mehrere <u>Toolbar-Objekte</u> im Frame neben den anderen im Frame schon vorhandenen Objekten Platz haben.

Wird ein FrameClient-Objekt einem Frame-Objekt hinzugefügt und besitzt das Frame-Objekt bereits untergeordnete Objekte, so werden alle Unterobjekte in das FrameClient-Objekt eingefügt. Nachfolgende Operationen (Drag & Drop, Copy & Paste, Erstellen von Objekten) geschieht dann im FrameClient, da dieser von nun an die Unterobjekte enthält.

In ein <u>AppFrame-Objekt</u> kann kein FrameClient-Objekt eingefügt werden und ist dort auch nicht notwendig.

ToolbarDock

ToolbarDock

Liste,

Siehe Eigenschaften,

Ereignisse

Das ToolbarDock-Objekt dient als Container für <u>Toolbar-Objekte</u>. Ein ToolbarDock-Objekt kann maximal fünf Toolbar-Objekte aufnehmen. Ein Toolbar-Objekt kann nicht in ein ToolbarDock-Objekt eingefügt werden, wenn dieses bereits ein Statusbar-Objekt enthält.

Umgekehrt ist es auch nicht möglich ein Statusbar-Objekt in ein ToolbarDock-Objekt einzufügen, das bereits ein oder mehrere Toolbars enthält.

Das ToolbarDock-Objekt kann in alle Frame-Objekte (<u>Frame</u>, <u>MdiFrame</u> und <u>AppFrame</u>) eingesetzt werden.

Durch die Eigenschaft <u>DockSide</u> kann bestimmt werden, an welcher Seite des Fensters das ToolbarDock-Objekt dargestellt werden soll. Um im unteren Bereich des Fensters sowohl eine Statuszeile, als auch eine Toolbar anzuzeigen, müssen zwei ToolbarDock-Objekte erzeugt und die Eigenschaft <u>DockSide</u> einmal auf <u>WinDockBottom und einmal auf WinDockBottomTop</u> gesetzt werden.

Bei horizontalen ToolbarDock-Objekten, kann die Höhe, bei vertikalen ToolbarDock-Objekten die Breite beliebig verändert werden. Zur Laufzeit passt sich die Höhe bzw. die Breite des Objektes an die beinhalteten Objekte an.

Toolbar

Toolbar

Liste,

Siehe Eigenschaften,

Ereignisse

Bei dem Toolbar-Objekt handelt es sich um einen Symbolbalken mit <u>Buttons</u>, die auf Mausklick oder eine <u>Funktionstaste</u> zugeordnete Funktion auslösen. Bevor das Toolbar-Objekt angelegt werden kann, muss das Objekt <u>ToolbarDock</u> in dem Frame eingebunden werden. Eine Toolbar kann am oberen, unteren, rechten und linken Rand eines Fensters dargestellt werden.

Die Buttons einer Toolbar werden mit dem <u>Toolbar-Editor</u> bearbeitet. Durch Doppelklick auf die Toolbar oder Auswahl des Menüpunktes "Toolbar bearbeiten" im Kontextmenü wird dieser erreicht.

Über die Palette können bereits mit Schaltflächen versehene Toolbars eingefügt werden.

Alle Schaltflächen sind vom Typ <u>Toolbar-Button</u> mit den entsprechenden Eigenschaften und Ereignissen.

Die <u>Toolbar für das RTF-Objekt</u> ist ein separates Objekt, das nicht über den <u>Editor</u> für die Toolbar bearbeitet werden kann.

Toolbar-Button Toolbar-Button

Siehe <u>Liste</u>, <u>Eigenschaften</u>

Eine Toolbar besteht aus einem oder mehreren Toolbar-Buttons. Das Auswählen eines Buttons kann mit der Maus oder über eine Funktionstaste erfolgen.

Für die Auswahl über Funktionstaste wird in der Eigenschaft MenuKey die gewünschte Funktionstaste angegeben. Neben einer Beschriftung (ImageOption) kann ein Button auch über ein Symbol verfügen. In der Eigenschaft ImageTile kann aus einer Reihe von vordefinierten Symbolen gewählt werden.

Das Verwenden eigener Symbole ist ebenfalls möglich. Dazu muss eine entsprechende Grafik in die Datenbank importiert und den Eigenschaften TileNameNormal, <u>TileNamePressed</u> und <u>TileNameSelected</u> zugewiesen werden. In der Eigenschaft ImageTileUser wird dann die Nummer der zu verwendenden Kachel angegeben. Ist die Werkzeugleiste zu klein, um alle Schaltflächen anzeigen zu können, wird am rechten bzw. unteren Rand der Werkzeugleiste die ">>" Schaltfläche angezeigt. Wird diese betätigt, erscheint ein Popup-Menü, sofern bei den nicht sichtbaren Schaltflächen die Eigenschaft Caption einen Wert enthält. Zusätzlich wird links neben dem Text ein Bild (Eigenschaft ImageTile) angezeigt, sofern vorhanden. Ist die Eigenschaft TileNameMenu gesetzt, werden nicht die Bilder des Toolbar-Objektes sondern die Bilder der hier angegebenen Grafik verwendet. Die Nummer der Kachel ergibt sich weiterhin aus der Eigenschaft ImageTile.

Buttons können nur im Toolbar-Editor erstellt werden.

Durch Anklicken eines Toolbar-Buttons (oder durch Eingeben des Tastatur-Kürzels) wird das Ereignis EvtMenuCommand des Fenster-Objektes ausgelöst. Hier kann der ausgewählte Button ermittelt und die entsprechende Funktionalität aufgerufen werden. Das Ereignis EvtMenuCommand beim Toolbar-Button wird nur ausgelöst, wenn ein Eintrag aus dem Kontextmenü des Buttons ausgewählt wird.

Toolbar für Copy Paste

Toolbar für Copy Paste

Liste,

Siehe <u>Ereignisse</u>,

Toolbar

Diese Toolbar enthält folgende Schaltflächen:

Darstellung Name

X tbnEditCut

tbnEditCopy 📑

tbnEditPaste

tbnEditDelete 🔓

tbnEditUndo 🖍

tbnEditRedo 🖎

Toolbar für M	1DI				
<u>Liste</u> ,					
Siehe <u>Eigenschaften</u> , <u>Ereignisse</u> ,					
Ereignisse,					
<u>Toolbar</u>					
Diese Toolbar enthält folgende Schaltflächen:					
Name	Darstellung				
	3				
tbn Mdi Cascade	•				
tbnMdiCascade tbnMdiArrange					
tbn Mdi Arrange					
$tbn M di Arrange \\ tbn M di Tile Vert$					

ToolbarRtf ToolbarRtf

Liste.

Siehe $\frac{Eigenschaften}{Ereignisse}$,

Toolbar

Dieses Toolbar-Objekt ist speziell zur Verwendung mit dem RtfEdit-Objekt vorbereitet. Wird in ein Frame- oder MdiFrame-Objekt, in dem bereits ein RtfEdit-Objekt existiert, diese Toolbar eingefügt, verbindet sie sich automatisch mit dem bereits existierenden Objekt. Für die Verbindung wird die Eigenschaft ObjLink verwendet.

Bevor das Toolbar-Objekt angelegt werden kann, muss das Objekt ToolbarDock in dem Frame eingebunden werden. Eine Toolbar kann am oberen, unteren, rechten und linken Rand eines Fensters dargestellt werden.

Die Toolbar stellt alle notwendigen Funktionen zum Editieren und Formatieren des Textes zur Verfügung.

In einem ToolbarRtf-Objekt sind folgende Objekte integriert:

Name	Тур	Darstellung	
edFontName	$\underline{FontNameEdit}$	Arial	¥
edFontSize	<u>FontSizeEdit</u>	10 💌	
btnRtfBold	<u>Button</u>	B	
btnRtfItalic	<u>Button</u>	I	
btnRtfUnderline	<u>Button</u>	U	
btnRtfAlignLeft	<u>Button</u>	MARIANA MARIANA MARIANA MARIANA MARIANA	
btnRtfAlignCenter	<u>Button</u>	MARIANA MARIANA MARIANA MARIANA MARIANA	
btnRtfAlignRight	<u>Button</u>	畫	
btnRtfAlignJustify	<u>Button</u>	MARIAMAN MARIAMAN MARIAMAN MARIAMAN MARIAMAN	
btnRtfColFg	<u>ColorButton</u>		
btnRtfColBkg	ColorButton		

Nicht benötigte Funktionen können aus der Toolbar über das Kontexmenü des Obiektes oder mit der entfernt werden. Bei Objekten vom Typ Button kann die Funktion über den Menüeintrag RTF-Toolbar des Kontextmenüs geändert werden.

In die Toolbar können weitere Objekte beliebigen Typs aufgenommen werden.

ToolbarMenu

ToolbarMenu

Liste,

Siehe Eigenschaften,

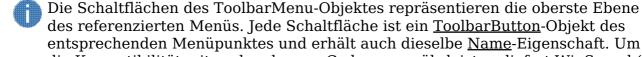
Ereignisse

Im Modern Theme Style kann das klassische (von Windows generierte) Menü nicht mehr angezeigt werden. Eine an das klassische Menü angelehnte Interaktion und Darstellungsweise bietet hier das ToolbarMenu-Objekt. Wird es in ein ToolbarDock-Objekt eingefügt, wird das vorhandene Menü ausgeblendet und das ToolbarMenu-Objekt übernimmt die Menüfunktion.

Das Objekt besteht aus einer Toolbar, welche die Menüpunkte des im Frame eingesetzten Menüs (Eigenschaft MenuName) besitzt. Wird das Menü geändert, indem z. B. durch WinMenuItemAdd() einer neuer Menüpunkt hinzugefügt oder die Eigenschaft MenuName neu gesetzt wird, dann aktualisiert sich das ToolbarMenu-Objekt entsprechend.

Kompatibilität

Damit die Kompatibilität mit dem klassischen Menü gewährleistet ist, löst das ToolbarMenu-Objekt dieselben Ereignisse aus, wie das Menu-Objekt. Somit kann in den meisten Fällen, das Menü ohne programmtechnische Änderungen durch das ToolbarMenu-Objekt ersetzt werden. Ein Frame kann maximal ein ToolbarMenu-Objekt enthalten.



des referenzierten Menüs. Iede Schaltfläche ist ein ToolbarButton-Objekt des entsprechenden Menüpunktes und erhält auch dieselbe Name-Eigenschaft. Um die Kompatibilität mit vorhandemem Code zu gewährleisten, liefert WinSearch() und der \$-Operator / \$:-Operator jedoch das entsprechende MenuItem und nicht den ToolbarButton.

Wird das ToolbarMenu-Objekt in einem AppFrame verwendet, dann werden die für MDI typischen Schaltflächen (rechts vom Menü) über das ToolbarMenu-Objekt emuliert.

Windowbar Windowbar

Liste,

Siehe Eigenschaften,

Ereignisse

Das Windowbar-Objekt kann nur in ein <u>ToolbarDock-Objekt</u> eingefügt werden. Es dient als Container-Objekt für beliebige andere Objekte. Es können alle Objekte, die in ein <u>Frame</u>-Objekt integriert werden können, ebenfalls in ein Windowbar-Objekt eingefügt werden.

Mit diesem Objekt wird es möglich innerhalb einer Werkzeugleiste zum Beispiel eine ComboBox oder ein <u>TreeView</u> zur Navigation innerhalb der Applikation zu integrieren.

Statusbar Statusbar

Liste,

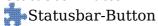
Siehe Eigenschaften,

Ereignisse

Die Statusbar dient der Anzeige von Informationen und kann nur an der unteren Seite eines Frame-Objekts dargestellt werden. Die Statusbar besteht aus einem oder mehreren Statusbar-Buttons.

Zum Bearbeiten einer Statusbar steht der Statusbar-Editor zur Verfügung.

Statusbar-Button



Siehe $\frac{\text{Liste}}{\text{Eigenschaften}}$

Eine Statusbar besteht aus einem oder mehreren Statusbar-Buttons. In der Eigenschaft Caption können folgende Makrodefinitionen angegeben werden:

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

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

```
$(KEY,NUM) // Anzeige des Toggle-Status der NumLock-Taste
$(KEY, INS) // Anzeige des Toggle-Status der Einfg.-Taste
$(KEY,CAP) // Anzeige des Toggle-Status der Umschalt-Taste
```

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

Neben einer Beschriftung (ImageOption) kann ein Button auch über ein Symbol verfügen. In der Eigenschaft ImageTile kann aus einer Reihe von vordefinierten Symbolen gewählt werden.

Das Verwenden eigener Symbole ist ebenfalls möglich. Dazu muss eine entsprechende Grafik in die Datenbank importiert werden. In der Eigenschaft ImageTileUser wird dann die Nummer der zu verwendenden Kachel angegeben.

Buttons können nur im Statusbar-Editor erstellt werden.

Anordnungs-Objekte				
Siehe <u>Alle</u> <u>Oberfläch</u>				
Oberfläch	<u>1en-Objekte</u>			
<u>GroupBox</u>	L			
<u>Notebook</u>				
<u>NotebookPage</u>				
<u>GroupSplit</u>				
<u>GroupTile</u>				
<u>Scrollbox</u>				
Divider	[]			

GroupBox

200

Gruppe

Liste,

Siehe Eigenschaften,

Ereignisse

Die Gruppe dient der Zusammenfassung von Objekten, um deren logische Zusammengehöhrigkeit auszudrücken. Über die Eigenschaft <u>Caption</u> kann eine Gruppenüberschrift angegeben werden.

Ob untergeordnete Objekte den Rahmen der Groupbox überdecken, kann über die Option <u>WinAppExtTransparentBkg</u> der Eigenschaft <u>wpFlagsExt</u> des <u>App</u>-Objektes definiert werden.

Notebook

Notizbuch

Liste,

Siehe Eigenschaften,

Ereignisse

Das Notebook eignet sich besonders gut zur Darstellung von Informationen, die in logisch zusammenhängenden Gruppen organisiert werden können. Zur Laufzeit kann der Wechsel der Notizbuchseiten durch Klicken mit der Maus auf den Registerreiter oder durch die Tastenkombination bzw. strg + bzw. erfolgen.

Im Designer kann durch Klick mit der rechten Maustaste auf den Registerreiter dessen Kontextmenü gestartet werden. In diesem Menü können <u>Registerseiten</u> eingefügt und geändert werden.

Das standardmäßige Verhalten des Notebook-Objekts kann über die Eigenschaft <u>Flags</u> geändert werden.

Bei der Anzeige von Bildern auf den Registerreitern (siehe <u>ImageTile</u> und <u>ImageTileUser</u>) wird die Angabe der Konstanten <u>WinImgLeftTextRight</u> bei der Eigenschaft <u>ImageOption</u> nicht unterstützt.

NotebookPage

Seite eines Registerreiter

Liste,

Siehe Eigenschaften,

Ereignisse

Dieses Objekt hat als Vaterobjekt immer ein Notebook.

Der Titel der Seite wird mit der Eigenschaft Caption gesetzt.

Auf den Registerreitern können Bilder (siehe <u>ImageTile</u> und <u>ImageTileUser</u>) angezeigt werden. Die Größe der mitgelieferten Bilder wird über die Eigenschaft <u>TileSize</u> für <u>WinTypeNotebook</u> definiert.

Sollen benutzerdefinierte Bilder angezeigt werden, müssen die Eigenschaften <u>TileNameNormal</u> und <u>TileNameSelected</u> des <u>Notebook</u>-Objektes gesetzt werden. Die Bilder können in der <u>Ressourcenverwaltung</u> als <u>Kachelgrafik für Schaltflächen- und Listenobjekte</u> importiert werden.

GroupSplit

Objekt zur dynamischen Aufteilung des Fensterbereichs

Siehe $\frac{Eigenschaften}{Ereignisse}$,

GroupTile

Das GroupSplit-Objekt erlaubt die dynamische Aufteilung des Darstellungsbereiches in bis zu vier Einzelbereiche. Jeder Einzelbereich besteht aus einem GroupTile-Objekt, das im GroupSplit-Objekt angeordnet wird. Zur Laufzeit können die untergeordneten -Objekte vom Benutzer vergrössert oder verschoben werden.

Beim Setzen von Eigenschaften des GroupSplit-Objekts ist je nach Eigenschaft folgendes zu beachten:

StyleGroup

Es darf kein maximiertes GroupTile-Objekt geben. Es darf kein GroupTile-Objekt geschlossen sein.

AreaMarginLeft, AreaMarginTop, AreaMarginRight, AreaBottom

Der übergebene Wert muss größer oder gleich Null betragen.

Vertical

Es darf kein maximiertes <u>GroupTile</u>-Objekt geben. Es darf kein GroupTile-Objekt geschlossen sein.

GroupTile

Dynamischer Einzelbereich

Liste.

Siehe $\frac{Eigenschaften}{Ereignisse}$,

GroupSplit

Das GroupTile-Objekt stellt einen Einzelbereich des GroupSplit-Objekts dar. Es ist widerum in der Lage weitere Objekte aufzunehmen. Auch das Einfügen eines weiteren GroupSplit-Objekts in das GroupTile-Objekt ist möglich.

Die einem GroupTile-Objekt untergeordneten Objekte können über die Eigenschaft Grouping so angeordnet werden, dass sie den gesamten zur Verfügung stehenden Raum des Objekts einnehmen. Die untergeordneten Objekte werden bei Größenänderungen des GroupTile-Objekts automatisch angepasst.

GroupTile-Objekte besitzen (ähnlich zu <u>Frame</u>-Objekten) eine Titelleiste, über die sie verschoben, maximiert und geschlossen werden können. Wird ein GroupTile geschlossen, wird dessen <u>Visible</u>-Eigenschaft auf <u>false</u> gesetzt. Durch das Setzen dieser Eigenschaft auf true kann das Objekt wieder sichtbar gemacht werden.

Besitzt ein untergeordnetes Objekt eines GroupTile-Objekts den Tastaturfokus, wird die Titelleiste aktiviert dargestellt, und das GroupTile wird als aktiv bezeichnet.

Beim Setzen von Eigenschaften des GroupTile-Objekts ist je nach Eigenschaft folgendes zu beachten:

• Width und Height

Das GroupTile-Objekt darf nicht maximiert sein. Innerhalb des GroupSplit-Objekts müssen alle GroupTile-Objekte sichtbar sein. Der zugewiesene Wert darf nicht größer 10000 sein.

AreaId

Das GroupTile-Objekt darf nicht maximiert sein. Es darf kein geschlossenes GroupTile-Objekt existieren. Der übergebene Wert muss mindestens Null sein und darf die Anzahl der im GroupSplit-Objekt angezeigten GroupTile-Objekte nicht übersteigen.

• FlagsTitlebar

Das GroupTile-Objekt darf nicht maximiert sein. Die Optionen können ausschließlich in Kombination mit der Option <u>WinTitlebarCaption</u> gesetzt werden.

• Visible

Die Eigenschaft AutoUpdate des GroupSplit-Objekts muss auf true gesetzt sein. Ein maximiertes GroupTile-Objekt kann nicht unsichtbar gesetzt werden. Die Eigenschaft Areald muss einen gültigen Wert besitzen.

• AreaWidthMin, AreaWidthMax, AreaHeightMin und AreaHeightMax

Der übergebene Wert muss >= -1 betragen.

Scrollbox

Scrollbares Containerobjekt

Liste, Eigenschaften,

Siehe <u>Ereignisse</u>,

WinBoxScrollVisible()

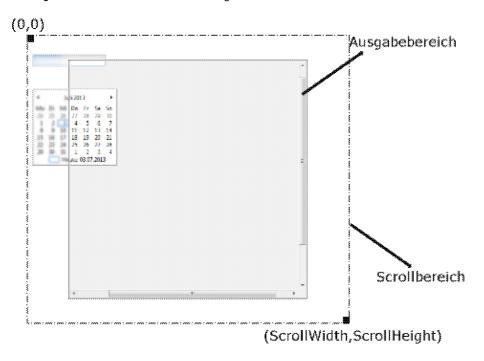
Das Scrollbox-Objekt ist ein Container-Objekt (ähnlich der <u>Groupbox</u>) zur Aufnahme von untergeordneten Objekten. Das Scrollbox-Objekt besitzt einen Scrollbereich, dessen Größe durch die Eigenschaften <u>ScrollWidth</u> und <u>ScrollHeight</u> definiert wird. Ist der Scrollbereich größer als der Ausgabebereich der Scrollbox (definiert durch die Objektgröße), dann werden Scrollbars angezeigt, mit denen der Scrollbereich verschoben werden kann. Über die Eigenschaften <u>ScrollLeft</u> und <u>ScrollTop</u> kann eine vorgegebene Position des Scrollbereichs eingestellt werden. Die der Scrollbox untergeordneten Objekte werden an den Grenzen des Scrollbereichs geclippt.

Die Eigenschaften <u>ScrollLeft</u> und <u>ScrollTop</u> können auch gesetzt werden, wenn das Scrolling durch die Eigenschaft <u>SbarStyle</u> unterbunden wurde.

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

Die Scrollgeschwindigkeit bei Verwendung des Mausrads berücksichtigt die Windows-Einstellung in der Systemsteuerung (Mauseinstellungen).

Beispiel für ein Scrollbox-Objekt:



Divider

Trennlinie

<u>Liste</u>,

Siehe <u>Eigenschaften</u>,

Ereignisse

Das Divider-Objekt kann dazu verwendet werden, Ausgabebereiche zu unterteilen und somit übersichtlicher zu gestalten.

COM-Objekte Objekte in der COM-Palette Siehe <u>Alle</u> <u>Oberflächen-Objekte</u>

CtxOffice

X<mark>W</mark>

CtxAdobeReader |

CtxDocEdit



CtxOffice

Kontroll-Objekt für Microsoft Office-Dokumente

Liste,

Eigenschaften,

Siehe Ereignisse,

InstallCtxOffice,

COM-Befehle

Über dieses Objekt kann in einem Dialog ein Microsoft Office-Dokument oder eine Karte von Microsoft MapPoint angezeigt und über die <u>COM-Schnittstelle</u> gesteuert werden. Damit ein Dokument angezeigt werden kann, muss die Office-Erweiterung über die Eigenschaft <u>InstallCtxOffice</u> installiert sein.

Ist bei der Initialisierung des Objekts ein Fehler aufgetreten, kann dieser mit der Anweisung <u>ComInfo()</u> ermittelt werden.

In dem Objekt kann in der Eigenschaft <u>FileName</u> ein bestehendes Dokument angegeben oder ein neues Dokument angelegt werden.

Der Deskriptor des Objekts kann dazu verwendet werden, um Methoden des COM-Objekts aufzurufen (siehe <u>ComCall()</u>) oder Eigenschaften der Oberfläche und des COM-Objekts zu setzen bzw. abzufragen.

Beispiel:

 $CtxOffice->wpFileName \# '*' + _Sys->spPathMyDocuments + '\MyDocument.doc';// Set document to real specifies of the supplement of the sup$

CtxAdobeReader



Kontroll-Objekt des Adobe Readers

Liste,

 $Siehe \frac{Eigenschaften}{Ereignisse},$

COM-Befehle

Über diese Objekt kann der Adobe Reader gesteuert werden. Der Deskriptor des Objekts kann dazu verwendet werden, um Methoden des COM-Objekts aufzurufen (siehe ComCall()) oder Eigenschaften der Oberfläche und des COM-Objekts zu setzen bzw. abzufragen.

Ist bei der Initialisierung des Objekts ein Fehler aufgetreten, kann dieser mit der Anweisung ComInfo() ermittelt werden.



Damit dieses Objekt die PDF-Datei anzeigen kann, muss Adobe Reader installiert sein.

Beispiel:

\$CtxAdobeReader->ComCall('setLayoutMode', 'SinglePage');\$CtxAdobeReader->wpFileName # '*' + _Sys-

CtxDocEdit

🕍 Objekt zu Textverarbeitung

Liste,

Eigenschaften,

Siehe $\frac{\text{Ereignisse}}{\text{InstallCtxDocEdit}}$,

CtxDocEdit-Befehle,

COM-Befehle, Blog

Über dieses Objekt können Texte verarbeitet werden. Das Objekt kann über die COM-Schnittstelle gesteuert werden. Damit ein Dokument angezeigt werden kann, müssen die Gemeinsamen Komponenten auf dem Rechner installiert und die DocEdit-Erweiterung über die Eigenschaft InstallCtxDocEdit eingerichtet sein.

Ist bei der Initialisierung des Objekts ein Fehler aufgetreten, kann dieser mit der Anweisung ComInfo() ermittelt werden.

In dem Objekt kann in der Eigenschaft FileName ein bestehendes Dokument angegeben werden. Standardmäßig wird ein leeres Dokument beim Start angezeigt. Mit dem COM-Befehl \$ctxDocEdit->cplResetContents kann der aktuelle Inhalt geleert werden. Jedoch bleibt die Eigenschaft FileName bei dem Befehl gesetzt.

Der Deskriptor des Objekts kann dazu verwendet werden, um Methoden des COM-Objekts aufzurufen (siehe ComCall()) oder Eigenschaften der Oberfläche und des COM-Objekts zu setzen bzw. abzufragen.



Eine Dokumentation der möglichen Befehle, Eigenschaften und Ereignissen ist auf der Hersteller-Seite des Moduls zu finden.

Beispiel:

// Dokument laden\$CtxDocEdit->wpFileName # '*' + Sys->spPathMyDocuments + '\MyDocument.doc';// [

Weitere Objekte Siehe <u>Alle</u> <u>Oberflächen-Objekte</u>

<u>Hyperlink</u>



<u>Calendar</u>



<u>Progress</u>



<u>GanttGraph</u>



RecNavigator

<u>Slider</u>



HyperLink HyperLink

Liste,

Siehe Eigenschaften,

Ereignisse

Ein Hyperlink ist ein Bezeichner, hinter dem eine Aktion liegt. Die Aktion wird ausgefürt, wenn der Benutzer auf das Objekt klickt. Auf diese Weise kann zum Beispiel der Web-Browser gestartet und eine bestimmte Seite aufgerufen werden.

Die auszuführende Aktion ist in der Eigenschaft LinkText anzugeben.

Calendar



Kalender

Liste,

Siehe Eigenschaften,

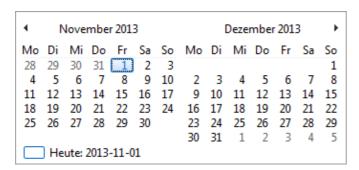
Ereignisse

Im Kalender-Objekt wird ein vollständiger Monat angezeigt. Durch Anklicken des Monats in der Titelzeile des Objekts kann der Monat und durch Anklicken der Jahresangabe das Jahr ausgewählt werden.

Unter Windows Vista ist das Calendar-Objekt betriebssystemseitig funktionell erweitert worden. Durch Anklicken des Titelbereichs wird die Anzeige innerhalb der Calendar-Seite verändert. Nach dem ersten Klicken werden die Monate, anschließend die Jahre, dann Jahresbereiche angezeigt. Durch Doppelklicken eines Monats / Jahres oder Jahresbereich wird die vorhergehende Anzeige wieder hergestellt.

Die Größe des Objektes kann bei verschiedenen Betriebssystemen unterschiedlich ausfallen. Damit das Objekt unter jeder Plattform angezeigt wird, ohne das Bereiche des Objekts abgeschnitten werden, kann die Eigenschaft <u>AutoSize</u> gesetzt werden. Die Größe des Objektes wird dann so angepasst, dass ein Monat angezeigt werden kann.

Das Objekt kann manuell so vergrößert werden, dass mehrere Monate dargestellt werden. Dies sieht beispielsweise wie folgt aus:



Über die Eigenschaft <u>StyleDisplay</u> wird festgelegt, ob im Kalender die Kalenderwochen angezeigt werden.

Das Datum kann mit der Eigenschaft CaptionDate gelesen und gesetzt werden.

Durch die Eigenschaft <u>StyleTheme</u> wird beeinflusst, ob das Objekt in Theme-Darstellung <u>WinStyleThemeSystem</u> oder in den angegebenen Farben <u>WinStyleThemeNone</u> gezeichnet wird. Die Eigenschaft wirkt sich bei dem Objekt nur aus, wenn die gleiche Eigenschaft beim <u>Application-Objekt</u> auf <u>WinStyleThemeSystem</u> gesetzt ist.

Ist die Eigenschaft <u>TabStop</u> auf <u>true</u> gesetzt, kann das Datum mit Hilfe der folgenden Tastaturkommandos relativ zum eingetragenen Datum verändert werden:

Tastenkombination Veränderung relativ zum aktuellen Datum



Gleicher Wochentag der vorherigen Woche



Gleicher Wochentag der nächsten Woche

Vorheriger Tag
Nächster Tag
Gleicher Tag des Vorherigen Monats
Gleicher Tag des nächsten Monats
Gleiches Datum des vorhergehenden Jahres
Gleiches Datum des nächsten Jahres
1. Tag des Monats des aktuellen Datums
Letzter Tag des Monats des aktuellen Datums

Ist das Datum der 29.2. eines Schaltjahres und wird und bild oder gedrückt, wird auf den 28.2. des vorhergehenden bzw. nachfolgenden Jahres positioniert.

Die Tasten , , und können nur verwendet werden, wenn die Eigenschaft <u>EmulateKeys</u> des übergeordneten Frame-Objekts auf <u>false</u> gesetzt ist. und haben dann die gleiche Wirkung wie <u>strg</u> + bzw. <u>strg</u> + .

Progress

Fortschrittsanzeige

Liste,

Eigenschaften,

Siehe Ereignisse,

Universelles

Modul (Blog)

Mit diesem Objekt kann der Fortschritt einer Operation angezeigt werden.

Der Fortschritt des Balkens kann mit den Eigenschaften <u>ProgressPos</u> und <u>ProgressMax</u> gesetzt werden.

GanttGraph

Gantt-Diagramm

Liste,

Eigenschaften,

Ereignisse,

Befehle,

PrtGanttGraph,

Siehe <u>Darstellung</u>

<u>von</u>

<u>Kennzahlen</u>

(Blog),

Verwendung

(Blog)

Mit diesem Objekt lassen sich zeitliche Abläufe darstellen. Das Gantt-Diagramm besteht aus mindestens zwei Achsen (eine horizontale und eine vertikale Achse) sowie einer Anzahl von Zellen und Intervallen.

Die Achsen werden mit dem GanttGraph-Editor erstellt und verändert.

Das Anlegen und Entfernen der Zeitintervalle erfolgt über die Befehle WinGanttIvlAdd() und WinGanttIvlRemove().

Axis

x- und y-Achse

Siehe $\frac{\text{Liste}}{\text{Eigenschaften}}$

Ein <u>GanttGraph</u> besteht aus einer oder mehrerer x- und y-Achsen. Die Achsen werden mit dem <u>GanttGraph-Editor</u> erstellt und verändert.

Box

Box-Objekt

Siehe <u>Liste</u>, <u>Eigenschaften</u>

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

Das Anlegen eines Box-Objekts geschieht durch den Befehl WinGanttBoxAdd().

Interval

Intervall-Objekt

Siehe $\frac{\text{Liste}}{\text{Eigenschaften}}$

Ein GanttGraph besteht aus einem oder mehreren Intervall-Objekten. Das Anlegen und Entfernen der Intervalle erfolgt über die Befehle WinGanttIvlAdd() und WinGanttIvlRemove().

Line

Line-Objekt

Siehe <u>Liste,</u> <u>Eigenschaften</u>

Bei einem Line-Objekt handelt es sich um eine Hilfslinie innerhalb eines Gantt-Diagramms.

Das Anlegen eines Line-Objekts geschieht durch den Befehl WinGanttLineAdd().

View

View-Objekt

Siehe Liste,

<u> Eigenschaften</u>

Ein <u>GanttGraph</u> besitzt mindestens ein View-Objekt. In dieser Ansicht werden die Intervalle des GanttGraphen dargestellt.

Ist die Eigenschaft <u>SplitStyle</u> nicht auf <u>WinSplitNone</u> gesetzt, kann über die "Splitter" (über bzw. links von den Scrollbars) die Ansicht geteilt werden. Auf diese Weise entstehen weitere View-Objekte (maximal vier). Der Deskriptor eines View-Objektes kann mit dem Befehl <u>WinInfo()</u> ermittelt werden.

tHdlView # <0bj>->WinInfo(_WinObject, <View-Nummer>, _WinTypeGanttView);

In <Obj> wird der Deskriptor des GanttGraphen und in <View-Nummer> die Nummer der Ansicht übergeben. Ist die Ansicht nicht geteilt, existiert nur die Ansicht mit der Nummer 1. Alle weiteren Nummern werden von links nach rechts und von oben nach unten vergeben. Ist der <u>GanttGraph</u> horizontal in zwei Ansichten aufgeteilt, hat die obere Ansicht die Nummer 1, während die untere Ansicht die Nummer 2 besitzt.

Der ermittelte Deskriptor ist solange gültig, wie diese Views existieren. Der Benutzer kann durch Verschieben der "Splitter" die Ansicht so verändern, dass nur noch ein View vorhanden ist. Alle anderen Deskriptoren sind dann nicht mehr gültig.

RecNavigator

Schaltflächenleiste für Datensatzoperationen

Liste

Siehe Eigenschaften,

Ereignisse

Dieses Objekt beinhaltet alle Standard-Datensatzoperationen. Die Datensatzoperationen beziehen sich auf die Datei, die in der Eigenschaft <u>DbFileNo</u> angegeben wurde.

Die Schaltflächen haben folgende Bedeutung:

- Vorigen oder nächsten Datensatz lesen
- Liest fünf Datensätze nach vorne oder hinten
- Ersten oder letzten Datensatz lesen
- Neuen Datensatz anlegen
- Bestehenden Datensatz löschen
- Gesperrten Datensatz zurückschreiben
- Datensatz sperren

Die Eigenschaft <u>StyleConsole</u> entscheidet über den Funktionsumfang des RecNavigators.

Wird über den RecNavigator ein neuer Datensatz gelesen, wird nach dem Lesen das Ereignis <u>EvtDbFldUpdate</u> beim Frame-Objekt (<u>Frame</u> oder <u>MdiFrame</u>) ausgelöst. Über das Ereignis kann in einem Frame in Abhängigkeit des gelesenen Datensatzes zusätzliche Informationen zum Beispiel aus anderen Datensätzen oder externen Dateien zur Verfügung gestellt werden.

Über das Ereignis <u>EvtMouseItem</u> kann ermittelt werden, welche Schaltfläche des Objekts angeklickt wurde. Bei Leseoperationen wird das Ereignis nach der Datensatzoperation und vor der Übertragung der Felder in die Objekte ausgelöst. Der Rückgabewert des Ereignisses wird nicht ausgewertet. Bei Schreiboperationen wird das Ereignis vor der Datensatzoperation ausgelöst. Die Schreiboperation kann mit dem Rückgabewert <u>false</u> verhindert werden.

Das Ergebnis der letzten Datensatzoperation kann über die Eigenschaft $\underline{\text{Erg}}$ ermittelt werden.

Slider

Schieberegler

Liste,

Siehe Eigenschaften,

Ereignisse

Beim Slider-Objekt handelt es sich um ein Oberflächenobjekt zur Auswahl eines ganzzahligen Wertes aus einem vorgegebenen Bereich.

Das Slider-Objekt setzt sich zusammen aus dem Kanal, einem Schieberegler, Markierungen und einem Textfeld. Über die Eigenschaft <u>Vertical</u> kann die Ausrichtung des Slider definiert werden.

Der Kanal wird durch einen farbigen Balken dargestellt, über den der Schieberegler verschoben werden kann. Die einzelnen Bestandteile sind in folgendem Bild ersichtlich:



Die Werte des Kanals werden durch die Eigenschaften MinInt und MaxInt definiert. Die Eigenschaft CurrentInt definiert die aktuelle Position des Schiebereglsers im Kanal. Steht der Schieberegler ganz links (bzw. oben) enthält CurrentInt den Wert von MinInt. Wird der Schieberegler nun nach rechts (bzw. unten) verschoben, erhöhen sich die Werte für CurrentInt, solange bis das rechte (bzw. untere) Ende des Kanals erreicht ist. Dort enthält die Eigenschaft CurrentInt den Wert von MaxInt. CurrentInt enthält also immer einen Wert im Bereich von MinInt bis MaxInt in Abhängigkeit von der Position des Schiebereglers im Kanal.

Standardmäßig erhöht sich <u>CurrentInt</u> immer um eins, wenn der Schieberegler um eine Position nach rechts bzw. unten verschoben wird. Dies kann durch die Eigenschaft <u>TickInterval</u> geändert werden. Wird dieser ein Wert von zwei zugewiesen, erhöht sich <u>CurrentInt</u> entsprechend um zwei, wenn der Schieberegler eine Position nach rechts bzw. unten verschoben wird.

Bei der Bedienung des Sliders mit der Tastatur haben die Eigenschaften <u>TickLineSize</u> und <u>TickPageSize</u> eine besondere Rolle. Sie geben die Anzahl der Positionen an, um die der Schieberegler bewegt wird, wenn die Pfeiltasten bzw. die betätigt werden. Dabei kann der Schieberegler nur Werte annehmen, die im <u>TickInterval</u> liegen. Sind die Eigenschaften <u>TickLineSize</u> bzw. <u>TickPageSize</u> kleiner als <u>TickInterval</u>, wird <u>CurrentInt</u> dennoch um den Wert aus <u>TickInterval</u> verändert.

Die Ausrichtung der Markierungen kann über die Eigenschaft <u>TickAlignment</u> gesteuert werden. Standardmäßig werden keine Markierungen angezeigt.

Jedem Wert aus dem Wertebereich [<u>MinInt</u>, <u>MaxInt</u>] kann ein Text zugeordnet werden, der als <u>HelpTip</u> und/oder im Textfeld des Sliders angezeigt wird. Die Texte werden hierbei durch das Zeichen '|' in der Eigenschaft <u>HelpTip</u> eingetragen. Sind weniger Texte vorhanden als Werte im Wertebereich, wiederholen sich die Texte

entsprechend. Das Textfeld des Sliders wird über die Eigenschaft <u>Caption</u> definiert. Hierbei ist die Verwendung von Escape-Sequenzen erlaubt. '\$CUR' stellt den Text für die aktuelle Position im Textfeld dar. Daneben gibt es noch '\$MIN' und '\$MAX' für den Text zu <u>MinInt</u> und <u>MaxInt</u>. Die Ausrichtung des Textes im Textfeld geschieht über die Eigenschaften <u>Justify</u> und <u>JustifyVert</u>.

Die Eigenschaft <u>ShowFocus</u> definiert, ob ein Fokusrechteck dargestellt werden soll, wenn der Slider den Eingabefokus erhält. In der systemabhängigen Darstellung hat die Eigenschaft keine Auswirkung. Dort wird immer ein Rechteck dargestellt. Nur in der <u>Modern Theme Style</u>-Darstellung wird die Eigenschaft berücksichtigt.

Eigenschaften eines Slider-Objekts

Das Objekt Slider unterstützt folgende Eigenschaften

Slider, Liste

Siehe der Objekte,

Liste der

Eigenschaften

- AlignGrouping
- AlignHeight
- AlignMarginBottom
- AlignMarginLeft
- AlignMarginRight
- AlignMarginTop
- AlignWidth
- Area
- AreaBottom
- AreaHeight
- AreaLeft
- AreaRight
- AreaTop
- AreaWidth
- Caption
- CurrentInt
- Custom
- Disabled
- Font
- FontParent
- Help
- HelpTip
- HelpTipIustify
- HelpTipSysFont
- <u>HelpTipTimeDelay</u>
- HelpTipTimeShow
- <u>Iustify</u>
- JustifvVert
- MaxInt
- MenuNameCntxt
- MinInt
- Name
- ShowFocus
- StyleBorder
- StyleTheme
- TabPos
- TabStop
- ThemeMenuCntxtTileSize
- ThemeSetId
- TickAlignment
- TickInterval
- <u>TickLineSize</u>
- TickPageSize
- Vertical

• Visible

Ereignisse eines Slider-Objekts

Das Objekt Slider unterstützt folgende Ereignisse.

<u>Slider</u>,

<u>Liste der</u>

Siehe Objekte,

<u>Liste der</u>

Ereignisse

- EvtChanged
- EvtHelpTip
- EvtMenuCommand
- <u>EvtMenuContext</u>
- EvtMenuInitPopup
- EvtMouse

Canvas

Anzeige und Interaktion vektorbasierter Grafikobjekte

Liste, Eigenschaften,

Siehe Ereignisse,

CodeLibrary-Beispiel

"Canvas"

Das Canvas-Objekt dient zur Anzeige und Interaktion vektorbasierter Grafikobjekte.

Das Canvas-Objekt ist ein Container, der Grafikobjekte verwaltet. Ein Grafikobjekt kann mittels WinCreate(_WinTypeCanvasGraphic, ...) erstellt und gleichzeitig oder nachträglich mit WinAed(")) zum Canvas-Objekt hinzugefügt und per WinRemove(")) bzw. <a href="WinDestroy(") wieder entfernt werden. Der von <a href="WinCreate(") zurückgegebene Deskriptor kann verwendet werden, um die Eigenschaften des Grafikobjektes zu setzen oder zu lesen. Die Form der erzeugten Grafikobjekte kann über die Eigenschaft FormType des jeweiligen Objektes gesetzt werden.

Die Ausgabe der Grafikobjekte findet innerhalb der Ausgabefläche des Canvas-Objektes statt. Liegen Grafikobjekte außerhalb des sichtbaren Bereiches, zeigt das Canvas-Objekt Scrollbalken an, um die Grafikobjekte in den sichtbaren Bereich verschieben zu können. Die Zeichenfläche wird in geräteunabhängigen, logischen Einheiten bereitgestellt. Für die Umrechnung von Einheiten (z. B. Millimeter, Zentimeter) in logische Einheiten kann der Befehl PrtUnitLog() verwendet werden.

Standardmäßig zeigt das Canvas-Objekt ein Raster an. Dieses kann über die Eigenschaften <u>GridWidth</u> und <u>GridHeight</u> ausgeblendet (einer der Werte = 0) oder in der Größe in logischen Einheiten definiert werden. Zudem werden am oberen und linken Rand standardmäßig Liniale angezeigt. Diese können mit der Eigenschaft <u>Ruler</u> ausgeblendet werden.

Beispiel:

// Canvas-Objekt erstellen und Objektgröße setzentCanvas # WinCreate(WinTypeCanvas, 'cnvCanvas'

Benutzerinteraktion

Das Canvas-Objekt unterstützt das Selektieren von <u>CanvasGraphic</u>-Objekten. Hierfür stehen mehrere Möglichkeiten zur Verfügung:

Auswahl mit der Maus

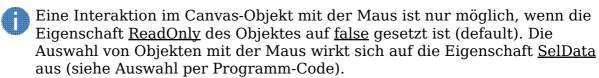
Eine Auswahl kann aufgespannt werden, indem mit der linken Maustaste () in einen freien Bereich des Canvas-Objektes geklickt und die Maustaste gehalten wird. Beim Bewegen der Maus wird ein rechteckiger Bereich aufgespannt. Wird die linke Maustaste losgelassen, werden alle <u>CanvasGraphic</u>-Objekte selektiert, die sich vollständig innerhalb des rechteckigen Bereiches befinden.

Ein nicht ausgewähltes <u>CanvasGraphic</u>-Objekt kann der Auswahl hinzugefügt werden, indem es mit angeklickt wird. Auf diese Weise kann das Objekt auch wieder aus der Auswahl entfernt werden. Ein Mausklick ohne Zusatztaste auf ein nicht ausgewähltes CanvasGraphic-Objekt entfernt die

bereits vorhandene Auswahl und wählt das geklickte Objekt aus.

Ausgewählte <u>CanvasGraphic</u>-Objekte werden mit einem Selektionsrahmen mit acht Ankerpunkten dargestellt. Wird die Maus über einen Ankerpunkt geführt, verändert sich der Mauszeiger entsprechend. Klickt der Benutzer mit auf einen Ankerpunkt und hält diese gedrückt, führen Bewegungen mit der Maus zu einer Größenänderung aller ausgewählten <u>CanvasGraphic</u>-Objekte. Die ausgewählten Objekte können auch verschoben werden. Dies geschieht, indem der Mauszeiger in ein bereits ausgewähltes Objekt geführt wird. Jetzt verändert sich der Mauszeiger und zeigt ein Verschieben-Symbol an. Wird nun die gedrückt und gehalten, werden alle ausgewählten Objekte verschoben.

Alle Operationen (Auswahl aufspannen, Größe ändern, Verschieben) können bei gehaltener linker Maustaste abgebrochen werden, indem die betätigt wird.



• Auswahl per Programm-Code

Die Mehrfachauwahl kann prozedural über die Eigenschaft <u>SelData</u> verändert werden. Die Eigenschaft enthält alle gegenwärtig im Canvas-Objekt ausgewählten <u>CanvasGraphic</u>-Objekte. Werden Objekte vom Anwender über Maus selektiert oder deselektiert, wird automatisch die Eigenschaft <u>SelData</u> aktualisiert. Zum prozedurealen Hinzufügen und Entfernen von <u>CanvasGraphic</u>-Objekten wird der Deskriptor des <u>CanvasGraphic</u>-Objektes bei <u>WinMsdInsert()</u> bzw. <u>WinMsdDelete()</u> angegeben. Zur Aktualisierung der Mehrfachauswahl muss anschließend <u>WinUpdate(_WinUpdState)</u> durchgeführt werden.

CanvasGraphic

Grafikobjekt zur Darstellung in einem <u>Canvas</u>-Objekt

Liste, Eigenschaften,

Siehe CodeLibrary-Beispiel

"Canvas"

Grafikobjekte stellen den Inhalt eines <u>Canvas</u>-Objektes dar. Sie können mit dem Befehl <u>WinCreate(WinTypeCanvasGraphic,</u> ...) angelegt werden.

Über die Eigenschaft <u>FormType</u> wird die Form des Grafikobjektes definiert. Die Position und Größe wird mit der Eigenschaft <u>Area</u> in logischen Einheiten definiert. Für die Umrechnung von Einheiten (z. B. Millimeter, Zentimeter) in logische Einheiten kann der Befehl <u>PrtUnitLog()</u> verwendet werden.

Farbgebung

Die Eigenschaft <u>ColBkg</u> definiert die Füllfarbe des Grafikobjektes. Eine transparente Darstellung kann über <u>WinColorOpacitySet()</u> erreicht werden. Die Grafikobjekte können mit einem Rahmen versehen werden. Die Stärke des Rahmens in logischen Einheiten, wird über die Eigenschaft <u>BorderWidth</u> definiert. Die Füllfarbe des Rahmens wird über die Eigenschaft <u>BorderColFg</u> definiert. Auch hier ist die Angabe einer teiltransparenten Farbe durch <u>WinColorOpacitySet()</u> möglich.



Die Füll- (<u>ColBkg</u>) und Rahmenfarbe (<u>BorderColFg</u>) werden beim <u>FormType WinFormTypeLine</u> ignoriert. Text- und Linien-Farbe werden über die Eigenschaft <u>ColFg</u> definiert. Hierbei werden teiltransparente Farben **nicht** berücksichtigt.

Inhalt

Als Inhalt können sowohl Texte als auch Grafiken dargestellt werden.

Der anzuzeigende Text wird über die Eigenschaft <u>Caption</u> definiert. Dieser kann mehrzeilig sein. Wortumbrüche können über die Eigenschaft <u>WordBreak</u> aktiviert werden. Die Eigenschaft <u>Vertical</u> gibt an, ob der Text vertikal oder horizontal (default) angezeigt wird. Die Textausrichtung geschieht über die Eigenschaften <u>Justify</u> und <u>JustifyVert</u>.

Ein Bild kann über die Eigenschaft <u>PictureName</u> gesetzt werden. Ein Stern ('*') als Präfix kennzeichnet ein extern vorliegendes Bild. Die Anzeige des Bildes erfolgt über die Eigenschaft <u>PictureMode</u>. Die Eigenschaft <u>Vertical</u> definiert, ob das Bild gekippt dargestellt wird oder nicht.

Besonderheiten je nach <u>FormType</u>

<u>WinFormTypeRectangle</u> / <u>WinFormTypeSquare</u>:

Bei der Ausgabe des rechteckigen Bereiches (<u>WinFormTypeRectangle</u>, <u>WinFormTypeSquare</u>) können abgerundete Ecken über die Eigenschaft <u>Justify</u> und <u>Radius</u> gesetzt werden. Die Eigenschaft definiert den Radius der abgerundeten Ecken in logischen Einheiten.

<u>WinFormTypeLine</u>:

Bei der Linie werden Füll- (<u>ColBkg</u>) und Rahmenfarbe (<u>BorderColFg</u>) ignoriert. Die Farbe der Linie wird über die Eigenschaft <u>ColFg</u> gesetzt. Die Linienstärke in logischen Einheiten definiert die Eigenschaft <u>LineWidth</u>. Die Linie kann mit einem Pfeiltyp für den Start- bzw. Endpunkt versehen werden. Hierfür existiert die Eigenschaften <u>LineCapStart</u> und <u>LineCapEnd</u>.

Die Ausrichtung der Linie kann horizontal ($\underline{\text{Vertical}} = \underline{\text{false}}$) oder vertikal ($\underline{\text{Vertical}} = \underline{\text{true}}$) sein. Die Ausrichtung der Linie (links, zentriert, mittig) geschieht über die Eigenschaft $\underline{\text{LineJustify}}$. Bild und Text werden in Abhängigkeit von der Ausrichtung links / über bzw. rechts / unter der Linie angezeigt.

Eigenschaften eines CanvasGraphic-Objekts

Das <u>CanvasGraphic</u>-Objekt unterstützt folgende Eigenschaften.

CanvasGraphic,

Liste der

Siehe Objekte, Liste

<u>der</u>

Eigenschaften

- Area
- AreaBottom
- AreaHeight
- AreaLeft
- AreaRight
- AreaTop
- AreaWidth
- BorderColFq
- BorderWidth
- Caption
- ColBka
- ColFa
- Custom
- Font
- FontParent
- FormType
- HelpTip
- <u>Iustify</u>
- <u>IustifyVert</u>
- <u>LineCapEnd</u>
- LineCapStart
- LineJustify
- <u>LineWidth</u>
- ModeEffect
- Name
- <u>PictureMode</u>
- <u>PictureName</u>
- Radius
- Rotation
- Vertical
- WordBreak

Eigenschaften eines Canvas-Objekts

Das Objekt <u>Canvas</u> unterstützt folgende Eigenschaften.

Canvas, Liste

Siehe <u>der Objekte</u>, <u>Liste der</u>

Eigenschaften

- AlignGrouping
- AlignHeight
- AlignMarginBottom
- AlignMarginLeft
- AlignMarginRight
- <u>AlignMarginTop</u>
- AlignWidth
- AutoUpdate
- ColBkq
- ColBkqApp
- ColBoundary
- ColFa
- Custom
- Font
- FontParent
- FormHeight
- FormWidth
- GridHeight
- GridWidth
- Name
- ReadOnly
- Ruler
- SelData
- <u>Unicode</u>
- Unit
- ZoomFactor

Ereignisse eines Canvas-Objekts

Das Objekt Canvas unterstützt folgende Ereignisse.

Canvas,

<u>Liste der</u>

Siehe Objekte,

Liste der

Ereignisse

- EvtFocusInit
- EvtFocusTerm
- EvtHelpTip
- EvtMenuCommand
- EvtMenuContext
- EvtMenuInitPopup
- <u>EvtMouseItem</u>

Druckformular-Objekte

Ein Druckformular besteht aus ein oder mehreren Objekte.

 $Siehe \frac{Alle}{Druckformular-Objekte}$

Die Objekte untergliedern sich in folgende Bereiche:

- <u>Druck-Objekte</u>
- <u>Drucker-Objekte</u>
- Seiten-Objekte
- Tabellen-Objekte
- Formular-Objekte

Die maximale Größe der Objekte beträgt 44x44 inch (ca. 112 cm).

Formular-Objekte

Siehe Alle Druckformular-Objekte

 $\underline{PrintDoc}$

PrintDocRecord



<u>PrintForm</u>



 $\underline{PrintFormList}$



PrintForm



PrintForm

Liste,

Eigenschaften,

Siehe Beispiel,

PrintFormList,

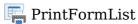
Blog

Das PrintForm-Objekt dient dem dynamischen Aufbau von Druckdokumenten (PrintDoc-Objekt). Hierzu gibt es spezielle Prozedurbefehle, die die im PrintForm-Objekt enthaltenen Druckobjekte einem PrintDoc-Objekt hinzufügen.

Das PrintForm-Objekt wird dann verwendet, wenn ein Druckjob aus kleineren oder sehr unterschiedlichen Elementen aufgebaut werden soll. Zum Beispiel eine Rechnung oder eine Liste, die auf Daten aus mehreren Tabellen zugreift.

Wird ein PrintForm-Objekt im Designer erstellt, ist in der Objektpalette nur der Registerreiter "Druck" vorhanden, und nur die dort enthaltenen Objekte können dem Objekt hinzugefügt werden.

PrintFormList



Liste,

 $Siehe \frac{Eigenschaften}{PrintForm},$

Blog

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

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

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

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

PrtLine

PrtLine

Liste,

Siehe Eigenschaften,

PrintFormList

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

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

Die Kriterien für die Sichtbarkeit des PrtLine-Objektes definiert die Eigenschaft SkipPrint.

PrtLineBorder

PrtLineBorder

Liste,

Siehe <u>Eigenschaften</u>,

PrintFormList

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

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

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

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

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

PrintDoc



PrintDoc

Liste,

Siehe $\frac{Eigenschaften}{PrtInfo()}$,

Blog

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

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

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

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

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

PrintDocRecord

PrintDocRecord

Liste,

Siehe Eigenschaften,

Bloq

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

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

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

Die angegebene Anzahl von Spalten werden beim Erzeugen gleichmäßig über die gesamte Seitenbreite angeordnet. Die Breite der Spalten kann entweder mit der Maus oder über die Eigenschaft Width verändert werden. Wird beim Ändern der Breite einer Spalte mit der Maus die -Taste gedrückt, werden alle übereinander liegenden Spalten mit verändert.

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

Druck-Objekte Siehe Alle Druckformular-Objekte abo <u>PrtText</u> PrtTblCellText •••• <u>PrtRtf</u> **PrtPicture** PrtMetaPicture <u>PrtPdf</u> **PrtBorder** PrtGanttGraph 📜 **PrtBarcode** <u>PrtDivider</u> 强 **PrtGroupbox**

PrtText

obc

PrtText

Siehe $\frac{\text{Liste}}{\text{Eigenschaften}}$

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

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

Mit der Eigenschaft AutoSize wird festgelegt, ob sich die Objektgröße an die Textgröße anpasst oder nicht.

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

Ausschließlich bei Fließkomma-Feldern sind auch Eigenschaften der Gruppe Function vorhanden.

Die Ausgabe von berechneten Funktionswerten ist möglich, wenn die Eigenschaft StyleCaption entsprechend gesetzt ist.

Eine Ausrichtung in horizontaler und vertikaler Richtung kann über die Eigenschaften Justify und JustifyVert erfolgen.

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

```
PrtRtf
PrtRtf

Liste,
Eigenschaften,
Text und
Daten
mischen, Text
suchen und
erstzen,
RtfEdit, Blog
```

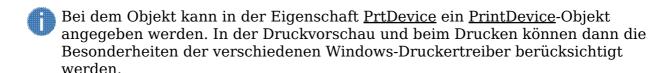
In diesem Objekt kann Text im Rich Text Format (RTF) ausgedruckt werden. Der formatierte Text kann aus unterschiedlichen Quellen stammen. Die Quelle wird in der Eigenschaft <u>StreamSource</u> angegeben. Je nach gewählter Quelle wird der Inhalt in der Eigenschaft <u>Caption</u>, <u>DbFieldName</u> oder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

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

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

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

Über die Funktion <u>PrtRtfSearch()</u> kann eine Textsuche- und Ersetzung vorgenommen werden.



Text und Daten mischen

Texte mit Datensätzen verbinden

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

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

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

• Beim Lesen eines Textes in ein RtfEdit-Objekt

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

• Beim Speichern eines Textes aus einem RtfEdit-Objekt

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

• Beim Lesen eines Textes in ein CtxDocEdit-Objekt

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

• Beim Speichern eines Textes aus einem CtxDocEdit-Objekt

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

• Beim Drucken eines Textes mit dem PrtRtf-Objekt

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

Folgende Markierungen werden beim Mischen von Text und Daten ersetzt:

• ~F:<Feldname>~

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

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

		Über diese Option wird die Anzahl der Nachkommastellen definiert. Soll die Anzahl der Nachkommastellen definiert werden, muss entweder zuvor mindestens einer der bereits genannten Formatoptionen oder ein "/" ohne Optionen angegeben werden. Werden keine Nachkommastellen angegeben, werden alle Nachkommastellen angezeigt.
Datum (<u>date</u>)	'/L'	Mit dieser Option wird das lange Datumsformat eingestellt. Wird kein langes Datumsformat gewählt, erscheint das Datum in der kurzen Anzeige.
Zeit (<u>time</u>)	'/S'	Die Uhrzeit wird mit Sekunden angezeigt.
	'/H'	Die Uhrzeit wird mit Sekunden und hundertstel Sekunden angezeigt.
Logisch (<u>logic</u>)	'/ <true>/<false>'</false></true>	Logische Felder und Variablen müssen mit Werten versehen werden, die in Abhängigkeit des Wertes angezeigt werden sollen. Die erste Zeichenkette wird angezeigt, wenn der Wert <u>true</u> ist, die zweite Zeichenkette wird angezeigt, wenn der Wert <u>false</u> ist.

• ~G:<Variablenbereich>:<Variablenname>~

Für Variablen gelten die gleichen Formatierungsmöglichkeiten, wie für Felder.

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

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

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

Beispiele:

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

PrtPdf

Objekt zum Drucken von PDF-Dokumenten

Liste,

Siehe Eigenschaften,

DocView

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

Der Pfad und der Name des Dokuments wird in der Eigenschaft FileName angegeben.

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

PrtTblCellText

PrtTblCellText

 $\frac{\text{Liste,}}{\text{Eigenschaften}}$ Das Zelltext-Objekt ist im wesentlichen identisch zum $\underline{\text{Text-Objekt}}$. Es kann jedoch nur in ein Zell-Objekt eingefügt werden.

PrtPicture



PrtPicture

Liste,

Siehe Eigenschaften,

Picture

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

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

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

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

Beispiele:

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

Mit der Eigenschaft Picture Mode können unterschiedliche Darstellungsweisen erfolgen.

PrtMetaPicture

🎮 PrtMetaPicture

 $\begin{array}{l} \text{Siehe} \stackrel{\underline{Liste}}{\underline{Eigenschaften}} \end{array}$

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

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

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

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

Beispiele:

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

Verschiedene Darstellungsweisen werden über die Eigenschaft <u>PictureMode</u> gesetzt.

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

Folgende Metafile-Formate werden unterstützt:

Windows Meta Files

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

Aldus Placeable Meta Files

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

Enhanced Meta Files

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

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

PrtDivider

Divider

 $Siehe \frac{Liste,}{Eigenschaften}$

Der PrtDivider zeichnet eine horizontale oder vertikale Trennlinie.

Die Richtung wird über die Eigenschaft Vertical gesetzt.

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

PrtBorder

Druckbare Umrandung

 $Siehe \frac{\underline{Liste},}{\underline{Eigenschaften}}$

Mit diesem Objekt können bestimmte Bereiche eines Ausdrucks hervorgehoben werden.

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

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

Wird das PrtBorder-Objekt eingefügt, wärend ein anderes Objekt selektiert ist, wird der Rahmen automatisch um dieses Objekt gezeichnet. Das Objekt wird in die Eigenschaft ObiLink eingetragen. Der Rahmen passt sich immer an die Größe des Objekts an.

PrtGroupbox

PrtGroupbox

 $Siehe \frac{Liste}{Eigenschaften}$

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

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

PrtBarcode

Barcode drucken

Liste,

Eigenschaften,

Siehe <u>Erzeugen von</u>

Barcodes (Blog)

Dieses Objekt stellt Barcodes oder Strichcodes dar. Der Barcode kann über die Eigenschaften <u>Caption</u> oder <u>DbFieldName</u> gesetzt werden. Der Typ des Barcodes wird in der Eigenschaft <u>TypeBarcode</u> oder als Präfix angegeben. Die Liste der unterstützten Barcodes befindet sich in der Eigenschaft <u>TypeBarcode</u>.

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

Beispiele:

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

PrtGanttGraph

Gantt-Diagramm

Liste,

Siehe $\frac{Eigenschaften}{Befehle}$,

GanttGraph

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

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

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

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

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

PrtGanttAxis

x- und y-Achse

Siehe $\frac{\text{Liste}}{\text{Eigenschaften}}$

Ein <u>PrtGanttGraph</u> besteht aus einer oder mehrerer x- und y-Achsen. Die Achsen werden mit dem PrtGanttGraph-Editor erstellt und verändert.

PrtInterval

PrtInterval-Objekt

Siehe $\frac{\text{Liste}}{\text{Eigenschaften}}$

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

PrtIvlBox

PrtIvlBox-Objekt

Siehe <u>Liste,</u> <u>Eigenschaften</u>

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

Das Anlegen eines Box-Objekts geschieht durch den Befehl WinGanttBoxAdd().

PrtIvlLine

PrtIvlLine-Objekt

Siehe <u>Liste</u>, <u>Eigenschaften</u>

Bei einem PrtLine-Objekt handelt es sich um eine Hilfslinie innerhalb eines PrtGanttGraph-Diagramms.

Das Anlegen eines Line-Objekts geschieht durch den Befehl WinGanttLineAdd().

Drucker-Objekte Siehe <u>Alle</u> <u>Druckformular-Objekte</u>

PrintDevice

<u>Printer</u>

PrinterList 🖷

<u>PrintJob</u>

PrintDevice
PrintDevice
Liste.

Siehe Eigenschaften,

PrtDeviceOpen()

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

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

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

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

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

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

Die Druckqualität kann über die Eigenschaften <u>QualityX</u> und <u>QualityY</u> gesetzt oder ermittelt werden.

Printer

Printer

Liste,

 $Siehe \frac{Eigenschaften}{Druckerattribute},$

Druckerstatus

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

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

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

PrinterList

PrinterList

Liste,

Siehe $\frac{\text{Eigenschaften}}{\text{PrtInfo()}}$,

PrtSearch()

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

PrintJob

PrintJob

Liste,
Eigenschaften,

Siehe PrtJobOpen(),
PrtPage,
Befehle,
Beispiel

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

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

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

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

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

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

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

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

Seiten-Objekte Siehe $\frac{\text{Alle}}{\text{Druckformular-Objekte}}$

<u>PrtPage</u>

<u>PrtPageTable</u>

<u>PrtPageHeader</u>

<u>PrtPageFooter</u>

PrtPage



PrtPage

Liste,

Siehe Eigenschaften,

Beispiel

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

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

<u>PrtPageHeader</u> und <u>PrtPageFooter</u> können über die Objektpalette in ein Seitenobjekt hinzugefügt werden, sofern die Objekte nicht schon in der Seite enthalten sind.

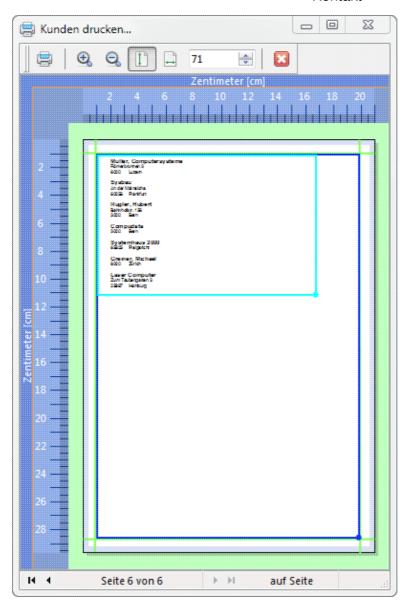
Der Seiteninhalt kann aus beliebigen Druckobjekten bestehen, die ebenfalls über die Objektpalette eingefügt werden.

Die Hintergrundfarbe der Seite kann über die Eigenschaft <u>ColBkgPage</u> gesetzt werden. Die Hintergrundfarbe für untergeordnete Objekte wird dagegen über die Eigenschaft <u>ColBkg</u> gesetzt.

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



Das Objekt hat standardmäßig einen Rand von einem Zentimeter. Dieser kann über die Eigenschaften <u>AreaMarginLeft</u>, <u>AreaMarginTop</u>, <u>AreaMarginRight</u> und <u>AreaMarginBottom</u> verändert werden.



PrtPageTable

PrtPageTable

Siehe $\frac{\text{Liste}}{\text{Eigenschaften}}$

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

Das PrtPageTable-Objekt kann ebenfalls <u>PrtPageHeader</u> und <u>PrtPageFooter</u> enthalten, jedoch kann der Seiteninhalt nur aus maximal einem Tabellen-Objekt bestehen.

Die Hintergrundfarbe der Seite kann über die Eigenschaft ColBkgPage gesetzt werden. Die Hintergrundfarbe für untergeordnete Objekte wird dagegen über die Eigenschaft ColBkg gesetzt.

PrtPageHeader

PrtPageHeader

 $Siehe \frac{Liste,}{Eigenschaften}$

Das PrtPageHeader-Objekt kann nur in ein <u>PrtPage</u>- oder <u>PrtPageTable</u>-Objekt eingefügt werden. Dem Objekt können beliebige Druckobjekte hinzugefügt werden.

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

Ist die Eigenschaft false, kann das PrtPageHeader-Objekt in der Größe verändert werden.

PrtPageFooter

PrtPageFooter

 $Siehe \frac{Liste,}{Eigenschaften}$

Das PrtPageFooter-Objekt kann nur in ein PrtPage- oder PrtPageTable-Objekt eingefügt werden. Dem Objekt können beliebige Druckobjekte hinzugefügt werden.

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

Ist die Eigenschaft false, kann das PrtPageFooter-Objekt in der Größe verändert werden.

Tabellen-Objekte
Siehe $\frac{Tabellen-Bereiche}{Druckformular-Objekte}$ PrtTblCommon
PrtTblAreaHdrTbl
PrtTblAreaHdrPage
PrtTblAreaHdrData
PrtTblAreaData
PrtTblAreaFtrData
PrtTblAreaFtrPage
PrtTblAreaFtrPage

PrtTblCommon

PrtTblCommon

Siehe <u>Liste</u>, <u>Eigenschaften</u>

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

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

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

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



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



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



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

PrtTblAreaData

PrtTblAreaData

Liste,

Siehe Eigenschaften,

Tabellen-Bereiche

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

Die Datensatzmenge wird im <u>PrintDocRecord-Formular</u> über die Eigenschaften DbFileNo, DbKevNo und DbLinkFileNo definiert.

PrtTblAreaHdrTbl

PrtTblAreaHdrTbl

Liste,

Siehe <u>Eigenschaften</u>, <u>Tabellen-Bereiche</u>

Die Ausgabe des Tabellenfuß-Bereiches erfolgt während des Druckens einmalig vor der Ausgabe aller anderen <u>Tabellen-Bereiche</u>.

PrtTblAreaFtrTbl

PrtTblAreaFtrTbl

<u>Liste</u>,

Siehe <u>Eigenschaften</u>, <u>Tabellen-Bereiche</u>

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

PrtTblAreaHdrPage

PrtTblAreaHdrPage

<u>Liste</u>

Siehe Eigenschaften,

Tabellen-Bereiche

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

PrtTblAreaFtrPage
PrtTblAreaFtrPage

Liste

Siehe Eigenschaften,

Tabellen-Bereiche

Die Ausgabe des Seitenfuß-Bereiches erfolgt während des Druckvorgangs auf jeder <u>Seite</u> nach der Ausgabe der <u>Datenbereiche</u> für die Seite und des <u>Datenfuß-Bereiches</u>. Im Designer kann maximal ein Seitenfuß-Bereich definiert werden.

PrtTblAreaHdrData

TblAreaHdrData

Liste,

Siehe <u>Eigenschaften</u>, <u>Tabellen-Bereiche</u>

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

PrtTblAreaFtrData

PrtTblAreaFtrData

<u>Liste</u>,

Siehe <u>Eigenschaften</u>, <u>Tabellen-Bereiche</u>

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

PrtTblCell

PrtTblCell

Siehe <u>Liste</u>, <u>Eigenschaften</u>

Das Zell-Objekt kann nur zu einem <u>Tabellen-Bereich</u> hinzugefügt werden. Über die Eigenschaft HeightMode kann festgelegt werden, ob und wie sich die Höhe der Zelle an den Zellinhalt anpasst.

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

System-Objekte Liste der System-Objekte

- Application
- Benutzer
- Benutzergruppe
- Binäre Objekte
- Chart-Objekt
- Dynamische Strukturen
- Elementgruppe
- HTTP-Objekt
- <u>Locale</u>
- Memory-Objekt
- PDF-Objekt
- System
- <u>TapiDevice</u>
- <u>Validierungselemente</u>
- Zentrale Datenobjekte

Locale

Länderspezifische Einstellungen

Siehe <u>Liste</u>, <u>Eigenschaften</u>

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

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

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

Die Eigenschaften des Objektes können mit dem Befehl SvsPropGet() ausgelesen und mit <u>SysPropSet()</u> gesetzt werden.

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

Das Objekt kann mit dem Befehl LocaleUnload() wieder entfernt werden.

System

Das System-Objekt

Siehe <u>Sys</u>, <u>Liste</u>, <u>Eigenschaften</u>

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

Der Deskriptor des System-Objektes kann durch das Schlüsselwort <u>Sys</u> ermittelt werden.

Benutzerverwaltung Vergabe und Verwaltung von benutzerabhängigen Rechten Siehe $\underline{\mathsf{Blog}}$

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

- Benutzer
- Benutzergruppen
- <u>Elementgruppen</u>
- Zusammenarbeit der Objekte
- Berechtigungen
- Verwaltung

Benutzer

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

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

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

Die Berechtigungen eines Benutzers ergeben sich aus seiner Zugehörigkeit zu einer oder mehreren <u>Benutzergruppen</u>, im Benutzer-Objekt selbst werden keine Rechte definiert.

Benutzergruppen

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

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

Benutzergruppen, zu denen er gehört.

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

Elementgruppen

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

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

In jeder Datenbank sind mehrere sogenannte Standard-Elementgruppen vordefiniert. In diesen Elementgruppen sind jeweils alle Elemente eines bestimmtem Typs enthalten. Zum Beispiel gehören alle Prozeduren automatisch zur Gruppe _procedure. Eine Liste aller Standard-Elementgruppen befindet sich im Abschnitt <u>Verwaltung der Elementgruppen</u>.

Zusammenarbeit der Objekte

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







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

Berechtigungen

In CONZEPT 16 werden folgende Berechtigungen verarbeitet:

Read (R)	<u>UrmPermRead</u>	Lesen und Anzeigen
Create (C)	<u>UrmPermCreate</u>	Anlegen
Modify (M)	<u>UrmPermModify</u>	Ändern
Modify Owner (MO)	<u>UrmPermModifyOwner</u>	Ändern bei Objektbesitz
Doloto (D)	I Imma Domma Doloto	Lässkan

Delete (D) <u>UrmPermDelete</u> Löschen

Delete Owner (DO) <u>UrmPermDeleteOwner</u> Löschen bei Objektbesitz

Execute (X) <u>UrmPermExecute</u> Ausführen
Config (CF) <u>UrmPermConfig</u> Konfiguration

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

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

Delete Element Berechtigung Element aus Gruppe (DE)

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

Insert Member UrmPermElmGroupRead Berechtigung Benutzer zu Benutzergruppe hinzufügen

Delete Member UrmPermElmGroupInsert Berechtigung Benutzer aus Benutzergruppe entfernen

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

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

Beispiel:

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

	_tables	ElmTblSales	ElmProcSales	ElmProcCom	
Visitor	M- D- CF-		C- M- D-	C- M- D-	
Sales		R+ C+ M+ D+	X+	X+	
Developer		R+ C+ M+ D+ X+ CF+	R+ C+ M+ D+ X+	R+ C+ M+ D+ X+	
Die Elementgruppe "_tables" wird von CONZEPT 16 zur Verfügung gestellt und enthält alle Tabellen der Datenbank. Die mit einem Minuszeichen versehenen Rechte werden der Benutzergruppe entzogen, die Rechte mit einem Pluszeichen werden					
erteilt.	7 201140201	gruppo omozogom, ano moc		oronom wordon	

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

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

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

Die effektiven Rechte des Besuchers sehen wie folgt aus:

_tables ElmTblSales ElmProcSales ElmProcCom

Effektiven Rechte M- D- CF- R+ C+ M- D- CF- C- M- D- X+ C- M- D- X+

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

_tables ElmTblSales ElmProcSales ElmProcCom

Effektiven Rechte M- D- CF- R+ C+ M- D- X+ CF- R+ C- M- D- X+ R+ C- M- D- X+ Verwaltung

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

- <u>Verwaltung der Benutzer</u>
- Verwaltung der Benutzergruppen
- Verwaltung der Elementgruppen

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

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

Über die Funktion <u>WinUrmDialog()</u> kann die Benutzerverwaltung in die eigene Anwendung integriert werden.

User

Benutzer-Objekt der Benutzerverwaltung

Liste.

Siehe Eigenschaften,

Benutzerbefehle

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

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

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

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

Beispiel

// Aktiviert den Benutzer Sales Temp für drei MonatetHdlUser # UrmOpen(UrmTypeUser, UrmLock, 'S

UserGroup

Objekt der Benutzergruppe in der Benutzerverwaltung

<u>Liste</u>

Siehe Eigenschaften,

Benutzerbefehle

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

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

Standardmäßig sind in der Benutzergruppe _Everyone alle vordefinierten Elementgruppen mit Ausnahme von _user und _usergroup mit den Rechten zum Erzeugen, Lesen, Ändern, Löschen, Ausführen und Konfigurieren eingetragen. Soll einer anderen Benutzergruppe die Ausführung aller Prozeduren erlaubt werden, muss der Benutzergruppe die Elementgruppe _procedure mit dem Recht <u>_UrmPermExecute</u> zugewiesen werden.

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

Die Eigenschaften der Benutzergruppe _Administrators können mit Ausnahme der Eigenschaft <u>UrmPropComment</u> nicht verändert werden. Insbesondere können die Berechtigungen der Gruppe nicht geändert werden.

Beispiel

// Benutzergruppe erzeugentErg # UrmCreate(0, _UrmTypeUserGroup, 'UserGrpLogin');if (tErg = _Err0

ElmGroup

Elementgruppe der Benutzerverwaltung

<u>Liste</u>

Siehe Eigenschaften,

Benutzerbefehle

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

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

<u>UrmTypeElmCustom</u> Benutzerdefiniert / virtuelles Objekt

<u>UrmTypeElmUser</u> Benutzer

<u>UrmTypeElmUsergroup</u> Benutzergruppe

<u>UrmTypeElmTable</u> Datei <u>UrmTypeElmDialog</u> Dialog <u>UrmTypeElmMenu</u> Menü

<u>UrmTypeElmPicture</u> Bild im Bereich <u>Picture</u>

<u>UrmTypeElmMetapicture</u> Bild im Bereich <u>MetaPicture</u>

<u>UrmTypeElmPrintform</u> <u>PrintForm-Objekt</u> <u>UrmTypeElmPrintformlist</u> <u>PrintFormList-Objekt</u>

<u>UrmTypeElmPrintdocument</u> <u>PrintDoc</u>-Objekt

<u>UrmTypeElmPrintdocrecord</u> <u>PrintDocRecord</u>-Objekt

<u>UrmTypeElmProcedure</u> Prozedur <u>UrmTypeElmText</u> interner Text

<u>UrmTypeElmBlob</u> <u>Binäre Objekte</u> in der <u>BLOb-Verwaltung</u>

<u>UrmTypeElmTheme</u> <u>Theme</u>-Objekt

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

Über den Typ <u>UrmTypeElmCustom</u> können Einträge durch den Programmierer vorgenommen werden: 'ModulArticle'. Die Rechte auf ein Element können in der Programmierung mit den Anweisungen <u>UrmPermElementGet()</u> oder <u>UrmPermElementGetRaw()</u> ermittelt werden.

Eine Elementgruppe wird mit der Anweisung <u>UrmCreate(0, _UrmTypeElmGroup, ...)</u> erzeugt. Eine vorhandene Elementgruppe kann mit der Anweisung <u>UrmOpen(0, _UrmTypeElmGroup, ...)</u> geöffnet werden. Beim Öffnen wird ein Deskriptor zurückgegeben, mit dem auf die <u>Eigenschaften</u> der Elementgruppe zugegriffen

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

Beispiel

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

Storage-Objekte

Zugriff auf Applikationsressourcen

Liste.

Siehe $\frac{\text{Eigenschaften}}{\text{Befehle für}}$

Storage-Objekte

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

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

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

• Dialog

Verzeichnis der Dialog-Objekte

• Menu

Verzeichnis der Menü-Objekte

• PrintForm

Verzeichnis der PrintForms

• PrintFormList

Verzeichnis der Drucklisten

• PrintDocument

Verzeichnis der Druckdokumente

• PrintDocRecord

Verzeichnis der Drucktabellen

• Picture

Verzeichnis der Raster- und Kachelgrafiken

MetaPicture

Verzeichnis der Vektorgrafiken

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

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

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

tHdlDir->StoClose();

Über den Deskriptor können verschiedene Eigenschaften eines Verzeichnisses abgefragt werden:

<u>ID</u> ID des Storage-Objektes in der Datenbank

Name des Storage-Objekts

FullName Vollständiger Name (inkl. Pfadangabe)

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

<u>ID</u> ID des Storage-Objektes in der Datenbank

Name des Storage-Objekts
Type Typ des Storage-Objekts

<u>FullName</u> Vollständiger Name (inkl. Pfadangabe)

<u>Created</u> Zeitpunkt der Objekt-Erstellung

<u>Modified</u> Zeitpunkt der letzten Änderung des Objekts <u>CreatedUser</u> Datenbank-Benutzer, der das Objekt erstellt hat

ModifiedUser Datenbank-Benutzer, der das Objekt zuletzt geändert hat

<u>SizeDba</u> Größe des Objektes in der Datenbank in Byte

<u>SizeOrg</u> Original-Größe des Objektes in Byte <u>Unicode</u> Unicode-Unterstützung des Objekts

<u>TypeFile</u> Dateityp des Objekts

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

```
Binäre Objekte
Große binäre Objekte (BLObs - Binary Large Objects)
<u>Liste</u>,
<u>Eigenschaften</u>,
Siehe Befehle für
```

<u>binäre</u>

Objekte, Blog

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

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

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

Beispiel:

```
tHdlDir # BinDirOpen(0, 'Documents', _BinCreate);tHdlSubDir # tHdlDir->BinDirOpen('Excel', _BinCreate)
```

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

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

```
tHdlDir ->BinClose(); tHdlSubDir->BinClose();
```

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

- ID Identität des Verzeichnisses
- Name Name des Verzeichnisses
- FullName Name und Pfad des Verzeichnisses
- <u>Custom</u> Benutzerdefinierte Eigenschaft

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

Beispiel:

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

Bestehende Objekte können ebenfalls wieder exportiert werden.

Beispiel:

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

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

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

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

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

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

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

Beispiele:

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

Binäre Objekte können direkt in die folgenden Objekte geladen werden:

Oberflächen-Objekte Eigenschaft / Funktion

DocViewFileNamePictureCaptionMetaPictureCaption

<u>RtfEdit</u> <u>StreamSource</u> = <u>WinStreamNameBin</u> und <u>FileName</u>

<u>RtfEdit</u> <u>WinRtfPicInsertName()</u>

<u>PrtJobPreview</u> <u>Caption</u>

Druck-Objekte Eigenschaft

 $\begin{array}{ccc} \underline{PrtPicture} & \underline{Caption} \\ \underline{PrtMetaPicture} & \underline{Caption} \\ \underline{PrtPdf} & \underline{FileName} \end{array}$

<u>PrtRtf</u> <u>StreamSource</u> = <u>WinStreamNameBin</u> und <u>FileName</u>

System-Objekte Funktion

Memory BinReadMem()

Objekte für dynamische Strukturen Dynamische Listen und Bäume

Liste.

Eigenschaften

Siehe $\frac{\text{der Elemente}}{\text{Befehle für}}$

dvnamische

Strukturen

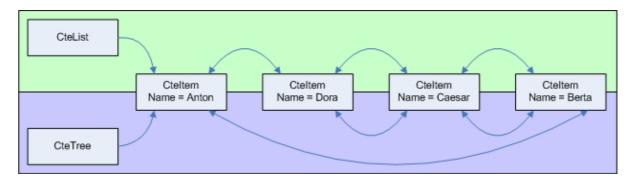
Innerhalb von CONZEPT 16 können sogenannte dynamische Strukturen erzeugt werden. Diese Strukturen werden im Arbeitsspeicher des Clients erzeugt und verarbeitet und ermöglichen so das Generieren von temporären Mengen, die effizient durchsucht oder sortiert werden können.

Für diese Strukturen stehen vier verschiedene Objekte zur Verfügung:

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

CteItem-Objekt

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



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

Das CteItem-Objekt verfügt über folgende Eigenschaften:

- Name
- <u>ID</u>
- Custom

CteList-Objekt

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

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

Das CteList-Objekt verfügt über folgende Eigenschaften:

- Name
- <u>ID</u>
- Custom

CteTree-Objekt

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

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

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

Das CteTree-Objekt verfügt über folgende Eigenschaften:

- Name
- <u>ID</u>
- Custom

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

CteNode-Objekt

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

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

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

erzeugt werden.

Das CteNode-Objekt verfügt über folgende Eigenschaften:

- Name
- <u>ID</u>
- Custom
- Flags
- Parent
- Type
- Value...
- AttribCount
- ChildCount

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

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

Anwendungsbeispiele

1. Temporäre Menge von Datensätzen

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

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

2. Beliebige temporäre Mengen

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

3. Strukturierte Informationen

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

4. Variable Übergabe von Parametern an Funtionen

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

Memory-Objekt

Objekt innerhalb des Speichers zur Verarbeitung beliebiger Daten

Siehe <u>Befehle</u>, <u>Eigenschaften</u>

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

Das Objekt wird mit der Anweisung MemAllocate() angelegt.

HTTP

Beschreibung des HTTP-Objekts

Liste,

Eigenschaften,

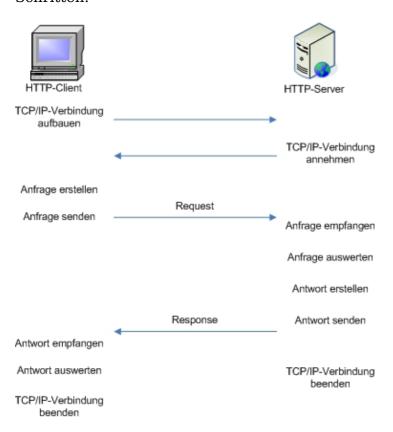
Siehe HttpOpen(),

Beispiel,

Kommunikation

im Web (Blog)

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



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

HttpSendRequest Erstellen und Versenden einer Anfrage

HttpRecvRequest Empfangen und Auswerten einer Anfrage

HttpSendResponse Erstellen und Versenden einer Antwort

HttpRecvResponse Empfangen und Auswerten einer Antwort

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

HttpRecvRequest HttpSendResponse HttpSendRequest HttpRecvRes

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

PDF

Beschreibung des PDF-Objekts

Liste,

Siehe $\frac{Eigenschaften}{PdfNew()}$,

PdfOpen()

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

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

Chart

Diagramm

Befehle,

 $Siehe \frac{Eigenschaften}{ChartData},$

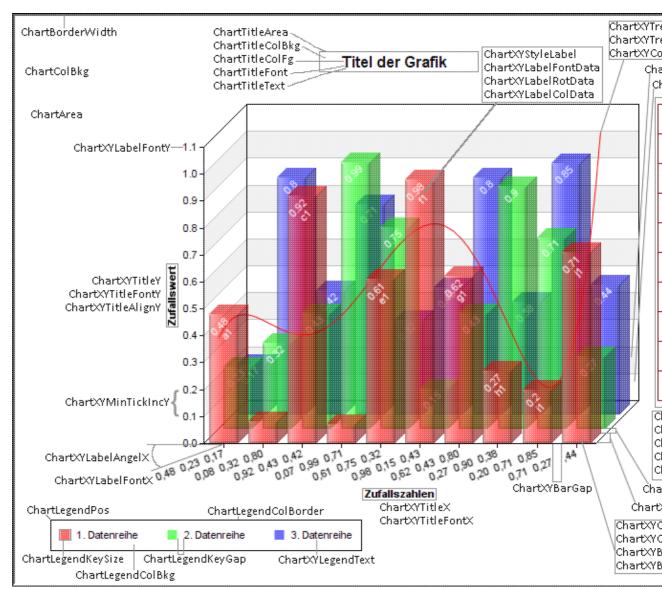
Blog

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

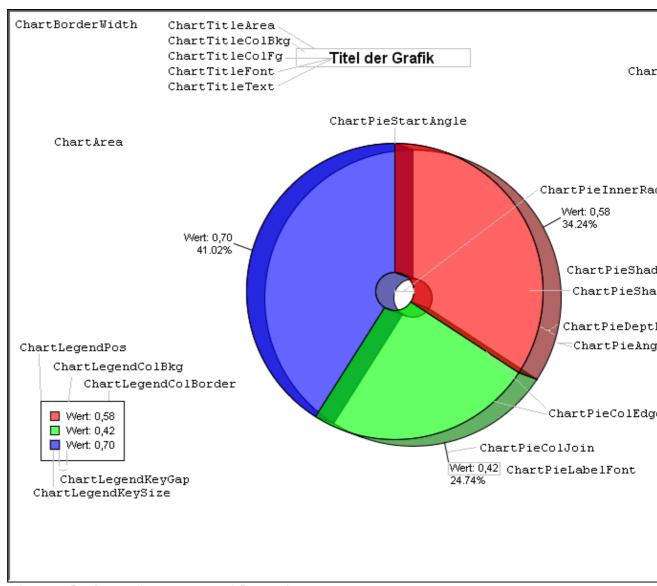
Die Eigenschaften des Objekts bestimmen die Darstellung der einzelnen Datenreihen (siehe ChartData). Bei den Eigenschaften ist vermerkt, ob diese für jede Datenreihe separat gesetzt werden kann.

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

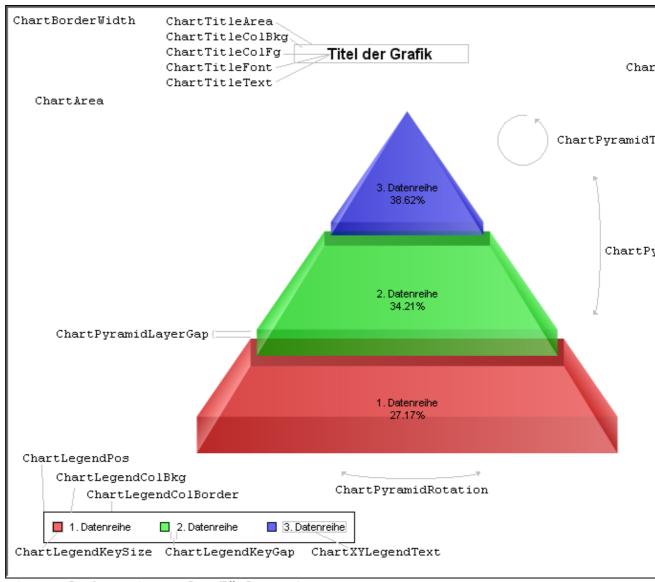
• Eigenschaften eines Koordinaten-Diagramms



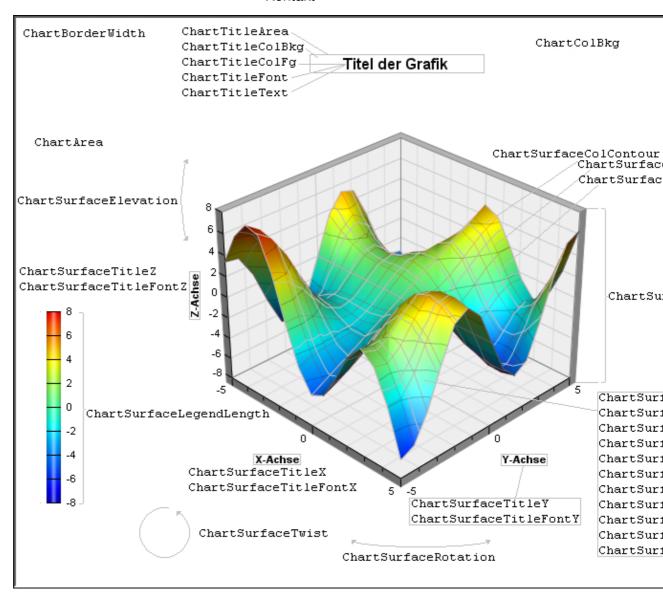
• Eigenschaften eines Torten-Diagramms



• Eigenschaften eines Pyramiden-Diagramms



• Eigenschaften eines Oberflächen-Diagramms



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

ChartData

Darzustellende Daten in einem Diagramm

Siehe $\frac{\text{Befehle}}{\text{Chart}}$,

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

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

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

Das Chart
Data-Objekt selbst verfügt nur über die Eigenschaften
 $\underline{\text{Name}},\,\underline{\text{ID}}$ und
 $\underline{\text{Custom}}.$

TapiDevice

TAPI-Gerätetreiber

Liste,

Siehe <u>Eigenschaften</u>,

Befehle

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

Selection

Selektions-Objekt

Liste,

Eigenschaften,

Siehe SelCreate(),

Dvnamische

Selektionen

(Blog)

In einem Selection-Objekt wird eine Selektion zusammengestellt.

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

Beispiel:

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

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

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

```
tSel # tSel->SelOpen();tErg # tSel->SelRun( SelDisplay | SelWait);
```

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

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

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

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

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

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

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

Weitere Informationen zum Aufbau von Selektionen befinden sich bei den entsprechenden <u>Befehlen für dynamische Selektionen</u>.

Validierungselemente Elemente der Datenbankvalidierung Siehe <u>Liste,</u> <u>Validierungsbefehle</u>

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

Mit dem Befehl <u>VldDirOpen()</u> kann ein Validierungsverzeichnis geöffnet werden.

Nicht mehr benötigte Deskriptoren können mit dem Befehl VldClose() entfernt werden.

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

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

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

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

Beispiel:

tVldDir # VldDirOpen(VldTypeDialog);if (tVldDir > 0){ tVldElm # tVldDir->VldOpen('Frm Main', V

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

Zentrale Datenobjekte Zentral verwaltetes Datenobjekt System-Objekte,

Befehle für

Siehe zentrale

Datenobjekte,

Blog

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



Die Daten werden nicht in der Datenbank gespeichert.

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

Es können beliebig viele Datenobjekte angelegt werden bis zu einem Gesamtspeicherverbrauch von 64 MB.

Drag & Drop-Objekte

Beschreibung der Drag & Drop-Objekte

Drag &

Drop-Ereignisse,

 $Siehe \frac{\underline{Blog},}{Prozedurbeispiel\ in}$

Beispiele-Datenbank

Folgende Objekte werden bei Drag & Drop-Operationen verwendet:

- DataPlace
- DragData
- DragDataFormat

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

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

• EvtDragInit

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

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

• EvtDragTerm

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

• EvtDropEnter

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

• EvtDrop

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

Das <u>DragDataFormat</u>-Objekt wird über das <u>DragData</u>-Objekt zur Verfügung gestellt. Es ist somit in den gleichen Ereignissen erreichbar.

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

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

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

DataPlace

Informationen zum Ziel bei Drag & Drop-Operationen

Siehe <u>Liste</u>, <u>Eigenschaften</u>

Das Objekt wird bei den Ereignissen EvtDragInit, EvtDropOver und EvtDrop übergeben.

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

Die Informationen können den entsprechenden Eigenschaften entnommen werden.

DragData

Daten zur Drag & Drop-Operation

Siehe <u>Liste</u>, <u>Eigenschaften</u>

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

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

DragDataFormat

Format der Drag & Drop-Daten

Siehe <u>Liste</u>, <u>Eigenschaften</u>

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

Der Deskriptor dieses Objekts ist in der Eigenschaft <u>Data</u> des <u>DragData</u>-Objekts hinterlegt.

Der Deskriptor auf die übertragenen Daten befindet sich in der Eigenschaft Data. Abhängig vom Format der Daten befindet sich in der Eigenschaft ein Deskriptor auf eine <u>CTE-Liste</u>, einen <u>CTE-Tree</u> oder einen Textpuffer.

Objekte der Mehrfachselektion Beschreibung der Objekte der Mehrfachselektion SelectionData,

Siehe EvtLstSelectRange,

Befehle der

Mehrfachselektion

Die Mehrfachselektion ist in folgenden Objekten möglich:

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

In den Common-Dialogen (_WinComFileOpen und _WinComFileSave) wird eine Mehrfachselektion über die Eigenschaften FileCount und FileName dargestellt. Damit mehrere Dateien selektiert werden können, muss die Ausprägung <u>WinComAllowMultiSelect</u> in der Eigenschaft <u>Flags</u> gesetzt werden. <u>FileCount</u> gibt dann die Anzahl der selektierten Dateien an, die Namen der Dateien können über das Array FileName abgefragt werden.

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

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

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

SelectionData



Selektionsobjekt der Mehrfachauswahl

Eigenschaften,

Befehle der

Siehe Mehrfachselektion,

MultiSelect,

SelData

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

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

Das SelectionData-Objekt besitzt folgende Eigenschaften:

<u>Custom</u> Benutzerdefinierte Eigenschaft

Deskriptor einer Cte-Liste und eines Cte-Trees

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

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

tCteData # tSelectionData->wpData(WinSelDataCteTree); // Cte-TreetCteData # tSelectionData->wpData

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

• Custom

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

• ID

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

Name

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

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

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

• RecList

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

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

• DataList

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

StoList

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

• TreeView

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

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

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

Eigenschaften

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

<u>Alphabetische</u>

Siehe Liste aller

Eigenschaften

Die Eigenschaften können im Designer durch das Eigenschafts-Fenster und prozedural durch die Befehle <u>WinPropGet()</u> und <u>WinPropSet()</u> bzw. <u>PrtPropGet()</u> und <u>PrtPropSet()</u> gelesen und gesetzt werden.

Die Eigenschaften untergliedern sich in folgende Bereiche:

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

Base (Basis-Eigenschaften)

Eigenschaften sortiert nach

 $Siehe \frac{Gruppen}{Alphabetische}$ Liste aller **Eigenschaften**

Caption Inhalt bzw. Titel des Objektes Inhalt des <u>BigIntEdit</u>-Objektes <u>CaptionBigInt</u> CaptionColAuto Titel für die automatische Farbe Inhalt des ColorEdit-Objektes <u>CaptionColor</u> CaptionDate Inhalt des <u>DateEdit</u>-Objektes CaptionDecimal Inhalt des DecimalEdit-Objektes **CaptionFloat** Inhalt des FloatEdit-Objektes Inhalt des IntEdit-Objektes **CaptionInt CaptionTime** Inhalt des <u>TimeEdit</u>-Objektes

CryptKey Passwort für Verschlüsselung von PrintJob- und binären Objekten

Custom Benutzerdefinierte Eigenschaft

<u>DecEditFlags</u> Optionen der Darstellung

<u>Name</u> Name zur eindeutigen Identifikation des Objektes

<u>StyleBorder</u> Darstellungsart des Randes

StyleCaption Anzeige eines Funktionsergebnisses

<u>TextLabel</u> TextLabel



Caption
Inhalt bzw. Titel des Objekts
Typ <u>alpha(8192)</u>
<u>Liste, Objekte,</u>
Siehe <u>WinPropGet(),</u>
WinPropSet()

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

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

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

• Schaltflächen-Objekte

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

- ♦ Button
- ♦ MenuButton
- **♦** Radiobutton
- ♦ Checkbox
- ♦ NotebookPage
- ♦ MenuItem
- ♦ Groupbox
- ◆ <u>Label</u> (sofern die Eigenschaft <u>ObjLink</u> gesetzt ist)

• Button und Toolbar-Button

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

Statusbar-Button

Folgende Makrodefinitionen können angegeben werden:

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

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

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

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

• MenuItem

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

• <u>Picture</u>, <u>MetaPicture</u>, <u>PrtPicture</u> oder <u>PrtMetaPicture</u>

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

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

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

Beispiele:

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

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

• Progress

Angabe der Einheit des Fortschrittbalkens. Die Einheit wird in dem Balken nach der Eigenschaft <u>ProgressPos</u> angegeben.

• Prt Job Preview

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

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

Beispiele:

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

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

PrtBarcode

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

Beispiel:

'ISBN3-88322-453-7'

• RtfEdit oder PrtRtf

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

Beispiele:

```
tHdlRtfEdit->wpCaption( WinEditDefault) # 'Neuer Text'; // Defaultwert setzen tHdlRtfEdit-
```

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

• TextEdit

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

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

• GroupSplit

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

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

• Application

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



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

• Chromium-Objekt

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

Caption...

Inhalt des Objekts

Typ var

Siehe <u>Liste</u>, <u>Objekte</u>

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

- <u>Caption</u>
- CaptionBigInt
- <u>CaptionColor</u>
- <u>CaptionDate</u>
- CaptionDecimal
- CaptionFloat
- <u>CaptionInt</u>
- <u>CaptionLogic</u>
- <u>CaptionTime</u>

Nur die $\underline{\text{Items}}$ und $\underline{\text{SubItems}}$ eines $\underline{\text{RecView-Objektes}}$ verfügen über einen dynamischen Inhaltstyp.

Captio	onBigInt	
Inhalt	des Objektes	
Typ	<u>bigint</u>	
	Liste, Objekte	<u> </u>
Siehe	WinPropGet()	<u>)</u> ,
	WinPropSet()	_

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

CaptionColAuto \Box	
Titel für automatische Farbe	
Typ <u>alpha</u>	
<u>Liste, Objekte, ColAuto,</u>	
Siehe <u>WinPropGet()</u> ,	
WinPropSet()	

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

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

Ist die Eigenschaft leer, wird "Automatisch" angezeigt.

	onColor : des Objektes	
Typ	<u>int</u>	
	Liste, Objekte	<u> </u>
Siehe	WinPropGet()	Ĺ,
	WinPropSet()	_

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

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

Captio	onDate	
Inhalt	des Objekte	S
Тур	date	
	Liste, Objekt	<u>:e</u> ,
Siehe	WinPropGet	Ω,
	WinPropSet()

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

-	onDecimal des Objektes
Тур	decimal
	<u>Liste</u> , <u>Objekte</u> ,
Siehe	WinPropGet(),
	WinPropSet()

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

Captio	onFloat	
Inhalt	des Objektes	
Тур	float	
	Liste, Objekte	<u> </u>
Siehe	WinPropGet()),
	WinPropSet()	<u> </u>

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

Captio	onInt	
Inhalt	des Objektes	3
Typ	<u>int</u>	
	Liste, Objekt	<u>e</u> ,
Siehe	WinPropGet(),
	WinPropSet())

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

-	onLogic : des Objektes	
Тур	logic	
	Liste, Objekte	<u>2</u> ,
Siehe	WinPropGet()	<u>)</u> ,
	WinPropSet()	

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

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

-	onTime : des Objektes	
Тур	time	
	Liste, Objekte	<u>2</u> ,
Siehe	WinPropGet()	<u>)</u> ,
	WinPropSet()	

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

CryptKey
Passwort für Verschlüsselung von Druckjobs und binären Objekten
Гур <u>alpha</u> (64)
Siehe <u>Liste</u> , <u>Objekte</u> , <u>WinPropGet()</u> , <u>WinPropSet()</u>
Die Eigenschaft definiert ein Passwort, das für die Entschlüsselung von
verschlüsselten Druckjob-Dateien oder binären Objekten verwendet wird. Der Inhalt
kann nur dann angezeigt werden, wenn der korrekte Schlüssel angegeben ist.

Die Eigenschaft kann im Eigenschaftsfenster oder über die Befehle $\underline{\text{WinPropGet()}}$ und $\underline{\text{WinPropSet()}}$ gelesen bzw. geändert werden.

Custo	om [
Benu	itzerdefinierte Eigenschaft	
Typ	<u>alpha</u> (8192)	
o. 1	Liste, Objekte, WinPropGe	et().

Siehe <u>Liste</u>, <u>Objekte</u>, <u>WinPropGet()</u>, <u>WinPropSet()</u>, <u>Beispiel</u>

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

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

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

DecEditFlags \square	
Optionen der Darstellung	
Typ <u>int</u>	
Siehe <u>Liste</u> , <u>Objekte</u>	
In dieser Eigenschaft können werden.	Darstellungsoptionen des Objekts <u>DecimalEdit</u> definiert

Es kann nur die Option $\underline{\underline{\hspace{1cm}WinDecEditDecimalUndef}}$ gesetzt werden.

FlagsTitlebar	
Funktionen der Titelzeile	
Typ <u>int</u>	
Siehe Obiekte	

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

Folgende Ausprägungen stehen zur Verfügung:

• <u>WinTitlebarCaption</u>

Ist diese Option gesetzt, wird eine Titelzeile angezeigt.

• <u>WinTitlebarMove</u>

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

• WinTitlebarClose

Das Objekt kann geschlossen werden.

• WinTitlebarMaximize

Das Objekt kann maximiert werden.

Ist die Option <u>WinTitlebarCaption</u> nicht gesetzt, können auch die anderen Optionen nicht gesetzt werden.

Name	
None a more aim desertioner Identification des Obieltes	

Name zur eindeutigen Identifikation des Objektes Typ <u>alpha</u>(60)

Siehe <u>Liste</u>, <u>Objekte</u>, <u>WinPropGet()</u>, <u>WinPropSet()</u>

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

Bei der Vergabe des Namens ist folgendes zu beachten:

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

 $Obj-Name := {A..z}[[{A..z,0..9, ,.}]]$

StyleBorder Darstellungsart des Randes

Typ int

Liste, Objekte,

Siehe $\frac{\overline{WinPropGet()}}{WinPropSet()}$,

ShowBorder

Diese Eigenschaft bestimmt das Aussehen des Objektrandes.

Es gibt folgende Randstile:

<u>WinBorStandard</u> Kein Rahmen

<u>WinBorSunken</u> Eingedrückt Herausgestellt <u>WinBorRaised</u> <u>WinBorFrame</u> Frame-Rahmen



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

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

StyleCaption
Ausgabe festlegen
Typ <u>int</u>

Liste, Objekte,

Siehe PrtPropGet(),

PrtPropSet()

Diese Eigenschaft bestimmt, welche Ausgabe im <u>PrtText</u> oder <u>PrtTblCellText</u>-Objekt erfolgen soll.

Folgende Konstanten sind zulässig:

<u>PrtStyleCapNormal</u> Der Inhalt der Eigenschaft <u>Caption</u> wird angezeigt

<u>PrtStyleCapPageNo</u> Die aktuelle Seitennummer wird angezeigt

<u>PrtStyleCapPageCount</u> Die Anzahl der Seiten im Druckjob wird angezeigt.

PrtStyleCapFnResult Das Ergebnis einer Funktionsberechnung wird angezeigt. Die

Funktion wird in den <u>Funktions-Eingenschaften</u> festgelegt.

Die Ausprägung wird nur beim PrtTblCellText-Objekt

ausgewertet.

TextLabel
Auszugebender Text
Typ <u>alpha</u>
<u>Liste</u> , <u>Objekte</u> ,
WinPropGet(),
Siehe WinPropSet(),
PrtPropGet(),
PrtPropSet()
Mit dieser Eigenschaft wird der Text definiert, der in einem <u>Picture</u> - oder
MetaPicture-Objekt ausgegeben wird.

Database (Datenbank-Eigenschaften)

<u>Eigenschaften</u> <u>sortiert nach</u>

Siehe Gruppen,

Alphabetische Liste aller Eigenschaften

<u>DbFieldName</u> Verknüpftes Datenbankfeld

DbFileNoNummer der zu verwendenden DateiDbFilterDeskriptor des zu verwendenden FiltersDbKeyNoNummer des zu verwendenden Schlüssels

DbKeyReverseSort Sortierreihenfolge umkehren

<u>DbLinkFileNo</u> Zieldatei der zu verwendenden Verknüpfung

<u>DbRecBuf</u> Datensatzpuffer

DbRecIDDatensatz-IDdes ausgewählten DatensatzesDbRecID6464-Bit Datensatz-IDder ausgewählten GruppeDbSelectionDeskriptor der zu verwendenden SelektionDbTextBufDeskriptor des zu verwendenden Textpuffers

<u>DbVar</u> Globale Datenbereiche <u>StoView</u> Anzuzeigende Objekte

<u>StoViewElmTypes</u> Anzuzeigende Elementtypen <u>UrmParentName</u> Name des Eltern-Objekts <u>UrmParentHandle</u> Deskriptor des Eltern-Objekts

DbFie	ldName	
Verkn	üpftes Datenbankfeld	
Тур	<u>alpha</u> (20)	
	<u>Liste</u> , <u>Objekte</u> ,	
Siehe	WinPropGet(),	
	WinPropSet()	

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

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

Ist in den <u>Einstellungen</u> die Option "Setzen von DbFieldName setzt auch Eingabelimit" aktiviert, wird nach der Angabe eines Datenbankfeldes der zulässige Eingabebereich des Objektes an das Datenbankfeld angepasst. Dies betrifft die Eigenschaften <u>LengthMax</u> (bei Feldern vom Typ <u>alpha</u>) sowie <u>MinInt</u> / <u>MinBigInt</u> und <u>MaxInt</u> / <u>MaxBigInt</u> (bei Feldern vom Typ Ganzzahlig kurz, Ganzzahlig lang und Ganzzahlig (64)). Die entsprechenden Eingabelimits werden sofort umgesetzt, wirken sich also auf die Eingabemöglichkeiten in der <u>Caption-, CaptionInt-</u> bzw. <u>CaptionBigInt-Eigenschaften</u> aus.

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

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

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

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

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

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

stattfinden soll, wird eine der beiden Optionen verwendet.

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

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

DbFi]	leNo	
Num	mer der zu verwendenden Datei	
Тур	<u>int</u>	
0: 1	Liste, Objekte, WinPropGet(),	

Siehe $\frac{\text{Liste}}{\text{WinPropSet()}}$,

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

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

Kontakt DbFilter Deskriptor des zu verwendenden Filters Typ Liste, Objekte,

RecFilterCreate(), Siehe

WinPropGet(), WinPropSet()

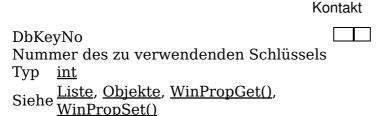
Mit dieser Eigenschaft wird die Anzeige der Datensätze in einem RecList-Objekt auf bestimmte Datensätze, die einem Filter-Kriterium genügen, beschränkt.

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

Beispiel:

tFltHdl # RecFilterCreate(ADR.D.Adressen, ADR.S.aPlz);tFltHdl->RecFilterAdd(1, FltAND, FltLike

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



Die Reihenfolge der Datensätze innerhalb eines <u>RecList</u>-Objekts wird durch den in dieser Eigenschaft angegebenen Schlüssel (oder bei einer Verknüpfung, der dort angegebene Schlüssel) bestimmt. Die Reihenfolge kann durch Setzen der Eigenschaft DbKevReverseSort umgedreht werden.

Ist in der Eigenschaft <u>DbLinkFileNo</u> eine Datei angegeben, erfolgt der Zugriff über die in dieser Eigenschaft angegebene <u>Verknüpfung</u>.

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

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

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

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

DbKe	yReverseSort	
Sortie	erreihenfolge umkehren	
Typ	logic	
	<u>Liste</u> , <u>Objekte</u> ,	
Siehe	WinPropGet(),	
	WinPropSet()	

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

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

Reihenfolge ohne DbKeyReverseSort

	•
CH	3000
CH	4589
CH	4658
CH	6000
D	05895
D	10258
D	25598
D	63150
D	70597

Reihenfolge mit DbKeyReverseSort

D	70597
ע	70397
D	63150
D	25598
D	10258
D	05895
CH	6000
CH	4658
CH	4589
CH	3000

Die Eigenschaften DbKeyReverseSort und <u>DbKeyNo</u> beim <u>RecList</u>-Objekt werden ignoriert, wenn die Sortierung durch Einstellungen in den Spalten vorgegeben werden. Dies wird erreicht durch Setzen der Ausprägung

<u>WinClmSortFlagsAutoActive</u> in der Eigenschaft <u>ClmSortFlags</u> des Spalten-Objekts. Die Sortierung der Liste wird bestimmt durch die Einstellungen der Spalte, die die Ausprägung <u>WinClmSortFlagsAutoSelected</u> in der Eigenschaft <u>ClmSortFlags</u> gesetzt hat.

DbLinkFileNo
Zieldatei der zu verwendenden Verknüpfung
Typ int

Siehe <u>Liste</u>, <u>Objekte</u>, <u>WinPropGet()</u>, <u>WinPropSet()</u> In dieser Eigenschaft wird die Nummer der Zieldatei angegeben, wenn in dem Objekt

nur verknüpfte Datensätze angezeigt werden sollen.

In der Eigenschaft <u>DbFileNo</u> ist die Dateinummer der Ausgangsdatei und in <u>DbKeyNo</u> die Nummer der <u>Verknüpfung</u> anzugeben.

DbRe	cBuf \square
Daten	satzpuffer
Тур	<u>handle</u>
	<u>Liste</u> , <u>Objekte</u> ,
	Datensatzpuffer,
Siehe	<u>Datensatzpuffer</u>
	<u>in MDI-Frames</u>
	(Blog)

Die Abfrage der Eigenschaft kann prozedural erst im <u>EvtCreated</u> des Objektes oder des übergeordneten Objektes erfolgen. Davor wird immer 0 zurückgegeben. Die Eigenschaft DbRecBuf hat je nach Objekt folgende Bedeutung:

• MDI-Anwendungsfenster

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

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

• RecList

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

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

Beispiel:

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

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

Außerhalb des Bereichs wird ebenfalls der Laufzeitfehler <u>ErrValueInvalid</u> ausgelöst. Liegt die Dateinummer innerhalb des Bereichs, existiert jedoch keine Datei mit der Nummer, kommt der Laufzeitfehler <u>ErrFileInvalid</u>.

Beispiele:

// Aktivieren des Datensatzpuffers für die Datei tblCstCustomeraEvt:Obj->wpDbRecBuf(tblCstCustome



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

1/	· _		1 -	1
ĸ	റ	n	ta	IK1

DbRecID RecID des ausgewählten Datensatzes Typ int Siehe <u>Liste</u>, <u>Objekte</u>, <u>WinPropGet()</u> Über diese Eigenschaft kann die <u>Datensatz-ID</u> eines in dem <u>RecList-Objekt</u>

selektierten Datensatzes ermittelt werden.

Die Eigenschaft kann über den Befehl WinPropGet() ermittelt werden.

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

DbRecID64	
64-Bit RecID des ausgewählten Datensatzes / der ausgewählten Gruppe	
Typ <u>bigint</u>	

Siehe <u>Liste</u>, <u>Objekte</u>, <u>WinPropGet()</u>

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

Die Eigenschaft kann über den Befehl WinPropGet() ermittelt werden.



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



DbSe	lection		
Desk	riptor der zu verwendenden Selektion		
Typ	int		
	Lista Ohiokta SalOnan() WinPranCa	+()	

Siehe <u>Liste</u>, <u>Objekte</u>, <u>SelOpen()</u>, <u>WinPropGet()</u>, <u>Beispiel</u>

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

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

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

Beispiel:

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

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

	rtoritaitt
DbTextBuf	
Deskriptor des zu verwendenden	Textpuffers
Typ <u>int</u>	
Sigha Liste, Objekte, TextOpen(),	WinPropGet(),

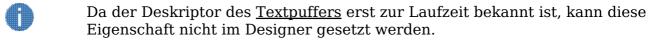
Siehe <u>Liste</u>, <u>Objekte</u>, <u>TextOpen()</u>, <u>WinPropGet()</u> <u>WinPropSet()</u>

Mit dem Setzen dieser Eigenschaft wird der Inhalt des Textpuffers in das Objekt <u>TextEdit</u> übertragen. Mit dem Befehl <u>WinUpdate()</u> und der Option <u>WinUpdBuf2Obj</u> kann das Übertragen des Textpuffers zu jedem Zeitpunkt erfolgen. Das Übertragen des Inhaltes eines TextEdit-Objektes in einen Textpuffer wird über <u>WinUpdate()</u> mit der Option <u>WinUpdObi2Buf</u> vorgenommen.

Kontakt

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

Das <u>TextEdit</u>-Objekt hat keine Limitierung der Zeilenlänge.



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

DbVa	ır 🗆
Globa	ale Datenbereiche
Тур	<u>int</u>
Siehe	<u>Liste, Objekte</u>
Wird	die Eigenschaft mit der Maus aktiviert, wird eine Liste aller globalen
Dater	nbereiche angezeigt.

Die Einträge, die in der Liste selektiert werden, werden automatisch beim Laden der Instanz allokiert. Somit entfällt das Anlegen und Freigeben der Datenbereiche über <u>VarAllocate()</u> und <u>VarFree()</u>.

Die Instanzen der globalen Datenbereiche werden bei Erhalt des Eingabefokus automatisch ausgewählt. Das prozedurale Auswählen der Instanzen über <u>VarInstance()</u> ist nicht notwendig.

StoView	
Anzuzeigende Objekte	
Typ <u>int</u>	
Siehe <u>Liste</u> , <u>Objekte</u>	

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

- WinStoViewDialog
- <u>WinStoViewElmGroup</u>
- <u>WinStoViewMenu</u>
- $\bullet \underline{\quad WinStoViewMetaPicture}$
- <u>WinStoViewPicture</u>
- <u>WinStoViewPrintDoc</u>
- <u>WinStoViewPrintDocRecord</u>
- WinStoViewPrintForm
- <u>WinStoViewPrintFormList</u>
- <u>WinStoViewProcedure</u>
- <u>WinStoViewText</u>
- <u>WinStoViewTheme</u>
- WinStoViewUser
- <u>WinStoViewUserGroup</u>

StoVi	.ewE	lmTypes	
Anzu	zeige	ende Elementtyp	en
Typ	<u>int</u>	-	

Siehe Liste, Objekte

Diese Eigenschaft wird nur ausgewertet, wenn die Eigenschaft <u>StoView</u> auf den Wert <u>WinStoViewElmGroup</u> und die Eigenschaft <u>UrmParentName</u> bzw. <u>UrmParentHandle</u> gesetzt sind. In der Liste werden nur die Elemente angezeigt, die den angegebenen Element-Typen entsprechen. Folgende Konstanten können der Eigenschaft zugewiesen werden:

WinStoViewElmTypeCustom Benutzerdefinierte Objekte

_WinStoViewElmTypeDialog Dialoge _WinStoViewElmTypeMenu Menüs

WinStoViewElmTypePicture Bilder

WinStoViewElmTypePrintDocRecord PrintDocRecord-Objekte

_WinStoViewElmTypeProcedure Prozeduren _WinStoViewElmTypeTable Dateien WinStoViewElmTypeText Texte

_WinStoViewElmTypeTheme <u>Theme</u>-Objekte

WinStoViewElmTypeUser Benutzer

 $_Win Sto View Elm Type User Group \\ Benutzer gruppen$

UrmPa	arentHandle [
Deskri	ptor des Eltern-Objekts	
Тур <u>}</u>	<u>nandle</u>	
Sigho I	<u>Liste, Objekte,</u> UrmParontNamo	
PIGITE I	[IrmDarontNamo	

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

Objekt Angezeigte Objekte

<u>User</u> Benutzergruppen, in dene der Benutzer Mitglied ist.

<u>UserGroup</u> Mitglieder der Benutzergruppe. <u>ElmGroup</u> Elemente der Elementgruppe.

Ist diese Eigenschaft gesetzt, wird die Eigenschaft <u>StoView</u> nicht ausgewertet.

Die Eigenschaft kann nur zur Laufzeit gesetzt werden.

UrmP	arentName	
Name	des Eltern-Objekts	
Typ	<u>alpha</u>	
Siehe	<u>Liste</u> , <u>Objekte</u> ,	

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

Objekt Angezeigte Objekte

<u>User</u> Benutzergruppen, in dene der Benutzer Mitglied ist.

<u>UserGroup</u> Mitglieder der Benutzergruppe. <u>ElmGroup</u> Elemente der Elementgruppe.

Placement (Positions-Eigenschaften)

<u>Eigenschaften</u> <u>sortiert nach</u> Gruppen.

Siehe Gruppen,
Alphabetische
Liste aller
Eigenschaften

<u>Area</u> Position des Objektes

<u>AreaBottom</u> Y-Position des Endpunktes vom Objekt <u>AreaClientRestore</u> Größe des Client-Bereichs herstellen

AreaHeight Höhe des Objektes
AreaHeightCells Höhe der Zellen

<u>AreaHeightMax</u> Maximale Höhe des Objektes

<u>AreaHeightMin</u> Mindesthöhe des Objektes

AreaId ID des dynamischen Bereiches

<u>AreaLeft</u> X-Position des Objektes

AreaMarginBottom Unterer Abstand
AreaMarginLeft Linker Rand
AreaMarginRight Rechter Rand
AreaMarginTop Oberer Rand

<u>AreaRight</u> X-Position des Endpunktes vom Objekt

<u>AreaScaling</u> Overlays mit Skalieren

AreaScalingFlags Optionen zum Skalieren von Overlays

<u>AreaScalingPageNo</u> Overlay an Seite verankern

AreaScreenBildschirmauflösungAreaTopY-Position des ObjektesAreaWidthBreite des ObjektesAreaWidthCellsBreite der Zellen

<u>AreaWidthMax</u> Maximale Breite des Objektes <u>AreaWidthMin</u> Mindestbreite des Objektes

<u>AreaWork</u> Arbeitsbereich

<u>AutoSize</u> Automatisches Anpassen der Größe

Bound Bound-Rechteck ermitteln

BoundAdd Verbrauchten Platz auf Seite ermitteln

BoundMax Maximal zu Verfügung stehenden Platz ermitteln

<u>DialogBoxArrange</u> Ausrichtung der Dialogbox

<u>DialogBoxLeft</u> Horizontale Position der Dialogbox <u>DialogBoxTop</u> Vertikale Position der Dialogbox <u>DividerSize</u> Höhe bzw. Breite der Splitter

<u>DockSide</u> Position der Toolbar

<u>FloatLeft</u> X-Koordinate der Toolbar <u>FloatTop</u> Y-Koordinate der Toolbar

<u>FormHeight</u> Höhe der PrintForm

<u>FormWidth</u> Breite der PrintForm <u>Height</u> Höhe des Objektes

<u>LockAreaSize</u> Größe beim Verändern von Objektpositionen fixieren

MarginBottomUnterer SeitenrandMarginLeftLinker SeitenrandMarginRightRechter SeitenrandMarginTopOberer Seitenrand

<u>Orientation</u> Seitenformat Ausrichtung

<u>PageFormat</u> Seitenformat

<u>PageFormatName</u> Name des Seitenformates

<u>PageHeight</u> Seitenhöhe <u>PageWidth</u> Seitenbreite

Position Position des Objektes

ScrollbarSize Höhe bzw. Breite der Scrollbars

ScrollLeftHorizontale ScrollpositionScrollTopVertikale ScrollpositionScrollWidthBreite des ScrollbereichesScrollHeightHöhe des ScrollbereichesShowPageOffsetSeitenabstand anzeigenStyleGroupGroupTile-Anordnung

Area			

Position des Objektes

Typ rect

Siehe Liste, Objekte

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

Die einzelnen Komponenten der Struktur entsprechen den folgenden Eigenschaften:

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

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

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

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

Beispiel:

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

AreaBottom	
Y-Position des Endpunktes vom Objekt	
Typ <u>int</u>	
Siehe <u>Liste, Objekte, AreaHeight, LockAreaSize</u>	
Vertikale <u>Position</u> des Endpunktes vom	Objekt.

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

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

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

Kontakt

Alternativ dazu kann die Eigenschaft <u>Area</u> auf die neuen Position gesetzt werden.

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

AreaC	ClientRestore	
Größe	e des Client-Bereichs herstellen	
Тур	<u>logic</u>	
Siehe	<u>Liste</u> , <u>Objekte</u> , <u>WinPropGet()</u> , WinPropSet()	

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

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

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

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

Beispiel:

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

Areal	Height [
Höhe	vom Objekt	
Typ	<u>int</u>	
Siehe	Liste, Objekt	<u>:e</u> ,
	AreaBottom, AreaWidth,	
	<u>AreaWidth</u> ,	
	LockAreaSiz	6

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

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

Areal	IeightCells 🔲
Höhe	der Zelle
Тур	<u>int</u>
Siehe	<u>Liste</u> , <u>Objekte</u> ,
	<u>AreaWidthCells</u> , <u>PrtPropGet()</u> ,
	PrtPropGet(),
	PrtPropSet()

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

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

Das Setzen der Eigenschaft ändert ebenfalls die Eigenschaft AreaBottom.

Areal	HeightMax	
Maxir	nale Höhe des Objekts	
Typ	<u>int</u>	
	<u>Liste</u> , <u>Objekte</u> ,	
Siehe	WinPropGet(),	
	WinPropSet()	

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

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

	leightMin esthöhe des Objekts	
Typ	U	
	<u>Liste</u> , <u>Objekte</u> ,	
Siehe	WinPropGet(),	
	WinPropSet()	

Über diese Eigenschaft kann die Mindesthöhe eines Frame- oder <u>GroupTile</u>-Objekts vorgegeben werden.

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

AreaId			
ID des dynamischen Bereiches			
Typ <u>int</u>			
<u>Liste</u> ,	<u>Objekte</u> ,		
Siehe WinPr	opGet(),		

WinPropSet()

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

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

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

AreaI	Left	
X-Pos	sition des Objektes	
Typ	<u>int</u>	
	<u>Liste</u> , <u>Objekte</u> ,	
Sicho	<u>AreaRight</u> ,	
Siene	AreaRight, AreaWidth,	
	<u>LockAreaSize</u>	
Horiz	contale <u>Position</u> des	s Objektes.

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

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

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

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

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

	MarginBottom Comer Abstand
Тур	<u>int</u>
	<u>Liste</u> , <u>Objekte</u> ,
	WinPropGet(),
Siehe	WinPropSet(),
	PrtPropGet(),
	PrtPropSet()

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

• GroupSplit-Objekt

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

• Spalte von RecView-Objekten

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

• Druck-Objekte

Die Eigenschaft definiert den unteren Seitenrand in logischen Einheiten.

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

Beispiel:

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

Area	MarginLeft□□□
Linke	er Rand
Typ	<u>int</u>
	<u>Liste</u> , <u>Objekte</u> ,
	WinPropGet(),
Siehe	e <u>WinPropSet()</u> ,
	<pre>PrtPropGet(),</pre>
	PrtPropSet()

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

• GroupSplit-Objekt

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

• Spalte von RecView-Objekten

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

• Druck-Objekte

Die Eigenschaft definiert den linken Seitenrand in logischen Einheiten.

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

Beispiele:

tGroupSplit->wpAreaMarginLeft # 2;tColumn->wpAreaMarginLeft # 4;tPage->ppAreaMarginLeft # PrtUni

Areal	MarginRight 🔲
Recht	er Rand
Тур	<u>int</u>
	<u>Liste</u> , <u>Objekte</u> ,
	WinPropGet(),
Siehe	WinPropSet(),
	PrtPropGet(),
	PrtPropSet()

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

• GroupSplit-Objekt

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

• Spalte von RecView-Objekten

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

• Druck-Objekte

Die Eigenschaft definiert den rechten Seitenrand in logischen Einheiten.

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

Beispiele:

tGroupSplit->wpAreaMarginRight # 2;tColumn->wpAreaMarginRight # 4;tPage->ppAreaMarginRight # Prtl

AreaN	1arginTop	
Oberer Rand		
Тур	<u>int</u>	
	<u>Liste</u> ,	
	<u>Objekte</u> ,	
Sioho	<pre>WinPropGet(), WinPropSet(),</pre>	
Siene	WinPropSet(),	
	PrtPropGet(),	
	PrtPropSet()	

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

• GroupSplit-Objekt

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

• Spalte von RecView-Objekten

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

• Druck-Objekte

Die Eigenschaft definiert den oberen Seitenrand in logischen Einheiten.

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

Beispiele:

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

AreaRight
X-Position des Endpunktes vom Objekt
Typ <u>int</u>
Siehe <u>Liste</u> , <u>Objekte</u> , <u>AreaWidth</u> , <u>LockAreaSize</u>
LockAreaSize LockAreaSize
Horizontale Position des Endpunktes vom Objekt.

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

Der Wert in der Eigenschaft AreaRight kann nicht kleiner sein als der Wert in <u>AreaLeft</u>.

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

Kontakt

Alternativ dazu kann die Eigenschaft <u>Area</u> auf die neue Position gesetzt werden.

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

AreaScaling	
Overlays mit	Skalieren
Typ <u>logic</u>	
Siehe <u>Liste, (</u>	<u>Objekte</u> ,
Sielle AreaSc	ralingFlags

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

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



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

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

AreaScalingFlags
Optionen zum Skalieren von Overlays
Typ <u>int</u>
Siehe <u>Liste</u>, <u>Objekte</u>, <u>AreaScaling</u>

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

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

Die Eigenschaft wird nur ausgewertet, wenn die Eigenschaft <u>AreaScaling</u> gesetzt ist.

AreaS	calingPageNo
Overla	ny an Seite verankern
Typ	<u>int</u>
Siehe	<u>Liste, Objekte</u>
Diese	Eigenschaft kann nur gesetzt werden, wenn das Objekt als Overlay in einem
überg	eordneten Objekt erzeugt wurde.

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

Areas	Screen	
Bilds	chirmauflösung	ī
Typ	<u>rect</u>	
	Liste, Objekte,	
Siehe	AreaWork,	

WinPropGet()

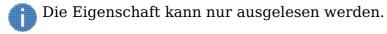
Über diese Eigenschaft kann die Größe des Primärbildschirms in Pixeln ermittelt werden. Das Resultat ist ein Wert vom Typ <u>rect</u>, wobei die Komponenten <u>left</u> und <u>top</u> beim Primärbildschirm immer 0 sind.

Der ermittelte Wert entspricht den Rückgabewerten von <u>WinInfo(_WinScreenWidth)</u> und <u>WinInfo(_WinScreenHeight)</u>.

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

Beispiel

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



Mögliche Laufzeitfehler:

 $\underline{\underline{\text{ErrValueInvalid}}}_{\text{vorhanden.}} \overset{\text{Ein Bildschirm mit der in Klammern angegebenen Nummer ist nicht}}{\text{vorhanden.}}$

AreaT	°op
Y-Posi	ition des Objektes
Тур	<u>int</u>
	<u>Liste</u> , <u>Objekte</u> ,
Ciobo	AreaBottom,
Siene	<u>AreaBottom</u> , <u>AreaHeight</u> ,
	<u>LockAreaSize</u>
Vertik	kale <u>Position</u> des Objektes.

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

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

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

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

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

AreaV	Vidth	
Breite	e vom Obje	kt
Typ	<u>int</u>	
	Liste, Obje	<u>ekte</u> ,
Siehe	AreaRight	,
	AreaHeigh	<u>1t</u> ,
	InckArea	Size

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

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

AreaV	VidthCells 🔲
Breite	e der Zelle
Typ	<u>int</u>
	<u>Liste</u> , <u>Objekte</u> ,
Siehe	<u>AreaHeightCells</u> , <u>PrtPropGet()</u> ,
	<pre>PrtPropGet(),</pre>
	PrtPronSet()

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

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

Das Setzen der Eigenschaft ändert ebenfalls die Eigenschaft AreaRight.

	VidthMin estbreite des Objekts	
Typ	<u>int</u>	
	Liste, Objekte,	
Siehe	WinPropGet(),	
	WinPropSet()	

Über diese Eigenschaft kann die Mindestbreite eines Frame- oder <u>GroupTile</u>-Objekts vorgegeben werden.

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

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

AreaWidthMax			
Maximale Breite des Objektes			
Typ <u>int</u>			
<u>Liste, Objekte,</u>			
Siehe <u>WinPropGet()</u> ,			
WinPropSet()			

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

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

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

Areal	Vork	
Arbei	tsbereich	
Тур	<u>rect</u>	
	Liste, Obje	<u>kte</u> ,
Siehe	AreaScreen	<u>1</u> ,
	WinPropGe	et()

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

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

Beispiel

// Arbeitsbereich des PrimärbildschirmstRect # _App->wpAreaWork;tRect # _App->wpAreaWork(0);// A



Die Eigenschaft kann nur ausgelesen werden.

Mögliche Laufzeitfehler:

 $\underline{\underline{\text{ErrValueInvalid}}}$ Ein Bildschirm mit der in Klammern angegebenen Nummer ist nicht vorhanden.

AutoSize	
Automatisches Anpassen der Größe	

Typ <u>logic</u>
<u>Liste</u>, <u>Objekte</u>, <u>WinPropGet()</u>,

Siehe WinPropSet(), PrtPropGet(),

PrtPropSet()

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

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

Boun	d	
Boun	d-Rechteck ermitteln	
Typ	<u>rect</u>	
Ciobo	Liste, Objekte, PrtPropGet()	
Siene	PrtPropGet()	

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

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

BoundAdd	
Bouria iaa	

Verbrauchten Platz auf Seite ermitteln

Typ <u>point</u>

Siehe <u>Liste, Objekte, BoundMax,</u> <u>PrtPropGet(), Beispiel</u>

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

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

Beispiel

// Bereits verbrauchter Platz auf SeitetAddSize # tPage->ppBoundAdd;// Maximal zu Verfügung stehe



BoundMax
Maximal zu Verfügung stehenden Platz ermitteln
Typ point
Siehe Liste, Objekte, BoundAdd, PrtPropGet(),

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

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

- <u>PrtBoundPage</u> Druckbereich (default)
- <u>PrtBoundHeader</u> Seitenkopf
- PrtBoundFooter Seitenfuß

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

Beispiel:

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

	gBoxArrange chtung der Dialogbox	
Тур	<u>int</u>	
	<u>Liste</u> , <u>Objekte</u> ,	
	DialogBoxLeft,	
Siehe	<u>DialogBoxTop</u> ,	
	WinPropGet(),	
	WinPropSet()	

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

Diese Eigenschaft wirkt sich nur aus, wenn die Eigenschaften <u>DialogBoxLeft</u> und <u>DialogBoxTop</u> den Wert -1 haben.

Folgende Werte sind zulässig:

<u>WinDialogCenter</u>	DialogBox zentriert zum Elternfenster anzeigen.
<u>_WinDialogCenterX</u>	DialogBox horizontal zentriert zum Elternfenster
•	anzeigen.
<u>_WinDialogCenterY</u>	DialogBox vertikal zentriert zum Elternfenster anzeigen.
<u>WinDialogCenterScreen</u>	DialogBox zentriert zum Bildschirm anzeigen.
<u>WinDialogCenterScreenX</u>	DialogBox horizontal zentriert zum Bildschirm anzeigen.
<u>WinDialogCenterScreenY</u>	DialogBox vertikal zum Bildschirm anzeigen.
<u> WinDialogCenterNone</u>	DialogBox vom Betriebssystem positionieren lassen
	(Standard).

DialogBoxLeft	
Horizontale Position der Dialog-	Box
Typ <u>int</u>	
Liste, Objekte, DialogBox	<u>Гор</u> ,
Siehe <u>Liste</u> , <u>Objekte</u> , <u>DialogBox</u> <u>WinPropGet()</u> , <u>WinPropSe</u>	t()

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

Enthält die Eigenschaft den Wert -1, übernimmt das Betriebssystem die Positionierung der Dialogbox. Bei einem Wert größer oder gleich 0, wird die Dialogbox entsprechend der Koordinate positioniert. Liegt die Position außerhalb des primären Bildschirms, wird die Dialogbox entsprechend justiert.

Die vertikale Position wird in der Eigenschaft <u>DialogBoxTop</u> angegeben.

DialogBoxTop \Box		
Vertil	kale Position der Dialog-B	Box
Тур	int	
Ciaba	Liste, Objekte, DialogBo	xLeft,
Siene	<u>Liste</u> , <u>Objekte</u> , <u>DialogBor</u>	Set()

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

Enthält die Eigenschaft den Wert -1, übernimmt das Betriebssystem die Positionierung der Dialogbox. Bei einem Wert größer oder gleich 0, wird die Dialogbox entsprechend der Koordinate positioniert. Liegt die Position außerhalb des primären Bildschirms, wird die Dialogbox entsprechend justiert.

Die horizontale Position wird in der Eigenschaft <u>DialogBoxLeft</u> angegeben.

Höhe bzw. Breite der Splitter

Typ int

Liste, Objekte,

Siehe WinPropGet(),

WinPropSet()

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

- CodeEdit
- DataList
- <u>DocView</u>
- <u>GanttGraph</u>
- GroupSplit
- RecList
- RecView
- StoList

Folgende Werte sind möglich:

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

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

DockSide
Position der Toolbar
Typ <u>int</u>
<u>Liste, Objekte,</u>
Siehe <u>WinPropGet()</u>,

WinPropSet()

Diese Eigenschaft bestimmt die Position der Toolbar.

Es gibt folgende Positionen:

WinDockTop obere Fensterseite
WinDockRight rechte Fensterseite
WinDockBottom untere Fensterseite
WinDockBottomTop untere Fensterseite
WinDockLeft linke Fensterseite

FloatLeft				
x-Koordinate der Toolbar				
Typ <u>int</u>				
<u>Liste, Objekte,</u>				
Siehe WinPropGet(),				

<u>WinPropSet()</u>
Diese Eigenschaft gibt die x-Koordinate an, die die <u>Toolbar</u> beim Lösen aus dem <u>ToolbarDock</u>-Objekt einnehmen soll.

FloatTo	•
1-K001'u	inate der Toolbar
Typ <u>in</u>	<u>t</u>
<u>Li</u>	ste, <u>Objekte</u> ,
Siehe W	<u>inPropGet()</u> ,
$\underline{\mathbf{W}}$	<u>'inPropSet()</u>
Diese Ei	igenschaft gibt die y-Koordinate an, die die Toolbar beim Lösen

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

FormHeight	
Höhe der PrintForm / des Canvas	
Typ <u>int</u>	
Siehe Liste, Objekte, Beispiel	

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

Wird die Höhe eines <u>PrintFormList</u>-Objektes ausgelesen, können mit der Option <u>PrtLstUpdate</u> (tHeight # tForm->ppFormHeight(_PrtLstUpdate)) alle <u>PrtLine</u>-Objekte neu angeordnet und die Höhe aller <u>PrtLine</u>-Objekte neu berechnet werden. Anschließend wird die neue Höhe des <u>PrintFormList</u>-Objektes (<u>MarginTop</u> + Höhe der <u>PrtLine</u>-Objekte) zurückgegeben. Diese Höhe kann z. B. bei der manuellen Ermittlung des Seitenumbruchs durch <u>PrtAdd()</u> herangezogen werden.

FormWidth	
Breite der PrintForm / des Canvas	
Typ <u>int</u>	
Siehe Liste, Obiekte	

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

HdrH	Height]
	e der Spaltenköpfe	
Тур	<u>int</u>	
	HdrVerticalText,	
	HdrWordBreak,	
Siehe	e <u>Liste</u> , <u>Objekte</u> ,	
	WinPropGet(),	
	WinPropSet()	

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

Objekte von HdrHeight

Folgende Objekte unterstützen die Eigenschaft HdrHeight

HdrHeight,

<u>Liste der</u>

Siehe Objekte, Liste

<u>der</u>

Eigenschaften

- DataList
- <u>DataListPopup</u>
- RecList
- RecListPopup
- RecView
- StoList
- StoListPopup

Heigh	ıt 🔲
Höhe	
Typ	<u>int</u>
	<u>Liste</u> ,
	Objekte,
Sioho	<pre>WinPropGet(), WinPropSet(),</pre>
Sielle	WinPropSet(),
	<pre>PrtPropGet(),</pre>
	PrtPropSet()

• GroupTile-Objekt

Die Eigenschaft bestimmt die Höhe des Objekts:

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

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

Beispiele:

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

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

• PrtTblCell-Objekt

Beim <u>PrtTblCell</u>-Objekt kann über die Eigenschaft die Zellhöhe in <u>logischen</u> <u>Einheiten</u> definiert werden.