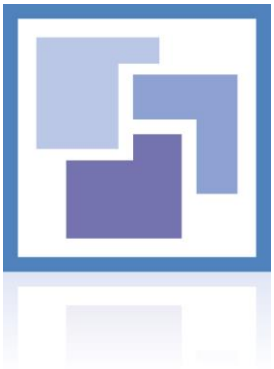


combit Software Development Kit (SDK) Dokumentation

Programmierer-Referenz Objekt Modell (COM)

Copyright © combit GmbH 1988-2016; Rev. 8.008

http://www.combit.net

Alle Rechte vorbehalten.

Inhalt

1 Einführung ... 4

1.1 Welche Scriptsprachen werden unterstützt? ... 4
1.2 Wo lassen sich Scripte integrieren? ... 5
1.3 Support zu Scripting-Funktionalitäten ... 5
1.4 Script-Erweiterungen ... 5

2 Objekthierarchie ... 11

3 Objekt-Referenz ... 12

3.1 ActiveWindow Objekt ... 12
3.2 Application/cRM Objekt ... 12
3.3 Appointment Objekt ... 27
3.4 Appointments Objekt ... 35
3.5 Attendee Objekt ... 37
3.6 Attendees Objekt ... 38
3.7 CallItem Objekt ... 40
3.8 CallList Objekt ... 41
3.9 Categories Objekt ... 42
3.10 Category Objekt ... 44
3.11 CompanyInfo Objekt ... 44
3.12 Container Objekt ... 48
3.13 DialogForm Objekt ... 49
3.14 DocMngr Objekt ... 60
3.15 InputForm Objekt ... 62
3.16 Link Objekt ... 66
3.17 Links Objekt ... 69
3.18 ListCodeDefinitions Objekt ... 70
3.19 ListContainers Objekt ... 71
3.20 ListWebElements Objekt ... 72
3.21 ListRelations Objekt ... 72
3.22 ListViewConfigs Objekt ... 74
3.23 ListViews Objekt ... 75
3.24 OLEError Objekt ... 77
3.25 phonemanager Objekt ... 77
3.26 Project Objekt ... 79
3.27 Record Objekt ... 90
3.28 RecordSet Objekt ... 106
3.29 RecurrencePattern Objekt ... 127
3.30 Relation Objekt ... 131
3.31 SQLShell Objekt ... 132
3.32 timemanager Objekt ... 134
3.33 ToDo Objekt ... 136
3.34 ToDos Objekt ... 145
3.35 User Objekt ... 147
3.36 Users Objekt ... 150
3.37 View Objekt ... 152
3.38 ViewConfig Objekt ... 157
3.39 WebElement Objekt ... 163
3.40 WScript Objekt ... 164

4 Menü-IDs ... 170

4.1 Projekt-Menü ... 170
4.2 Ansichten-Menü ... 171
4.3 Termine & Aufgaben Menü ... 176
4.4 Kontextmenüs ... 177
4.5 phone manager Menü ... 180

5 Ereignisse ... 181

5.1 Event Objekt ... 181
5.2 Projektspezifische Ereignisse ... 183
5.3 Termine und Aufgaben Ereignisse ... 185
5.4 Ansichtenspezifische Ereignisse ... 187
5.5 EventFieldChange Objekt ... 191
5.6 ListEventFieldChange Objekt ... 192

6 eMail Autopilot ... 193

6.1 Attachment Objekt ... 193
6.2 Attachments Objekt ... 194
6.3 HostApp Objekt ... 194
6.4 Mail Objekt ... 196

7 Änderungen und Neuerungen in der Version 8 ... 200

7.1 Eigenschaften und Methoden ... 200

8 Änderungen und Neuerungen in der Version 7 ... 202

8.1 Eigenschaften und Methoden ... 202
8.2 Ereignisse ... 202

9 Änderungen und Neuerungen in der Version 6 ... 203

9.1 Objekte ... 203
9.2 Eigenschaften und Methoden ... 203
9.3 Ereignisse ... 204

10 Änderungen und Neuerungen in der Version 5 ... 206

10.1 Objekte 206
10.2 Eigenschaften und Methoden ... 206
10.3 Ereignisse ... 207

11 Webansicht ... 208

11.1 Dynamische Inhalte in HTML-Seiten ... 208

12 Web-Elemente ... 210

12.1 Direkt per URL-Aufruf ... 210
12.2 Innerhalb einer Webseite ... 210
12.3 Aus einem Script ... 210

13 FAQ ... 211

13.1 Script-Beispiele ... 211

14 COM-Schnittstelle für .NET ... 243

15 combit Knowledgebase ... 244

1 Einführung

Ihnen steht ein mächtiges Objektmodell zur Verfügung, das den Zugriff auf die Ansichten, die Telefonie, das Terminmanagement und vieles mehr erlaubt. Mit Hilfe von Scripten können Sie dieses Objektmodell ansprechen und somit viele Routineaufgaben komfortabel automatisieren.

Sogar eventuell nicht vorhandene spezielle Funktionen können Sie so leicht selbst integrieren. Darüber hinaus können Scripte andere Programme ansprechen und deren Funktionalitäten nutzen, sofern diese eine entsprechende Schnittstelle anbieten.

> Hinweis: Ein Script ist eine Abfolge von Befehlen, die bei der Ausführung sequenziell abgearbeitet werden. Die Befehle entstammen dabei dem "Wortschatz" einer bestimmten Scriptsprache. Dieser Befehlssatz bestimmt dabei, welche Möglichkeiten die Sprache bietet und wie ein Script aufgebaut sein muss.

Scripts sind in der Regel nicht allzu umfangreich und führen schon mit wenigen Befehlen zu beachtlichen Leistungen. Ein durchschnittliches Script umfasst vielleicht 20 bis 40 Zeilen Befehle. Nicht zuletzt aus diesen Gründen sind Scriptsprachen meist sehr leicht zu erlernen.

Obwohl oberflächlich sehr ähnlich, gibt es doch eine ganze Reihe von entscheidenden Unterschieden zwischen Scripts und ausführbaren Programmen.

So sind Scripts beispielsweise nicht selbst lauffähig, sondern benötigen immer eine Umgebung, in der sie ablaufen. Diese sogenannten Hosts sind für die Verwaltung der Scripts verantwortlich und erweitern die Möglichkeiten der jeweiligen Sprache meist in Form von zusätzlichen Objekten.

In unserem Fall ist die combit Anwendung der Host; eine Erweiterung findet durch dessen Objektmodell statt. Ein weiterer wichtiger Unterschied ist die fehlende Möglichkeit, eigenständig Dialoge zu implementieren. Hierzu sind externe Objekte notwendig.

1.1 Welche Scriptsprachen werden unterstützt?

Grundsätzlich werden alle Scriptsprachen des Windows Scripting Host unterstützt. Am weitesten verbreitet sind VBScript und JScript, die direkt vom Hersteller Microsoft angeboten werden. Es gibt aber auch andere Umsetzungen wie z.B. Python. Die Auswahl der Sprache erfolgt dabei über die Dateiendung, z.B. "vbs" bei VBScript und "js" bei JScript.

> Hinweis: VBScript und JScript sind üblicherweise schon auf Ihrem System installiert. Falls nicht oder falls andere Sprachen verwendet werden sollen, müssen diese vom jeweiligen Hersteller bezogen werden und entsprechend seinen Angaben auf dem System installiert werden.

Neben den Scriptsprachen des Windows Scripting Host wird auch C# als Scriptsprache unterstützt. Die Dateiendung lautet in diesem Fall "csscript".

1.1.1 Hinweise zur Benutzung von C#-Scripten

Bei C# Scripten steht Ihnen der volle Funktionsumfang des .NET Frameworks 4.0 zur Verfügung, weshalb für das Ausführen von C# Scripten auch mindestens dieses oder ein neueres .NET Framework installiert sein muss. Außerdem müssen die Dateien "combit.CSharpScript.Engine.dll", "combit.CSharpScript.Interface.dll", "MSVCR120.dll" und "MSVCP120.dll" im Hauptverzeichnis der cRM-Installation liegen.

Kleinere C#-Scripte können direkt als einzelne Zeilen in eine Datei oder die Eingabemaske geschrieben werden. Wenn jedoch verschiedene Methoden verwendet werden sollen, so muss der Hauptteil des Scripts in die Main-Methode (public static void Main()) verlagert werden. Ein C#-Script ohne Main-Methoden, mit jedoch mindestens einer eigenen Methodendefinition kann nicht ausgeführt werden.

Alle im Script definierten Methoden müssen mit dem Schlüsselwort "static" versehen sein. Klassendefinitionen werden derzeit nicht unterstützt, um dennoch eigene Klassen zu verwenden können diese in externen Bibliotheken definiert und diese im Script inkludiert werden.

Die in diesem Dokument aufgelisteten Objekte der Objekt-Referenz dürfen ausschließlich im Ursprungsthread des Scripts erstellt werden. Wenn im Script ein neuer Thread erstellt und in diesem wiederum cRM-Objekte erstellt werden, so können diese beim Beenden des Scripts nicht korrekt aufgeräumt werden. Dadurch kann es bei der weiteren Benutzung des cRMs zu Fehlern kommen.

Da C# Scripte vor der Benutzung kompiliert werden müssen und dies unter Umständen länger dauern kann als das Ausführen des Scriptes selbst, werden die 30 neuesten kompilierten Scripte für ein erneutes Ausführen zwischengespeichert. Die entsprechenden Dateien befinden sich im Ordner "%temp%\combitCSharpScriptCache\". Dort gibt es zum einen die Datei "combitCSharpScriptCache.cache", welche Informationen über die gespeicherten Scripte enthält, und zum anderen für jedes Script einen Ordner mit einem Namen in der Form "combitCSharpScript_[GUID]" (z.B. "combitCSharpScript_e9527e037aa149f3ba79bd408a8232db"). In diesem Ordner befinden sich jeweils alle vom Script benutzten .dll-Dateien, Debug-Informationen und das Script selbst (ebenfalls in Form einer .dll-Datei).

> **Hinweis:** Die Definition einer Ausnahme für den Cache-Ordner bei Echtzeit-Virenscannern kann eine Geschwindigkeitssteigerung bei der Ausführung der Scripte bewirken.

1.2 Wo lassen sich Scripte integrieren?

Die Integration von Scripten ist über Schaltflächen innerhalb der Eingabemaske (Scriptdatei ausführen, Scriptzeilen ausführen, Funktionsdefinitionen) und Ansichten-Symbolleiste (Funktionsdefinitionen), über den Befehl **Extras > Script > Ausführen**, über Folgeverknüpfungen oder über Ereignisse möglich.

Eine Beschreibung, wie die Scripts dort konkret hinterlegt werden, finden Sie innerhalb der zugehörigen Kapitel des Handbuches. Sie können dabei jeweils entscheiden, ob Sie das Script direkt eingeben oder in Form einer externen Textdatei hinterlegen möchten. Insbesondere bei größeren Scripts ist letzteres die bessere Wahl, da diese eine leichte Wiederverwertbarkeit an anderer Stelle erlaubt.

1.3 Support zu Scripting-Funktionalitäten

Die Möglichkeiten der Scripting-Technologie sind sehr weitreichend und deren Beschreibung daher sicherlich Stoff für ein eigenes Buch. Wir bitten um Ihr Verständnis, dass wir keine Beschreibung der verwendeten Scriptsprachen liefern können. Entsprechende Werke finden Sie im Buchprogramm vieler größerer Fachverlage.

Eine ausführliche Referenz zu VBScript und JScript bietet Microsoft im Internet an: http://www.microsoft.com/germany/technet/scriptcenter/.

Selbstverständlich wollen wir Ihnen im Rahmen unseres Supports bei Ihren Fragen und Wünschen gerne mit Rat und Tat zur Seite stehen, um Ihnen einen optimalen Einsatz unseres Produktes zu ermöglichen. Wir möchten jedoch um Verständnis bitten, dass wir nur auf Fragen zum Objektmodell selbst, nicht aber zu Modellen anderer Produkte oder zu den Scriptsprachen selbst eingehen können.

1.4 Script-Erweiterungen

Diese Script-Erweiterungen gelten nur für interne Scripte, d.h. für Scripte die über Schaltflächen (Script Direkt, Script Datei, Funktionsdefinitionen) innerhalb der Eingabemaske, über den Befehl **Extras > Script > Ausführen** oder über Ereignisse ausgeführt werden.

> **Hinweis:** Präprozessor-Direktiven wie "#include..." und "#pragma..." können nicht auskommentiert werden und werden daher in jedem Fall ausgeführt.

1.4.1 Scripte in Scripten einbinden

Bei häufig benötigten Funktionen bietet es sich an, diese zentral abzulegen, so dass sich eventuell notwendige Änderungen auf alle darauf basierenden Scripts auswirken. Hierzu wird die Einbindung von Scripts über eine spezielle Anweisung der folgenden Form unterstützt:

<!--#include file="c:\scripts\include.vbs"-->

Die Anweisung wird dabei durch den kompletten Inhalt der angegebenen Datei ersetzt. Um eventuellen Syntaxfehlern vorzubeugen, empfiehlt es sich daher, die Anweisung in eine eigene Zeile zu setzen.

> Hinweis: Alle auf diese Weise eingebundenen Scripts müssen die gleiche Scriptsprache verwenden wie das Hauptscript. Eine Mischung von mehreren Sprachen ist nicht möglich und führt zu Syntaxfehlern.

Sofern Sie Ihre Scripts unterhalb des Programmverzeichnisses abgelegt haben, können Sie statt einer festen Angabe des Verzeichnisses die %APPDIR% Variable verwenden:

<!--#include file="%APPDIR%\include.vbs"-->

Tipp: Die Variable %APPDIR% steht auch an anderen Stellen zur Verfügung, z.B. in der Eingabemaske, bei den Funktionsdefinitionen und innerhalb der Symbolleisten.

%APPDIR% kann als Platzhalter verwendet werden, der vom Programm durch den Pfad der Anwendung ersetzt wird, wie z.B. C:\Programme\combit\cRM\.

%PRJDIR% wird analog dazu durch den Pfad des Projekts ersetzt.

1.4.2 Synchrone und Asynchrone Scripte

Da die Einsatzgebiete von Scripten sehr unterschiedlich sind, stehen zwei verschiedene Ausführungsarten zur Auswahl.

1.4.2.1 Synchrone Scripte

Ein synchrones Script hält den Host solange auf, bis es vollständig abgearbeitet ist. Wenn Sie also bspw. innerhalb eines Scriptes zeitlich längere Operationen durchführen, können Sie während der kompletten Ausführungszeit nicht mit dem Programm arbeiten.

Synchrone Scripte bieten sich daher für Abläufe an, die im direkten Kontext zur aktuellen Arbeit stehen, wie z.B. die Automatisierung von mehreren kleineren Routineaufrufen, wie einem Filter und dem anschließenden Export. Jedes nicht anders gekennzeichnete Script ist automatisch ein synchrones Script.

1.4.2.2 Asynchrone Scripte

Diese Art von Scripts wird parallel zum Host ausgeführt, so dass Sie währenddessen mit diesem arbeiten können. Es wird hierfür ein eigener sogenannter "Thread" geöffnet. Auf diese Art ist der Host während der Ausführung nicht blockiert.

Wir empfehlen die Verwendung von asynchronen Scripten bei längerfristigeren Scripten, die nicht im direkten Kontext zur aktuellen Arbeit stehen, wie dem Ansteuern anderer Applikationen.

Damit ein Script asynchron ausgeführt wird, muss es in der ersten Zeile die folgende Anweisung enthalten:

```
<!--#pragma asynchronous-->
```

(Längere) asynchrone Scripte sollten eine Abbruchsbedingung vorsehen und hierzu regelmäßig den Wert der Terminate Eigenschaft abfragen.

Bitte beachten Sie dabei, dass die noch laufenden, asynchronen Scripte insgesamt 4 Sekunden Zeit haben, sich zu beenden, bevor die Beendigung erzwungen wird. Sleeps und andere Aktionen im Script sollten also keinesfalls länger als 3 Sekunden dauern, bevor wieder die .Terminate Eigenschaft geprüft und das Script dann ggf. sich so schnell wie möglich kontrolliert beenden muss.

> Wichtig: Bei C# Scripten kann die Beendigung nicht erzwungen werden, es ist daher umso wichtiger die .Terminate Eigenschaft rechtzeitig zu prüfen und das Script kontrolliert zu beenden. Sollte sich das Script nicht innerhalb von 4 Sekunden beendet haben, kann es zu Programmfehlern kommen!

1.4.3 Scripte und der Edit-Modus

Soll ein Script bei Klicken einer Schaltfläche in der Eingabemaske ohne Speichern-Frage gestartet werden, muss folgende Anweisung enthalten sein:

```
<!--#pragma keepeditmode-->
```

Auf diese Weise wird die Speichern-Frage unterdrückt und Sie bleiben im Edit-Modus.

1.4.4 Verschlüsselung von Scripten

Wählen Sie Extras > Script > Verschlüsseln um den Scriptcode zu verschlüsseln und den Inhalt zu schützen. Wenn Sie beim Verschlüsseln als Zieldatei die Quelldatei angegeben, wird diese überschrieben. Ein Entschlüsseln der verschlüsselten Scriptdatei ist nicht mehr möglich.

Enthält ein Script die Anweisung

```
<!--#pragma encrypted-->
```

so wird es ab dieser Stelle bis zum Ende komplett verschlüsselt. Damit kann ein Teil des Scriptes bearbeitbar bleiben, damit bestimmte Parameter oder Variablen veränderbar sind, der Rest des Scriptes bleibt jedoch geschützt.

1.4.5 Auswahl der Scriptsprache

Über den Befehl

```
<!--#language="[Scriptsprache]"-->
```

kann die Scriptsprache explizit gesetzt werden. Das ist immer dann nötig, wenn ein Script nicht als Datei ausgeführt und die Sprache anhand der Dateiendung ermittelt werden kann (z.B. beim Ausführen als Scriptzeilen in der Eingabemaske)

Mögliche Werte sind u.a. "VBScript", "JScript", "C#", "CSharp" oder "C#Script".

1.4.6 Protokollierung [C#]

Diese Option steht nur bei C# Scripten zur Verfügung.

Um die Protokollierung eines Scriptes zu aktivieren, muss folgende Anweisung enthalten sein:

```
<!--#pragma forcelogging-->
```

Die Protokollausgaben erfolgen in die Datei "%temp%\combitCSharpScript.log".

1.4.7 Debugmodus [C#]

Diese Option steht nur bei C# Scripten zur Verfügung.

Enthält ein Script die Anweisung

```
<!--#pragma debugmode-->
```

dann wird zum einen zu Beginn des Scripts das Starten eines Debuggers ausgelöst (sofern auf Ihrem System ein solcher installiert ist), mit dem Sie das Script Schritt für Schritt auf Fehler überprüfen können und zum anderen enthalten durch Ausnahmen ausgelöste Fehlertexte mehr Informationen und Zeilennummern.

1.4.8 Hinzufügen von Referenzen [C#]

Diese Option steht nur bei C# Scripten zur Verfügung.

Über den Befehl

```
<!--#include ref="[Pfad einer .dll]"-->
```

können einem Script externe Referenzen / Komponenten, wie z.B. die Windows Forms Bibliothek von Microsoft (System.Windows.Forms.dll), hinzugefügt werden.

Geben Sie keinen Pfad sondern nur einen Dateinamen an, so wird versucht die Referenz aus dem "Global Assembly Cache" zu laden. Bei Angabe eines kompletten Pfads wird eine Kopie der Datei in einem temporären Ordner angelegt und von dort geladen.

> Hinweis: Dieser Befehl muss im Script nach den für alle Scriptsprachen geltenden Präprozessor-Direktiven erfolgen.

1.4.9 Hinzufügen von Namespaces [C#]

Diese Option steht nur bei C# Scripten zur Verfügung.

Über den Befehl

```
<!--#include using="[Namespace]"-->
```

können using-Anweisungen zum Script hinzugefügt werden. Dies hat den Vorteil, dass Namespace-Namen nicht immer explizit angegeben werden müssen.

Anstatt des Aufrufs "System.Windows.Forms.MessageBox.Show(…);" für jede einzelne MessageBox würde der Aufruf nach dem Hinzufügen von "System.Windows.Forms" als using-Anweisung nur noch "MessageBox.Show(…);" lauten.

> Hinweis: Dieser Befehl muss im Script nach den für alle Scriptsprachen geltenden Präprozessor-Direktiven erfolgen.

1.4.10 Benutzerrechte

Es gibt Benutzerrechte, die festlegen, ob ein Benutzer Scripts grundsätzlich ausführen darf. Mehr dazu finden Sie im Kapitel "Benutzer- und Rechteverwaltung" des Handbuchs.

1.4.11 Zugriff von außen

Mit Ausnahme des WScript Objektes steht Ihnen auch außerhalb von Scripten der Leistungsumfang des Objektmodells zur Verfügung. Sie können so beispielsweise eine Anwendung schreiben, die Ihre Daten aus der combit Anwendung bezieht und diese in einer beliebigen Weise bearbeitet, um sie anschließend zurück zu schreiben.

Sie benötigen zum Zugriff lediglich eine Instanz des Application Objektes. Die sogenannte ProgID des Application Objektes lautet "cRM.Application".

Die Instanzierung erfolgt in der Regel über einen CreateObject Befehl (Late Binding). Die zur Verfügung stehenden Möglichkeiten sowie die genaue Syntax hängen von der verwendeten Programmiersprache ab. Entsprechende Hinweise finden Sie in der jeweiligen Dokumentation.

> **Hinweis:** C# Scripte können momentan ausschließlich intern benutzt werden.

1.4.12 Ereignisse für Scripte

Über die Ereignisverwaltung kann auf verschiedene Ereignisse mit automatischen Aktionen reagiert werden. Hierzu können für jede der angebotenen Ereignisse Scripte hinterlegt werden. Weitere Informationen finden Sie im Abschnitt "Ereignis Verwaltung" im Kapitel "Workflows und Ereignisse" des Handbuchs.

Beispiel VBScript: Ereignis 'Datensatz wird gespeichert' der Ansicht Firma, um dem Anwender eine Hinweismeldung anzuzeigen, dass er für den aktuellen Datensatz nur Leserechte besitzt.

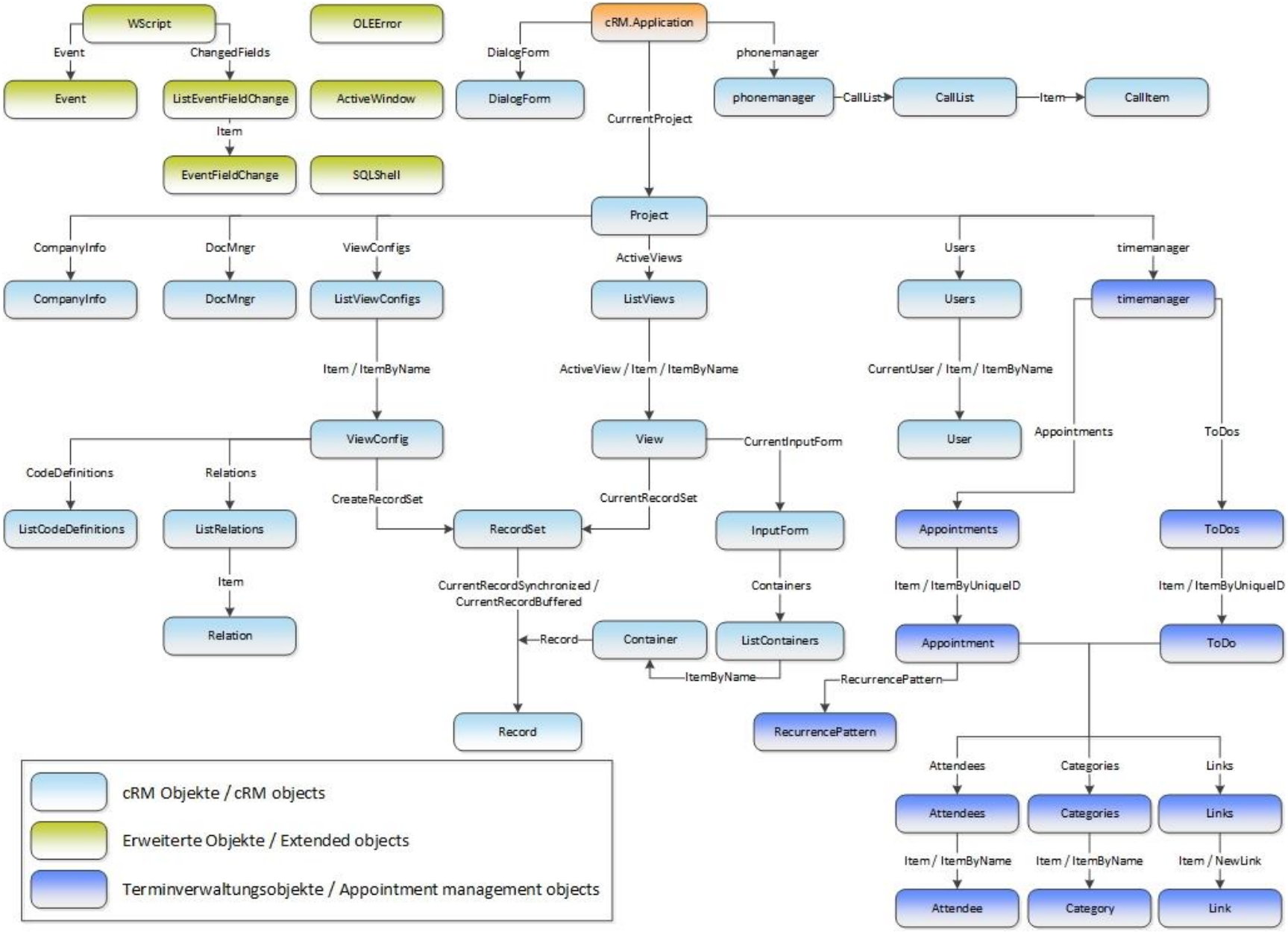
```
Dim oEvent

Set oEvent = WScript.Event

Dim DoCancel

DoCancel = false

Dim Value

Value = oEvent.Record1.GetContentsByName("ABC")

If(Value = "A") Then

    DoCancel = True

    MsgBox "Sie haben nur Leserechte auf diesen Datensatz!", vbInformation,
    "combit Anwendung"

End If

oEvent.Cancel = DoCancel
```

Beispiel C# Script:

```
object event = WScript.Event;
```

```
bool doCancel = false;

string value = event.Record1.GetContentsByName("ABC");

if (value == "A")

{

    doCancel = true;

    MessageBox.Show("Sie haben nur Leserechte auf diesen Datensatz!",
    cRM.AppTitle, MessageBoxButtons.OK);

}

event.Cancel = doCancel;
```

2 Objekthierarchie



3 Objekt-Referenz

3.1 ActiveWindow Objekt

Liefert Informationen zum aktuellen, also aktiven, Windows-Fenster.

3.1.1 Eigenschaften

Class, read-only

Beschreibung:

Gibt den Klassennamen des aktiven Fensters zurück.

Typ:

String

Handle, read-only

Beschreibung:

Gibt das Handle (hWnd) des aktiven Fensters zurück.

Typ:

Long

Title, read-only

Beschreibung:

Gibt den Titel/Name des aktiven Fensters zurück.

Typ:

String

3.2 Application/cRM Objekt

Bei Scripten, die innerhalb der combit Anwendung ausgeführt werden, steht Ihnen direkt das Application-Objekt als cRM-Objekt zur Verfügung. Beim Zugriff von außen benötigen Sie lediglich eine Instanz des Application-Objektes. Die sogenannte ProgID des Application-Objektes lautet "cRM.Application". Die Instanzierung erfolgt in der Regel über einen CreateObject-Befehl (Late Binding). Die zur Verfügung stehenden Möglichkeiten sowie die genaue Syntax hängen von der verwendeten Programmiersprache ab.

Die Verwendung per VBScript wird im Beispiel "cRM.Application Instanz erzeugen" veranschaulicht.

3.2.1 Eigenschaften

AppDir, read-only

Beschreibung:

Liefert den vollständigen Pfad des Hauptverzeichnisses der combit Anwendung, d.h. dort wo sich die Applikationsdateien (EXE und OVL) befinden: z.B. C:\Programme\combit\cRM\.

Typ:

String

AppTitle, read-only

Beschreibung:

Liefert den Namen der combit Anwendung zurück.

Typ:

String

DBServerName, read-only

Beschreibung:

Liefert den konfigurierten Datenbankservernamen zurück.

Typ:

String

DialogForm, read-only

Beschreibung:

Liefert ein Objekt vom Objekt **DialogForm** zurück. Nach dem Erzeugen des Objekts muss die WScript-Funktion **ConnectObject** aufgerufen werden, um auf die Events des Dialog-Objektes (Button-Clicks) reagieren zu können.

Wichtig: Dieses Objekt steht erst ab der cRM Professional Edition zur Verfügung.

Typ:

DialogForm

Beispiel VBScript:

```
Option Explicit

Dim oDlgObj

Set oDlgObj = cRM.DialogForm
```

```
WScript.ConnectObject oDlgObj,"DIALOG_"
```

Beispiel C# Script:

```
DialogForm dlgObj = cRM.DialogForm;
```

Edition, read-only

Beschreibung:

Liefert die Edition der combit Anwendung zurück.

Typ:

Long

Wert	Beschreibung
1	Standard Edition
2	Professional Edition
3	Enterprise Edition

FileVersion, read-only

Beschreibung:

Rückgabe der Dateiversion der installierten combit Anwendung.

Typ:

Long

Wert	Beschreibung
HIWORD	Hauptversion (Dividiert durch 65535)
LOWORD	Nebenversion (Modulo 0xFFFF0000)

Beispiel:

```
Dim FileVersion

FileVersion = (cRM.FileVersion Mod &HFFFF0000)

If Len(FileVersion) = 1 Then

    FileVersion = CStr((cRM.FileVersion / 65535)) & ".00" & CStr(FileVersion)

ElseIf Len(FileVersion) = 2 Then

    FileVersion = CStr((cRM.FileVersion / 65535)) & ".0" & CStr(FileVersion)

ElseIf Len(FileVersion) = 3 Then

    FileVersion = CStr((cRM.FileVersion / 65535)) & "." & CStr(FileVersion)

Else

    FileVersion = CStr("Information über Dateiversion nicht verfügbar.")

End If

MsgBox FileVersion
```

Beispiel C# Script:

```
long fileVersion = (cRM.FileVersion % 0xFFFF0000);

int fileVersionNumberCount = fileVersion.ToString().Length;


if (fileVersionNumberCount == 1)

    MessageBox.Show((fileVersion / 65535).ToString() + ".00" + fileVer-
sion.ToString(), cRM.AppTitle, MessageBoxButtons.OK);

else if (fileVersionNumberCount == 2)

    MessageBox.Show((fileVersion / 65535).ToString() + ".0" + fileVer-
sion.ToString()(), cRM.AppTitle, MessageBoxButtons.OK);

else if (fileVersionNumberCount == 3)

    MessageBox.Show((fileVersion / 65535).ToString() + "." + fileVer-
sion.ToString()(), cRM.AppTitle, MessageBoxButtons.OK);

else

    MessageBox.Show("Information über Dateiversion nicht verfügbar", cRM.AppTitle,
MessageBoxButtons.OK);
```

MainWindowHandle, read-only

Beschreibung:

Liefert das Haupt-Fensterhandle zurück.

Typ:

long

phonemanager, read-only

Beschreibung:

Liefert das phonemanager Objekt vom Typ phonemanager zurück.

Typ:

phonemanager

ProductVersion, read-only

Beschreibung:

Rückgabe der Produkt-Versionsnummer der installierten combit Anwendung.

Typ:

Long

Wert	Beschreibung
HIWORD	Hauptversion (Dividiert durch 65535)
LOWORD	Nebenversion (Modulo 0xFFFF0000)

Beispiel:

```
Dim ProductVersion

ProductVersion = (cRM.ProductVersion Mod &HFFFF0000)

If Len(ProductVersion) = 1 Then

    ProductVersion = CStr((cRM.ProductVersion / 65535)) & ".00" &
CStr(ProductVersion)

ElseIf Len(ProductVersion) = 2 Then

    ProductVersion = CStr((cRM.ProductVersion / 65535)) & ".0" &
CStr(ProductVersion)

ElseIf Len(Version) = 3 Then

    ProductVersion = CStr((cRM.ProductVersion / 65535)) & "." &
CStr(ProductVersion)

Else

    ProductVersion = CStr("Information über Produktversion nicht verfügbar.")

End If

MsgBox ProductVersion
```

Beispiel C# Script:

```
long productVersion = (cRM.ProductVersion % 0xFFFF0000);

int productVersionNumberCount = productVersion.ToString().Length;



if (productVersionNumberCount == 1)

    MessageBox.Show((productVersion / 65535).ToString() + ".00" + productVer-
sion.ToString(), cRM.AppTitle, MessageBoxButtons.OK);

else if (productVersionNumberCount == 2)

    MessageBox.Show((productVersion / 65535).ToString() + ".0" + productVer-
sion.ToString(), cRM.AppTitle, MessageBoxButtons.OK);

else if (productVersionNumberCount == 3)

    MessageBox.Show((productVersion / 65535).ToString() + "." + productVer-
sion.ToString(), cRM.AppTitle, MessageBoxButtons.OK);

else

    MessageBox.Show("Information über Produktversion nicht verfügbar.",
cRM.AppTitle, MessageBoxButtons.OK);
```

TempDir, read-only

Beschreibung:

Rückgabe des Benutzer Temp-Verzeichnisses inkl. Pfad.

Typ:

String

TimemanagerType, read-only

Beschreibung:

Rückgabe der eingestellten Terminverwaltung.

Typ:

Long

Wert	Beschreibung
0	Interne Termin- und Aufgabenverwaltung
1	Microsoft Outlook
2	Tobit David
3	IBM Lotus Notes

TrialVersion, read-only

Beschreibung:

Abfrage, ob die aktuelle Version eine Demoversion ist.

Typ:

Bool

UILanguageID, read-only

Beschreibung:

Liefert die Sprachkonstante (LANGID gemäß Microsoft Windows SDK) der aktuell eingestellten Sprache des Benutzerinterfaces. Der Wert korrespondiert zu den ".lng"-Unterverzeichnissen der Installation.

Typ:

Long

Wert	Beschreibung
7	Deutsch
9	Englisch

Version, read-only

Beschreibung:

Rückgabe der Versionsnummer der installierten combit Anwendung.

Typ:

Long

Wert	Beschreibung
HIWORD	Hauptversion (Dividiert durch 65535), z.B. 2007
LOWORD	Nebenversion (Modulo 0xFFFF0000), z.B. 12

Wichtig: Die Eigenschaft **Version** ist noch aus Gründen der Abwärtskompatibilität enthalten. Für neue Scripte muss die Eigenschaft **FileVersion** verwendet werden.

Beispiel VBScript:

```
Dim Version

Version = (cRM.Version Mod &HFFFF0000)

If Len(Version) = 1 Then

    Version = CStr((cRM.Version / 65535)) & ".00" & CStr(Version)

ElseIf Len(Version) = 2 Then

    Version = CStr((cRM.Version / 65535)) & ".0" & CStr(Version)

ElseIf Len(Version) = 3 Then

    Version = CStr((cRM.Version / 65535)) & "." & CStr(Version)

Else

    Version = CStr("Information über Version nicht verfügbar.")

End If

MsgBox Version
```

Beispiel C# Script:

```
long version = (cRM.ProductVersion % 0xFFFF0000);

int versionNumberCount = version.ToString().Length;



if (versionNumberCount == 1)

    MessageBox.Show((version / 65535).ToString() + ".00" + version.ToString(),
cRM.AppTitle, MessageBoxButtons.OK);

else if (versionNumberCount == 2)

    MessageBox.Show((version / 65535).ToString() + ".0" + version.ToString(),
cRM.AppTitle, MessageBoxButtons.OK);

else if (versionNumberCount == 3)

    MessageBox.Show((version / 65535).ToString() + "." + version.ToString(),
cRM.AppTitle, MessageBoxButtons.OK);

else
```

```
    MessageBox.Show("Information über Version nicht verfügbar.", cRM.AppTitle,
MessageBoxButtons.OK);
```

Visible

Beschreibung:

Anzeigezustand der combit Anwendung. Standardmäßig ist beim Zugriff per COM der Anzeigezustand der combit Anwendung unsichtbar (False).

Typ:

Bool

3.2.2 Methoden

CheckAbortedWaitDlg

Beschreibung:

Die Methode muss zwischen StartWaitDlg und EndWaitDlg aufgerufen werden und gibt zurück, ob der Benutzer die Schaltfläche "Abbrechen" des Wartedialoges betätigt hat, sofern StartWaitDlg mit Abbruch-Schaltfläche angezeigt wurde. Die Methode wird vorzugsweise in Ausführungsschleifen verwendet.

Rückgabewert:

Bool

ConvertHTML2PlainText

Beschreibung:

Konvertiert einen als HTML übergebenen String nach Klartext.

Parameter:

Parametername	Typ	Beschreibung
sHTML	String	Zu konvertierender HTML-String.

Rückgabewert:

String

ConvertLocalToUTCDateTime

Beschreibung:

Gibt das UTC-Datum und -Zeit zurück. Dies wird für die Konvertierung der Termine und Aufgaben benötigt, da eine Speicherung in die Datenbanktabelle (cmbt_tm_appointments = Termine | cmbt_tm_todos = Aufgabe) im UTC-Format erfolgt. Diese Verwendung finden Sie in der Info-Zentrale des mitgelieferten Beispiel-Projektes.

Parameter:

Parametername	Typ	Beschreibung

LocalDateTime	Date	Lokales Datum und Uhrzeit, z.B. dtmpDate = CStr(Date & " 00:00:00") cRM.ConvertLocalToUTCDateTime(dtmpDate)

Rückgabewert:

Date

ConvertPlainText2HTML

Beschreibung:

Konvertiert einen als Klartext übergebenen String nach HTML.

Parameter:

Parametername	Typ	Beschreibung
sPlainText	String	Zu konvertierender Klartext.

Rückgabewert:

String

ConvertUTCToLocalDateTime

Beschreibung:

Gibt das lokale Datum und die Uhrzeit anhand des UTC-Datums zurück.

Parameter:

Parametername	Typ	Beschreibung
UTCDateTime	Date	UTC Datum und Zeit

Rückgabewert:

Date

CreateTempFile

Beschreibung:

Erzeugt im System-Verzeichnis %TEMP% eine temporäre eindeutige Datei ohne Inhalt und gibt deren Dateiname zurück. In diese Datei kann anschließend bspw. über die Methode GetContentsByNameToFile der Inhalt eines Feldes vom Typ "Eingebettete Datei" oder "Eingebettete Grafik" gespeichert werden ohne dass es Probleme mit einem nicht-eindeutigen Dateinamen gibt.

Parameter:

Parametername	Typ	Beschreibung
sPrefix	String	Präfix des Dateinamens
sFileExtension	String	Dateiendung
bAutoRemove	Bool	True: Die Datei wird beim Beenden der combit Anwendung wieder weggeräumt.

		False: Die Datei bleibt beim Beenden der combit Anwendung bestehen.

Beispiel VBScript:

```
Dim sFieldName, sFileExtension, oRecord, sTempFileName

Set oRecord = cRM.CurrentProject.ActiveViews.ActiveView.CurrentRecordSet.
CurrentRecordSynchronized

sFieldName = "Photo"

sFileExtension = ".jpg"

sTempFileName = cRM.CreateTempFile("Photo", sFileExtension, True)

Call oRecord.GetContentsByNameToFile(sFieldName, sTempFileName)
```

Beispiel C# Script:

```
string fieldName = "Photo";

string fileExtension = ".jpg";

string tempFileName = cRM.CreateTempFile("Photo", FileExtension, true);

Record record = cRM.CurrentProject.ActiveViews.ActiveView.CurrentRecordSet.
CurrentRecordSynchronized;

record.GetContentsByNameToFile(fieldName, tempFileName);
```

Rückgabewert:

String

CurrentProject

Beschreibung:

Gibt das geladene Projekt als Objekt vom Typ Project zurück (leer, wenn kein Projekt geladen).

Rückgabewert:

Project

DebugOutput

Beschreibung:

Debugausgabe in Debwin.

Parameter:

Parametername	Typ	Beschreibung
Output	String	Debwin Output

Hinweis: Beachten Sie bitte, dass Debwin vor der der combit Anwendung gestartet werden muss, damit eine Ausgabe erfolgt.

DialogSelectDir

Beschreibung:

Zeigt einen Verzeichnisauswahldialog an und gibt den ausgewählten Verzeichnispfad zurück.

Parameter:

Parametername	Typ	Beschreibung
sTitle	String	Dialogtitel
bOpenFileDlg	Bool	True: Verzeichnis Öffnen False: Verzeichnis Speichern
sInitialPath	String	Initielles Verzeichnis als Voreinstellung für den Dialog oder leer.
dwFlags	Long	Reserviert, muss 0 sein.

Beispiel VBScript:

```
sDir = cRM.DialogSelectDir("Quell-Verzeichnis auswählen", true, "", 0)

sDir = cRM.DialogSelectDir("Ziel-Verzeichnis auswählen", false, "C:\temp", 0)
```

Beispiel C# Script:

```
dir = cRM.DialogSelectDir("Quell-Verzeichnis auswählen", true, "", 0);

dir = cRM.DialogSelectDir("Ziel-Verzeichnis auswählen", false, "C:\temp", 0);
```

Rückgabewert:

String (ausgewählter Verzeichnispfad oder leer bei Benutzerabbruch)

DialogSelectFile

Beschreibung:

Zeigt einen Dateiauswahldialog an und gibt den ausgewählten Dateipfad zurück.

Parameter:

Parametername	Typ	Beschreibung
sTitle	String	Dialogtitel
bOpenFileDlg	Bool	True: Datei öffnen Dialog False: Datei speichern Dialog
sInitialPath	String	Initielles Verzeichnis als Voreinstellung für den Dialog. Kann einen kompletten Dateipfad enthalten oder leer sein.
sFileFilter	String	Kann eine Liste von Dateifiltern enthalten oder leer sein. Format: BeschreibungFilter1|DateiwildcardsFilter1|BeschreibungFilter2|DateiwildcardsFilter2|...||
dwFlags	Long	Kann 0 oder eine Kombination (Addition/verODERung) folgender Zahlenwerte sein (Beschreibung siehe MSDN unter 'OPENFILENAME'): const OFN_FILEMUSTEXIST = 4096 const OFN_CREATEPROMPT = 8192 const OFN_HIDEREADONLY = 4

		const OFN_OVERWRITEPROMPT = 2 const OFN_NOREADONLYRETURN = 32768

Beispiel VBScript:

```
const OFN_FILEMUSTEXIST = 4096

const OFN_CREATEPROMPT = 8192

const OFN_HIDEREADONLY = 4

const OFN_OVERWRITEPROMPT = 2

const OFN_NOREADONLYRETURN = 32768

sFile = cRM.DialogSelectFile("Druckvorlage auswählen", True, "C:\temp",
"Druckvorlagen (*.lbl;*.crd;*.lst)|*.lbl;*.crd;*.lst|Alle Dateien|*.*||",
OFN_FILEMUSTEXIST)

sFile = cRM.DialogSelectFile("Datei speichern unter", False, "C:\temp\neu.txt",
"Textdateien (*.txt)|*.txt|Alle Dateien|*.*||",
OFN_OVERWRITEPROMPT+OFN_HIDEREADONLY+OFN_NOREADONLYRETURN)
```

Beispiel C# Script:

```
long OFN_FILEMUSTEXIST = 4096;

long OFN_CREATEPROMPT = 8192;

long OFN_HIDEREADONLY = 4;

long OFN_OVERWRITEPROMPT = 2;

long OFN_NOREADONLYRETURN = 32768;

file = cRM.DialogSelectFile("Druckvorlage auswählen", true, @"C:\temp", "Druckvor-
lagen (*.lbl;*.crd;*.lst)|*.lbl;*.crd;*.lst|Alle Dateien|*.*||",
OFN_FILEMUSTEXIST);

file = cRM.DialogSelectFile("Datei speichern unter", false, @"C:\temp\neu.txt",
"Textdateien (*.txt)|*.txt|Alle Dateien|*.*||", OFN_OVERWRITEPROMPT +
OFN_HIDEREADONLY + OFN_NOREADONLYRETURN);
```

Rückgabewert:

String (ausgewählter Dateipfad oder leer bei Benutzerabbruch)

EndWaitDlg

Beschreibung:

Blendet einen zuvor mit StartWaitDlg angezeigten Wartedialog aus.

GetcRMByProcessID

Beschreibung:

Liefert ein bestimmtes cRM-Objekt anhand seiner ProcessID zurück.

Parameter:

Parametername	Typ	Beschreibung

nProcessID	DWORD	ProcessID

Rückgabewert:

Application

NULL (falls kein Prozess mit dieser ID existiert)

Beispiel VBScript:

```
Dim oApplication, oProject

' Hole irgendein laufendes cRM-Objekt

Set oApplication = GetObject(, CStr("cRM.Application"))

' Hole damit ein spezifisches cRM-Objekt anhand der ProcessID

Set oApplication = oApplication.GetcRMByProcessID(3984)

Set oProject = oApplication.Login("", "", "")

MsgBox "Projektname: " & oProject.Name

Set oProject = Nothing

Set oApplication = Nothing
```

Beispiel C# Script:

```
cRM.GetcRMByProcessID(3984);

cRM.Login("", "", ""); // Bitte beachten: Es darf noch kein Projekt geladen sein

MessageBox.Show("Projektname: " + cRM.CurrentProject.Name, cRM.AppTitle, Message-
BoxButtons.OK);
```

GetIniProfileString

Beschreibung:

Liest einen Eintrag aus einer INI-Datei aus.

Parameter:

Parametername	Typ	Beschreibung
sSection	String	Name der Sektion in der INI-Datei, in der der zu lesende Eintrag steht.
sProperty	String	Name des zu lesenden Eintrags.
sDefaultValue	String	Der Standardwert, der zurückgegeben wird, wenn kein Eintrag mit dem angegebenen Namen existiert.
sIniFile	String	Pfad zur INI-Datei.

Rückgabewert:

String

InvokeMenu

Beschreibung:

Ruft einen Menüeintrag der combit Anwendung auf. Neben der ID des Menüeintrages wird angegeben, ob das Script solange warten soll, bis der Befehl abgearbeitet wurde (und evtl. Dialoge geschlossen wurden) oder ob das Script direkt weiterlaufen soll. Die Menü-IDs der combit Anwendung finden Sie im Kapitel "Menü-IDs".

Hinweis: Es werden nur Menü-IDs von direkt sichtbaren Menüs unterstützt, d.h. Kontextmenüs können nicht verwendet werden.

Parameter:

Parametername	Typ	Beschreibung
MenuID	Long	Die ID des Menüeintrages.
Synchron	Bool	True: synchrone Ausführung False: asynchrone Ausführung

Rückgabewert:

Bool

Wert	Beschreibung
True	Befehl wurde ausgeführt.
False	Befehl konnte nicht ausgeführt werden, dies kann z.B. der Fall sein, wenn der Befehl derzeit nicht zur Verfügung steht.

Login

Beschreibung:

Öffnet ein Projekt.

Parameter:

Parametername	Typ	Beschreibung
ProjectFilePath	String	Verzeichnispfad der Projektdatei. Übergibt man für das zu ladende Projekt leer, so wird zunächst versucht das bereits geladene Projekt zurückzugeben. Wurde Login aber unmittelbar nach CreateObject mit leer aufgerufen, wird versucht das zuletzt geladene Projekt zu öffnen.
UserName	String	Benutzername. Übergibt man diesen und das Passwort leer, wird Windows Authentifizierung versucht. Schlägt diese fehl, wird der Login-Dialog angezeigt.
Password	String	Passwort des Benutzers

Rückgabewert:

Project

Hinweis: Diese Methode sollte bei C# Scripten momentan nur dann aufgerufen werden, wenn noch kein anderes Projekt geladen ist. Sollte schon ein Projekt geladen sein, so wirft diese Methode eine InvalidOperationException.

SendMail

Beschreibung:

Sendet eine eMail über die MAPI-Schnittstelle, unabhängig von den Record oder RecordSet Objekten.

Parameter:

Parametername	Typ	Beschreibung
EMailAddress	String	Empfänger eMail-Adresse. SendMail unterstützt auch die Auflistung mehrerer Empfänger per Semikolon getrennt. Die Empfänger können dabei einen Prefix ("TO:", "CC:", "BCC:") haben, der bestimmt, um welche Art von Empfänger es sich handelt (Voreinstellung: "TO:", wenn kein Präfix angeben ist).
Subject	String	Betreff der eMail.
Contents	String	Text der eMail.
Files	String	Kann eine Liste von eMail-Anhängen beinhalten. Diese müssen dann per Semikolon separiert übergeben werden. z.B. C:\MyFiles\Report.pdf;C:\Info.doc
ShowDialog	Long	Anzeige eines Bestätigungsdialoges. 0 = Dialog wird nicht angezeigt. 1 = Dialog wird angezeigt.

Rückgabewert:

Bool

Beispiel VBScript:

```
cRM.SendMail(CStr("info@relationship-
manager.net;CC:support@combit.net;BCC:webmaster@combit.net;TO: info@combit.net"),
CStr("Test"), CStr("Mailtext."), CStr(""), 1)
```

Beispiel C# Script:

```
cRM.SendMail("info@relationship-
manager.net;CC:support@combit.net;BCC:webmaster@combit.net;TO:info@combit.net",
"Test", "Mailtext.", "", 1);
```

SetIniProfileString

Beschreibung:

Setzt einen Eintrag in einer INI-Datei.

Parameter:

Parametername	Typ	Beschreibung
sSection	String	Name der Sektion in der INI-Datei, in der der zu setzende Eintrag steht.

sProperty	String	Name des zu setzenden Eintrags.
sValue	String	Der zu setzende Wert.
sIniFile	String	Pfad zur INI-Datei.

SetWaitDlgText

Beschreibung:

Hiermit kann der Text in einem per StartWaitDlg gestarteten Wartedialog nachträglich geändert werden.

Parameter:

Parametername	Typ	Beschreibung
Text	String	Der anzuzeigende Informationstext.

StartWaitDlg

Beschreibung:

Zeigt einen Wartedialog mit einer Fortschrittsanimation sowie einem zu übergebenden Informationstext an. Die Ausführung des Scripts wird während der Anzeige fortgeführt. Die Ausblendung erfolgt mit EndWaitDlg. Über den zweiten Parameter kann bestimmt werden, ob der Dialog eine "Abbrechen"-Schaltfläche enthalten soll. Wenn ja, dann kann mit CheckAbortedWaitDlg geprüft werden, ob die Schaltfläche betätigt wurde.

Parameter:

Parametername	Typ	Beschreibung
Text	String	Der anzuzeigende Informationstext.
Cancel	Bool	Schaltfläche "Abbrechen" anzeigen?

3.3 Appointment Objekt

3.3.1 Eigenschaften

ActionData

Beschreibung:

Enthält die Daten einer Aktion.

Rückgabewert:

String

Siehe auch:

ActionID

ActionID

Beschreibung:

Eindeutige ID der zugeordneten Aktion.

Folgende Werte stehen zur Verfügung:

Wert	Beschreibung
<leer>	< keine Aktion ausgewählt >
TMMAIL	Senden einer eMail Benachrichtigung
TMSHEX	Datei/Dokument ausführen bzw. öffnen
DIAL_CRM	Telefonanruf ausführen

Rückgabewert:

String

ActionType

Beschreibung:

Art der Ausführung der Aktion.

Parameter:

Parametername	Typ	Beschreibung
Type	String	0: Automatisch 1: Manuell in Erinnerungsdialog

Rückgabewert:

String

AllDayEvent

Beschreibung:

Legt fest, ob es sich um einen ganztägigen Termin handelt.

Rückgabewert:

Bool

Attendees, read-only

Beschreibung:

Liefert alle Teilnehmer des Termins als Objekt vom Typ **Attendees** zurück.

Rückgabewert:

Attendees

Body

Beschreibung:

Inhalt/Text des Termins (mit RTF-Formatierung).

Rückgabewert:

String

BodyPlain

Beschreibung:

Inhalt/Text des Termins (ohne RTF-Formatierung).

Rückgabewert:

String

Categories, read-only

Beschreibung:

Liefert alle Kategorien des Termins als Objekt vom Typ **Categories** zurück.

Rückgabewert:

Categories

ChangeDate

Beschreibung:

Letztes Änderungsdatum des Termins.

Rückgabewert:

Date

ChangeUser

Beschreibung:

Letzter Änderungsbenutzer des Termins.

Rückgabewert:

String

Contact

Beschreibung:

Kontakt, mit dem der Termin stattfindet ("Mit").

Rückgabewert:

String

CreationDate

Beschreibung:

Erfassungsdatum des Termins.

Rückgabewert:

Date

CreationUser

Beschreibung:

Erfassungsbenutzer des Termins.

Rückgabewert:

String

End

Beschreibung:

Enddatum und -zeit des Termins.

Rückgabewert:

Date

ExtUserData1

Beschreibung:

Ermöglicht das Hinterlegen von Zusatzinformationen in Form von Zeichenketten (Strings).

Hinweis: Die Zeichenketten können maximal jeweils 250 Zeichen lang werden.

Rückgabewert:

String

ExtUserData2

Beschreibung:

Ermöglicht das Hinterlegen von Zusatzinformationen in Form von Zeichenketten (Strings).

Hinweis: Die Zeichenketten können maximal jeweils 250 Zeichen lang werden.

Rückgabewert:

String

HostDatabase

Beschreibung:

Eindeutige Projekt-ID. Der Aufbau ist folgender:

<ProjektID>|<Ansichtenname>|<Ansichtenfamilienname(optional)>|<Primärschlüsselfeldname>|<Primärschlüsselwert>

Rückgabewert:

String

Beispiel VBScript:

```
Dim sRecordRef

sRecordRef = CStr(cRM.CurrentProject.ID) & "|" & _

CStr(cRM.CurrentProject.ActiveViews.ActiveView.Name) & "|" & _

CStr(oViewConfig.FamilyName) & "|" & _

CStr(oViewConfig.PrimaryKeyFldName) & "|" & _

CStr(oRecord.GetContentsByName(CStr(sPKFieldName)))

Appointment.HostDatabase = sRecordRef
```

Beispiel C# Script:

```
ViewConfig viewConfig =
cRM.CurrentProject.ViewConfigs.ItemByName("<Ansichtsname>");

Record record = viewConfig.CreateRecordSet().CurrentRecordSynchronized;

Appointment appointment = cRM.CurrentProject.TimeManager.Appointments.Item(0);

string recordRef = cRM.CurrentProject.ID + "|" +
cRM.CurrentProject.ActiveViews.ActiveView.Name + "|" + viewConfig.FamilyName + "|"
+ viewConfig.PrimaryKeyFldName + "|" +

record.GetContentsByName(viewConfig.PrimaryKeyFldName);

appointment.HostDatabase = recordRef;
```

Importance

Beschreibung:

Diese Eigenschaft ermöglicht es die Priorität eines Termins zu setzen und abzufragen.

Parameter:

Parametername	Typ	Beschreibung
Priority	Long	Priorität

Rückgabewert:

Long

IsRecurring, read-only

Beschreibung:

Gibt an, ob es sich um einen Serientermin handelt.

Rückgabewert:

Bool

Links, read-only

Beschreibung:

Liefert alle Verknüpfungen des Termins als Objekt vom Typ Links zurück.

Rückgabewert:

Links

Location

Beschreibung:

Name des Ortes, an dem der Termin stattfindet.

Rückgabewert:

String

Private

Beschreibung:

Legt fest, ob der Termin als "Privat" markiert werden soll.

Rückgabewert:

Bool

RecurrencePattern

Beschreibung:

Eigenschaften und Einstellungen eines Serientermins. Gibt ein Objekt vom Typ RecurrencePattern zurück.

Rückgabewert:

RecurrencePattern

Reminder

Beschreibung:

Legt fest, ob an diesen Termin erinnert werden soll.

Rückgabewert:

Bool

ReminderMinutesBeforeStart

Beschreibung:

Gibt die Anzahl der Minuten zurück, die vor der Fälligkeit an einen Termin erinnert werden soll.

Rückgabewert:

Long

Start

Beschreibung:

Startdatum und -zeit des Termins.

Rückgabewert:

Date

Subject

Beschreibung:

Beschreibung des Termins.

Rückgabewert:

String

TimeStamp

Beschreibung:

Zeichenkette die aus dem Änderungsdatum und einem fortlaufender Zahlenwert (als String) besteht. Der Zahlenwert wird bei jeder Datensatzänderung verändert. Durch diesen Wert kann man auch feststellen, wenn ein Datensatz sich an einem Tag mehrfach geändert hat.

Format:YYYYMMDD:XXXX
z.B. 20010219:0020

Rückgabewert:

String

UniqueID, read-only

Beschreibung:

Eindeutige RecordID des Termins.

Rückgabewert:

String

User

Beschreibung:

Eigentümer des Termins.

Rückgabewert:

String

3.3.2 Methoden

ClearRecurrencePattern

Beschreibung:

Kennzeichnet einen Termin als Einzeltermin und setzt IsRecurring auf False.

Display

Beschreibung:

Zeigt einen Dialog zum Termin an.

Parameter:

Parametername	Typ	Beschreibung
bModal	Bool	Display Methode kann sich über einen Parameter gesteuert wahlweise modal verhalten. Dies geht jedoch momentan nur bei Neuanlage. z.B. oAppoint.Display(True)

Rückgabewert:

Bool

ExportAsICal

Beschreibung:

Exportiert den Termin in das iCal-Format.

Parameter:

Parametername	Typ	Beschreibung
sFileName	String	Pfad und Dateiname der zu exportierenden Datei.

Rückgabewert:

Bool

Wert	Beschreibung
True	Datei wurde exportiert.
False	Beim Exportieren der Datei ist ein Fehler aufgetreten.

Remove

Beschreibung:

Löscht den Termin.

Rückgabewert:

Bool

Save

Beschreibung:

Speichert die Einstellungen des Termins.

Rückgabewert:

Bool

Wichtig: Um den neu angelegten Termin in der Ansicht gleich angezeigt zu bekommen, muss UpdateViews nach der Speicherung aufgerufen werden.

3.4 Appointments Objekt

3.4.1 Eigenschaften

Count, read-only

Beschreibung:

Liefert die Anzahl der Einträge in der Sammlung zurück.

Rückgabewert:

Long

3.4.2 Methoden

Add

Beschreibung:

Legt einen neuen Termin an und liefert diesen als Objekt vom Typ Appointment zurück.

Rückgabewert:

Appointment

Item

Beschreibung:

Gibt einen Termin zurück. Es muss die Index-Nummer des Termins übergeben werden. Der Index geht von 1 bis Count.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Index-Nummer.

Rückgabewert:

Appointment

ItemByUniqueID

Beschreibung:

Gibt einen Termin anhand der eindeutigen RecordID zurück.

Parameter:

Parametername	Typ	Beschreibung
UniqueID	String	Eindeutige RecordID des Termins.

Rückgabewert:

Appointment
NULL (wenn der Benutzer kein Recht hat den Termin zu sehen).

Remove

Beschreibung:

Löscht einen Termin. Es muss die Index-Nummer des Termins übergeben werden. Der Index geht von 1 bis Count.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Index-Nummer.

Rückgabewert:

Bool

RemoveAll

Beschreibung:

Löscht alle Termine.

Rückgabewert:

Bool

SetFilter

Beschreibung:

Filtert die Anzeige auf Termine, die bestimmte Eigenschaften erfüllen. Die Filterung kann ausgehend von Datenbank oder Datensatz der Host-Applikation erfolgen oder auf Basis des Benutzers.

Parameter:

Parametername	Typ	Beschreibung
FilterType	TMListFilter Constants	Art der Filterung.
HostDatabase	String	Datenbank (Pfad+Name) .
HostRecordID	Long	Eindeutige Datensatznummer des zugeordneten Datensatzes.
UserName	String	Name des Benutzers.

Konstanten der TMListFilterConstants:

Konstante	Wert	Beschreibung
TM_FILTER_HOSTDB	1	Filterung aller Termine zu einer bestimmten Datenbank, korrespondierend zu der Eigenschaft HostDatabase des Appointment-Objektes.
TM_FILTER_HOSTRECID	2	Filterung aller Termine zu einem bestimmten Datensatz, korrespondierend zu der Eigenschaft HostRecordID des Appointment-Objektes. Impliziert die Verwendung eines Filters auf Datenbank-Ebene, vgl. TM_FILTER_HOSTDB.
TM_FILTER_USER	4	Filterung aller Termine zu einem bestimmten Benutzer.

Rückgabewert:

Bool

3.5 Attendee Objekt

3.5.1 Eigenschaften

Name, read-only

Beschreibung:

Liefert den Namen des Teilnehmers zurück.

Rückgabewert:

String

Type, read-only

Beschreibung:

Liefert den Typ des Teilnehmers zurück.

Rückgabewert:

Long

Wert	Beschreibung
0	Benutzer.
1	Gruppe.
2	Ressource.
-1	Unbekannter Typ.

3.6 Attendees Objekt

3.6.1 Eigenschaften

Count, read-only

Beschreibung:

Liefert die Anzahl der Einträge in der Sammlung zurück.

Rückgabewert:

Long

3.6.2 Methoden

Add

Beschreibung:

Fügt einen neuen Teilnehmer zu einer Aufgabe/einem Termin hinzu.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Login Name des Benutzers/Name der Gruppe in der Benutzerverwaltung bzw. Name der Ressource.

Rückgabewert:

Bool

Item

Beschreibung:

Gibt einen Teilnehmer zurück. Es muss die Index-Nummer des Teilnehmers übergeben werden. Der Index geht von 1 bis Count.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Index-Nummer.

Rückgabewert:

Attendee
NULL (wenn der Teilnehmer nicht existiert)

ItemByName

Beschreibung:

Gibt einen Teilnehmer anhand seines Namens zurück.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Login Name des Benutzers/Name der Gruppe in der Benutzerverwaltung bzw. Name der Ressource.

Rückgabewert:

Attendee
NULL (wenn der Teilnehmer nicht existiert)

Remove

Beschreibung:

Löscht einen Teilnehmer. Es muss die Index-Nummer des Teilnehmers übergeben werden. Der Index geht von 1 bis Count.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Index-Nummer.

Rückgabewert:

Bool

RemoveByName

Beschreibung:

Löscht einen Teilnehmer anhand seines Names.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Login Name des Benutzers/Name der Gruppe in der Benutzerverwaltung bzw. Name der Ressource.

Rückgabewert:

Bool

3.7 CallItem Objekt

3.7.1 Eigenschaften

DialRetriesCount, read-only

Beschreibung:

Die Anzahl der bisherigen Wählversuche.

Rückgabewert:

Long

FirstInfo, read-only

Beschreibung:

Infotext (in der Regel Zeitpunkt) des ersten Wählversuches.

Rückgabewert:

String

Info, read-only

Beschreibung:

Der zugeordnete Beschreibungstext (z.B. Zeitpunkt des eingehenden oder Beschreibung des verknüpften Datensatzes des ausgehenden Anrufes).

Rückgabewert:

String

LastInfo, read-only

Beschreibung:

Infotext (in der Regel Zeitpunkt) des letzten Wählversuches.

Rückgabewert:

String

Number, read-only

Beschreibung:

Die hinterlegte Rufnummer.

Rückgabewert:

String

3.7.2 Methoden

GotoRecord

Beschreibung:

Springt zum korrespondierenden Datensatz der combit Anwendung.

Rückgabewert:

Bool

Wert	Beschreibung
True	Datensatz konnte "angesprungen" werden.
False	Keine Verknüpfung zum Datensatz; Datensatz konnte nicht "angesprungen" werden.

Remove

Beschreibung:

Entfernt den Eintrag aus der Anrufliste.

3.8 CallList Objekt

3.8.1 Eigenschaften

Count, read-only

Beschreibung:

Liefert die Anzahl der Einträge in der Anrufliste zurück.

Rückgabewert:

Long

3.8.2 Methoden

Item

Beschreibung:

Gibt einen Eintrag der Anrufliste zurück. Es muss die Index-Nummer des Eintrages übergeben werden. Der Index geht von 1 bis Count.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Index-Nummer.

Rückgabewert:

CallItem

3.9 Categories Objekt

3.9.1 Eigenschaften

Count, read-only

Beschreibung:

Liefert die Anzahl der Einträge in der Sammlung zurück.

Rückgabewert:

Long

3.9.2 Methoden

Add

Beschreibung:

Fügt eine neue Kategorie zu einer Aufgabe/einem Termin hinzu.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Name der Kategorie.

Rückgabewert:

Bool

Item

Beschreibung:

Gibt eine Kategorie zurück. Es muss die Index-Nummer der Kategorie übergeben werden. Der Index geht von 1 bis Count.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Index-Nummer.

Rückgabewert:

Category
NULL (wenn die Kategorie nicht existiert)

ItemByName

Beschreibung:

Gibt eine Kategorie anhand ihres Namens zurück.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Name der Kategorie.

Rückgabewert:

Category
NULL (wenn die Kategorie nicht existiert)

Remove

Beschreibung:

Löscht eine Kategorie. Es muss die Index-Nummer der Kategorie übergeben werden. Der Index geht von 1 bis Count.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Index-Nummer.

Rückgabewert:

Bool

3.10 Category Objekt

3.10.1 Eigenschaften

ID, read-only

Beschreibung:

Liefert die ID der Kategorie zurück.

Rückgabewert:

String

Name, read-only

Beschreibung:

Liefert den Namen der Kategorie zurück.

Rückgabewert:

String

Symbol, read-only

Beschreibung:

Liefert die ID des Symbols der Kategorie zurück.

Rückgabewert:

Long

3.11 CompanyInfo Objekt

Ermöglicht den Zugriff per COM auf die konfigurierten Firmenstammdaten des Projektes (Menüpunkt: Datei > Information > Firmenstammdaten).

Beispiel VBScript:

```
' Ausgabe der Firma:

MsgBox cRM.CurrentProject.CompanyInfo.Company
```

Beispiel C# Script:

```
// Ausgabe der Firma:

MessageBox.Show(cRM.CurrentProject.CompanyInfo.Company, cRM.AppTitle,
MessageBoxButtons.OK);
```

3.11.1 Eigenschaften

AccountNo, read-only

Beschreibung:

Liefert die Konto-Nr. zurück.

Rückgabewert:

String

Bank, read-only

Beschreibung:

Liefert den Banknamen (Bank) zurück.

Rückgabewert:

String

BankCode, read-only

Beschreibung:

Liefert die Bankleitzahl (BLZ) zurück.

Rückgabewert:

String

City, read-only

Beschreibung:

Liefert den Ort zurück.

Rückgabewert:

String

Company, read-only

Beschreibung:

Liefert den Firmennamen (Firma) zurück.

Rückgabewert:

String

Company2, read-only

Beschreibung:

Liefert den Firmennamen (Firma2) zurück.

Rückgabewert:

String

Company3, read-only

Beschreibung:

Liefert den Firmennamen (Firma3) zurück.

Rückgabewert:

String

Country, read-only

Beschreibung:

Liefert das Land zurück.

Rückgabewert:

String

Email, read-only

Beschreibung:

Liefert die eMail-Adresse (eMail) zurück.

Rückgabewert:

String

Extra1, read-only

Beschreibung:

Liefert Zusatz zurück.

Rückgabewert:

String

Extra2, read-only

Beschreibung:

Liefert Zusatz2 zurück.

Rückgabewert:

String

Fax, read-only

Beschreibung:

Liefert die Telefaxnummer (Telefax) zurück.

Rückgabewert:

String

IBAN, read-only

Beschreibung:

Liefert den IBAN-Code zurück.

Rückgabewert:

String

Internet, read-only

Beschreibung:

Liefert die Internetadresse (Internet) zurück.

Rückgabewert:

String

Logo, read-only

Beschreibung:

Liefert das konfigurierte Firmenlogo (Dateiname) zurück.

Rückgabewert:

String

Phone, read-only

Beschreibung:

Liefert die Telefonnummer (Telefon) zurück.

Rückgabewert:

String

Street, read-only

Beschreibung:

Liefert die Strasse zurück.

Rückgabewert:

String

VatID, read-only

Beschreibung:

Liefert die Steuer-Nr. zurück.

Rückgabewert:

String

ZIP, read-only

Beschreibung:

Liefert den Postleitzahl (PLZ) zurück.

Rückgabewert:

String

3.12 Container Objekt

Bietet Zugriff auf einen Container.

3.12.1 Methoden

CurrentRecord

Beschreibung:

Liefert den aktuellen Datensatz als Objekt vom Typ **Record** zurück.

Rückgabewert:

Record

Beispiel VBScript:

```
Dim oContainers, oContainer

Set oContainers = oInputForm.Containers

Set oContainer = oContainers.ItemByName("ID.Aktivitäten.ContactID#{B3C0768A-5599-
44B5-B4F2-7D31A6C10EC5}")
```

```
Dim oCurrentRecord

Set oCurrentRecord = oContainer.CurrentRecord


Dim sComment

sComment = CStr(oCurrentRecord.GetContentsByName("Comment"))

MsgBox "Kommentar: " & sComment, vbInformation, "combit Anwendung"
```

Beispiel C# Script:

```
long mode = 1;

InputForm inputForm =
cRM.CurrentProject.ActiveViews.ActiveView.CurrentInputForm(mode);

ListContainers containers = inputForm.Containers;

Container container = containers.ItemByName("ID.Aktivitäten.ContactID#{B3C0768A-
5599-44B5-B4F2-7D31A6C10EC5}");

Record currentRecord = container.CurrentRecord;

string comment = currentRecord.GetContentsByName("Comment");

MessageBox.Show("Kommentar: " + comment, cRM.AppTitle, MessageBoxButtons.OK);
```

3.13 DialogForm Objekt

Mit diesem Objekt können Sie eigene Dialoge gestalten und per Script darauf zugreifen.

> Wichtig: Dieses Objekt steht ab der cRM Professional Edition zur Verfügung.

3.13.1 Eigenschaften

DialogTitle

Beschreibung:

Setzt den Dialogtitel oder liefert ihn zurück.

Parameter:

Parametername	Typ	Beschreibung
Title	String	Dialogtitel.

Rückgabewert:

String

DLIPath

Beschreibung:

Setzt den Pfad inkl. Dateiname zur DLI-Eingabemaskendatei oder liefert ihn zurück.

Wichtig: Der Eingabemaskendesigner sollte beim ersten Start des Scripts durch Ausführen der Tastenkombination STRG+J aufgerufen und die DLI-Eingabemaskendatei gestaltet und anschließend gespeichert werden.

Parameter:

Parametername	Typ	Beschreibung
Path	String	Pfad inkl. Dateiname zur DLI-Eingabemaskendatei. Dabei können auch die Variablen %APPDIR% bzw. %PRJDIR% verwendet werden, z.B. "%PRJDIR%\Scripts\Questionnaire.dli".

Rückgabewert:

String

IconPath

Beschreibung:

Setzt oder liefert das Icon für den Dialog.

Parameter:

Parametername	Typ	Beschreibung
Path	String	Pfad inkl. Dateiname zu einer .ico-, .exe- oder .dll-Datei. Dabei können auch die Variablen %APPDIR% bzw. %PRJDIR% verwendet werden, z.B. "%APPDIR%\Scripts\combit.ico". Es werden nur 16x16 Pixel große Icons unterstützt. Bei .exe- und .dll-Dateien wird immer das erste Icon verwendet.

Rückgabewert:

String

HeightInPixel

Beschreibung:

Setzt die Höhe des Dialogs in Pixel oder liefert sie zurück.

Parameter:

Parametername	Typ	Beschreibung
Height	Short	Höhe in Pixel (maximal 32767 Pixel).

Rückgabewert:

Short

Sizable

Beschreibung:

Bestimmt, ob sich die Größe des Dialogs verändern lässt oder nicht bzw. liefert diesen Wert zurück.

Parameter:

Parametername	Typ	Beschreibung
Sizable	Bool	True: Größe veränderbar False: Größe nicht veränderbar

Rückgabewert:

Bool

SizeToContent

Beschreibung:

Bestimmt, ob sich die Größe des Dialogs an die Größe der Eingabemaske anpasst oder nicht bzw. liefert diesen Wert zurück.

Parameter:

Parametername	Typ	Beschreibung
Sizable	Bool	True: Größe wird an Eingabemaske angepasst (Die Eigenschaften HeightInPixel und WidthInPixel werden dabei ignoriert) False: Größe wird nicht an Eingabemaske angepasst

Rückgabewert:

Bool

WidthInPixel

Beschreibung:

Setzt die Breite des Dialogs in Pixel oder liefert sie zurück.

Parameter:

Parametername	Typ	Beschreibung
Width	Short	Breite in Pixel (maximal 32767 Pixel).

Rückgabewert:

Short

3.13.2 Methoden

AddTimer

Beschreibung:

Meldet einen Timer-Event an. Dieser meldet sich in den in Parameter **Seconds** angegebenen Abständen über den Event **TimerEvent**.

Parameter:

Parametername	Typ	Beschreibung
TimerName	String	Name des Timers.
Seconds	Long	Intervall in Sekunden, in denen sich der Event meldet.

BindFctHandler [C#]

Beschreibung:

Diese Methode steht nur bei C# Scripten zur Verfügung.

Bindet eine Methode (delegate) an einen Buttonklick. Der erste Parameter muss hierbei dem ersten Paramenter der Funktion **DefineFct** entsprechen.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Name des Buttons an den die Methode gebunden werden soll.
FctEventHandler	Delegate	Eine Methode mit der Signatur *Methode()* und keinem Rückgabewert (void).

BindTimerHandler [C#]

Beschreibung:

Diese Methode steht nur bei C# Scripten zur Verfügung.

Bindet eine Methode (delegate) an einen Timer. Der erste Parameter muss hierbei dem ersten Paramenter der Funktion **AddTimer** entsprechen.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Name des Timers an den die Methode gebunden werden soll.
TimerEventHandler	Delegate	Eine Methode mit der Signatur *Methode()* und keinem Rückgabewert (void).

Close

Beschreibung:

Schließt den Dialog.

DefineComboboxItems

Beschreibung:

Fügt zu einer in der Eingabemaske bestehenden Combobox Auswahleinträge hinzu.

Parameter:

Parametername	Typ	Beschreibung
FieldName	String	Feldname.
ComboboxItems	String	Liste mit Einträgen. Mehrere Einträge werden dabei mit dem Separator 'vbCr' (Char(13)) getrennt.

Rückgabewert:

Long

Wert	Beschreibung
0	Erfolgreich
1	Fehler

DefineFct

Beschreibung:

Meldet einen Button an.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Anzeigename.
Tooltip	String	Tooltip.
CauseCloseDialog	Bool	True: Der Dialog wird beim Drücken geschlossen (z.B. bei OK oder Abbrechen).
CauseCallback	Bool	True: Der Event ExecuteFct wird beim Drücken aufgerufen.
RetValOnClose	Long	Rückgabewert der Show Methode wenn der Dialog durch diesen Button geschlossen wird.

Hinweis: Wird der Dialog durch einen Button beendet (siehe Parameter CauseCloseDialog), werden, falls die Button-ID nicht vbCancel (2) ist, vor dem Schließen alle Eingaberegeln geprüft und, falls diese erzwungen werden, das Schließen des Dialogs verhindert.

DefineFctStart

Beschreibung:

Initialisiert die Funktionen.

DefineVariable

Beschreibung:

Meldet ein Eingabefeld an.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Anzeigename.
ID	Long	ID.
Type	String	Feldtyp. Mögliche Werte: "C" für Zeichenfelder "N" für Numerische Felder "D" für Datumsfelder "T" für Datum+Zeit Felder "L" für Boolesche Felder "M" für Bemerkungsfelder
Length	Long	Feldlänge.
Precision	Long	Präzision bei Nachkommastellen.
DefaultValue	String	Initieller Wert. Verwenden Sie die folgenden Formate für den DefaultValue: Bei Datumsfeldern: YYYYMMDD Bei Datumsfeldern mit Zeitanteil: YYYYMMDDHHMMSS

DefineVariableStart

Beschreibung:

Initialisiert die Variablen.

GetVariableContents

Beschreibung:

Liefert den aktuellen Inhalt einer Variablen.

Hinweis: Diese Methode sollte in VBScript/JScript nicht verwendet werden, da diese keine Stringreferenzen unterstützen und der zurückgegebene Wert daher entweder fehlerhaft ist oder ein Typenkonflikt entsteht. Verwenden Sie stattdessen die Methode 'GetVariableContentsVariant'.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Anzeigename.
Contents	String (Referenz)	Rückgabewert (Inhalt der Variable).

Rückgabewert:

Long

Wert	Beschreibung
0	Erfolgreich
1	Fehler

GetVariableContentsVariant

Beschreibung:

Liefert den aktuellen Inhalt einer Variablen.

Hinweis: Diese Methode hat die gleiche Funktion wie die Methode 'GetVariableContents', sollte dieser aber bei Verwendung in VBScript/JScript vorgezogen werden, um fehlerhafte Ergebnisse oder Typenkonflikte zu vermeiden.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Anzeigename.
Contents	Variant (Referenz)	Rückgabewert (Inhalt der Variable).

Rückgabewert:

Long

Wert	Beschreibung
0	Erfolgreich
1	Fehler

Layout

Beschreibung:

Öffnen des Dialoges im Eingabemaskendesigner. Dieser kann auch durch Drücken der Tastenkombination STRG+J direkt nach dem Öffnen des Dialogs mit der Methode Show erreicht werden.

SetFocusToField

Beschreibung:

Setzt den Eingabecursor in ein bestimmtes Feld und wechselt in den Bearbeitungsmodus.

Der typische Anwendungsfall ist die Benutzung im Rahmen eines Events. Falls mit dieser Methode der Fokus für die initielle Anzeige gesetzt werden soll, dann muss dies über einen einmalig ausgeführten Timer (siehe Event TimerEvent) realisiert werden, da ein Methodenaufruf vor Show zu früh ist.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Anzeigename.
SelectText	Bool	True: Der bereits bestehende Text des Feldes wird selektiert.

Rückgabewert:

Long

Wert	Beschreibung
0	Erfolgreich

1	Fehler

SetForeground

Beschreibung:

Bringt den Dialog in den Vordergrund.

SetVariableContents

Beschreibung:

Setzt den aktuellen Inhalt einer Variablen.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Anzeigename.
Contents	String	Neuer Inhalt der Variable. Verwenden Sie folgende Formate: Bei Datumsfeldern: YYYYMMDD Bei Datumsfeldern mit Zeitanteil: YYYYMMDDHHMMSS

Rückgabewert:

Long

Wert	Beschreibung
0	Erfolgreich
1	Fehler

Show

Beschreibung:

Stellt den Dialog im normalen Anzeigemodus dar.

Update

Beschreibung:

Aktualisiert die Ansicht der geladenen Eingabemaske. Dies kann nötig werden um bspw. mit der Methode DefineComboboxItems neu hinzugefügte Comboboxeinträge zu aktualisieren.

3.13.3 Events

ExecuteFct

Beschreibung:

Wird aufgerufen, wenn in einem Button-Click der Parameter CauseCallback auf True gesetzt wurde.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Anzeigename des Buttons.

TimerEvent

Beschreibung:

Wird aufgerufen, wenn ein mit der Methode **AddTimer** gemeldeter Event abläuft.

Parameter:

Parametername	Typ	Beschreibung
TimerName	String	Name des Events.

3.13.4 Beispiel VBScript:

```
Option Explicit

Dim oDlgObj

Set oDlgObj = cRM.DialogForm

WScript.ConnectObject oDlgObj,"DIALOG_"


oDlgObj.DialogTitle = "DialogForm"


oDlgObj.DefineVariableStart

oDlgObj.DefineVariable "Var1", 0, "C", 20, "0", ""

oDlgObj.DefineVariable "Var2", 0, "N", 20, "3", ""


oDlgObj.DefineFctStart

oDlgObj.DefineFct "OK", "Bestätigen", True, False, vbOK

oDlgObj.DefineFct "Abbrechen", "Beenden", True, False, vbCancel

oDlgObj.DefineFct "Event", "Event", False, True, -1


oDlgObj.DLIPath = " %PRJDIR%\Test.dli"

oDlgObj.Sizable = True

oDlgObj.SizeToContent = False

oDlgObj.WidthInPixel = 640

oDlgObj.HeightInPixel = 480

oDlgObj.Show


Dim contents
```

```
contents = ""
oDlgObj.GetVariableContentsVariant "Var2", contents
MsgBox "Var2: " & contents
oDlgObj.GetVariableContentsVariant "Var1", contents
MsgBox "Var1: " & contents

' Ereignis, bei dem CauseCallback = true aufgerufen wird
Sub DIALOG_ExecuteFct(sName)
    MsgBox "Event: " & sName
End Sub
MsgBox "Ende"
```

3.13.5 Beispiel C# Script:

```
public static void Main()
{
    long OK = 1;
    long CANCEL = 2;
    long ABORT = 3;
    long RETRY = 4;
    long IGNORE = 5;
    long YES = 6;
    long NO = 7;

    DialogForm dlgObj = cRM.DialogForm;
    dlgObj.DialogTitle = "DialogForm";

    dlgObj.DefineVariableStart();
    dlgObj.DefineVariable("Var1", 0, "C", 20, 0, "");
    dlgObj.DefineVariable("Var2", 0, "N", 20, 3, "");

    dlgObj.DefineFctStart();
    dlgObj.DefineFct("OK", "Bestätigen", true, false, OK);
    dlgObj.BindFctHandler("OK", ButtonOK);
    dlgObj.DefineFct("Abbrechen", "Beenden", true, false, CANCEL);
```

```
    dlgObj.BindFctHandler("Abbrechen", ButtonCancel);

    dlgObj.DefineFct("Event", "Event", false, true, -1);

    dlgObj.BindFctHandler("Event", ButtonEvent);


    dlgObj.DLIPath = @"%PRJDIR%\Test.dli";

    dlgObj.Sizable = true;

    dlgObj.SizeToContent = false;

    dlgObj.WidthInPixel = 640;

    dlgObj.HeightInPixel = 480;

    dlgObj.Show();


    object contents = null;

    dlgObj.GetVariableContentsVariant("Var2", ref contents);

    MessageBox.Show("Var2: " + contents.ToString(), cRM.AppTitle,
    MessageBoxButtons.OK);

    dlgObj.GetVariableContentsVariant("Var1", ref contents);

    MessageBox.Show("Var1: " + contents.ToString(), cRM.AppTitle,
    MessageBoxButtons.OK);

}

public static void ButtonOK()

{

    MessageBox.Show("Die Schaltfläche \"OK\" wurde betätigt.", cRM.AppTitle,
MessageBoxButtons.OK);

}

public static void ButtonCancel()

{

    MessageBox.Show("Die Schaltfläche \"Abbrechen\" wurde betätigt.",
cRM.AppTitle, MessageBoxButtons.OK);

}

public static void ButtonEvent()

{

    MessageBox.Show("Die Schaltfläche \"Event\" wurde betätigt.", cRM.AppTitle,
MessageBoxButtons.OK);

}
```

3.14 DocMngr Objekt

3.14.1 Eigenschaften

LastError, read-only

Beschreibung:

Liefert ein OLEError-Objekt zurück.

Typ:

OLEError

Wert	Beschreibung
32	Schreibzugriff verweigert.
33	Kopieren der Datei fehlgeschlagen.

3.14.2 Methoden

AppendFile

Beschreibung:

Fügt einen neuen relationalen Datensatz mit einer Datei hinzu. Die Datei wird in das erste gefundene Dokumentenfeld der relationalen Ansicht geschrieben. Alle weiteren Felder werden ignoriert.

Dokumentenfelder: (eingebettete) Datei, (eingebettete) Grafik oder DMS Dokument.

Parameter:

Parametername	Typ	Beschreibung
oRecord	Objekt	Datensatz-Objekt, zu dem ein relationaler Datensatz angehängt werden soll, z.B. *Firmen*
oRelation	Objekt	Relations-Objekt, in dem der neue relationale Datensatz mit der angehängten Datei eingefügt wird.
sFile	String	Datei, welche hinzugefügt werden soll (gesamter Pfad + Dateiname).
sDocDescriptionCfg	String	Dateiendung der zu verwendeten Dokumentenkonfiguration oder mit | getrennte Dokumentenkonfigurationsbezeichnung, welche in der Konfiguration der Dokumentenverwaltung angeben wurde, z.B. **.msg* oder **.msg|eMails*

Rückgabewert:

Bool

Wert	Beschreibung
True	Neuer relationaler Datensatz mit der eingefügten Datei konnte geschrieben werden.
False	Neuer relationaler Datensatz konnte nicht erstellt werden.

AppendFileExt

Beschreibung:

Fügt einen neuen relationalen Datensatz mit einer Datei hinzu. Die Datei wird in das erste gefundene Dokumentenfeld der relationalen Ansicht geschrieben. Alle weiteren Felder werden ignoriert.

Dokumentenfelder: (eingebettete) Datei, (eingebettete) Grafik oder DMS Dokument.

Parameter:

Parametername	Typ	Beschreibung
oRecord	Objekt	Datensatz-Objekt, zu dem ein relationaler Datensatz angehängt werden soll, z.B. *Firmen*
oRelation	Objekt	Relations-Objekt, in dem der neue relationale Datensatz mit der angehängten Datei eingefügt wird.
sFile	String	Datei, welche hinzugefügt werden soll (gesamter Pfad + Dateiname).
sDocDescriptionCfg	String	Dateiendung der zu verwendeten Dokumentenkonfiguration oder mit | getrennte Dokumentenkonfigurationsbezeichnung, welche in der Konfiguration der Dokumentenverwaltung angeben wurde, z.B. **.msg* oder **.msg|eMails*

Rückgabewert:

Record

Wert	Beschreibung
Record	Gültiges Record-Objekt. Kann mit `If oObject Is Nothing Then` überprüft werden.
NULL	Ungültiges Record-Objekt. Der relationale Datensatz mit der angefügten Datei konnte nicht erstellt werden.

Wichtig: Das zurückgelieferte **Record**-Objekt muss unbedingt wieder freigeben werden (`Set oObject = Nothing setzen`), um den Speicher freizugeben!

AppendFileExt2

Beschreibung:

Fügt einen neuen relationalen Datensatz mit einer Datei hinzu. Die Datei wird in das erste gefundene Dokumentenfeld der relationalen Ansicht geschrieben. Alle weiteren Felder werden ignoriert.

Dokumentenfelder: (eingebettete) Datei, (eingebettete) Grafik oder DMS Dokument.

Parameter:

Parametername	Typ	Beschreibung
sViewName	String	Name der Ansicht, zu der ein relationaler Datensatz angehängt werden soll, z.B. *Firmen*
sRecordID	String	Datensatz-ID, zu der ein relationaler Datensatz angehängt werden soll.
sRelationToken	String	Relation, zu der ein neuer relationaler Datensatz

		mit der angehängten Datei eingefügt werden soll, z.B. *ID.Aktivitäten.ContactID*
sFile	String	Datei, welche hinzugefügt werden soll (gesamter Pfad + Dateiname).
sDocDescriptionCfg	String	Dateiendung der zu verwendeten Dokumentenkonfiguration oder mit | getrennte Dokumentenkonfigurationsbezeichnung, welche in der Konfiguration der Dokumentenverwaltung angeben wurde, z.B. **.msg* oder **.msg|eMails*

Rückgabewert:

Record

Wert	Beschreibung
Record	Gültiges Record-Objekt. Kann mit `If oObject Is Nothing Then` überprüft werden.
NULL	Ungültiges Record-Objekt. Der relationale Datensatz mit der angefügten Datei konnte nicht erstellt werden.

Wichtig: Das zurückgelieferte Record-Objekt muss unbedingt wieder freigeben werden (`Set oObject = Nothing setzen`), um den Speicher freizugeben!

3.15 InputForm Objekt

Dieses Objekt bietet verschiedene Manipulationsmöglichkeiten direkt in der Eingabemaske in einem Datensatz.

Soll ein Script bei Klicken eines Buttons in der Eingabemaske ohne Speichern-Frage gestartet werden, muss folgende Anweisung enthalten sein:

```
<!--#pragma keepeditmode-->
```

Auf diese Weise wird die Speichern-Frage unterdrückt und Sie bleiben im Edit-Modus.

3.15.1 Eigenschaften

Containers

Beschreibung:

Liefert alle Container der Eingabemaske als Objekt vom Typ ListContainers zurück.

Rückgabewert:

ListContainers

WebElements

Beschreibung:

Liefert alle Web-Elemente der Eingabemaske als Objekt vom Typ ListWebElements zurück.

Rückgabewert:

ListWebElements

3.15.2 Methoden

ConvertToLatestVersion

Beschreibung:

Konvertiert die Eingabemaske (<Ansichtenname>.dli) sowie die etwaige zugehörige WebAccess Eingabemaske (<Ansichtenname>.web.dli) automatisch in das aktuelle Format. Hintergrund ist, dass für die Bereitstellung der Eingabemasken für cRM.WebAccess mittels cRM.WebDeploy diese mindestens in einer Version > 5.000 vorliegen müssen.

Hinweis: Der Anwender muss hierfür Schreibrechte für die DLI-Dateien besitzen.

GetContentsByName

Beschreibung:

Liefert den Inhalt des Feldes als Zeichenkette zurück, dessen physikalischer Feldname übergeben wurde.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Physikalischer Name des gewünschten Feldes in der verwendeten/aktiven Eingabemaske.

Rückgabewert:

String

Hinweis: Der Inhalt von Datumsfeldern wird im Format YYYYMMDD zurückgeben, Datumsfelder mit Zeitanteil werden mit YYYYMMDDHHMMSS formatiert.

GetContentsValueByName

Beschreibung:

Liefert den Inhalt entsprechend des Feldtyps des Feldes zurück, dessen Feldname übergeben wurde, z.B. Datumzeit-Typen als Datumsvariable, numerische Typen als numerische Variable etc. Somit werden bspw. Lokalisierungsprobleme (Komma oder Punkt als Dezimalzeichen? Datumsformatierung?) bei der Weiterverarbeitung des Wertes vermieden.

Hinweis: Wird ein formatiertes Notizenfeld abgerufen, so wird immer der entsprechende HTML-Code zurückgeliefert.

Parameter:

Parametername	Typ	Beschreibung

Name	String	Physikalischer Name des gewünschten Feldes.

Rückgabewert:

Variant

Beispiel:

Hinweis: Bitte beachten Sie, dass "" nicht gleich NULL ist. NULL Werte können wie folgt geprüft werden.

Für ein Beispiel siehe Methode GetContentsValueByName des Record-Objektes.

SetContentsByName

Beschreibung:

Legt den Inhalt des Feldes fest, dessen physikalischer Feldname übergeben wurde.

Hinweis: Wird nur ausgeführt, falls sich das InputForm-Objekt im Bearbeiten-Modus befindet!

Parameter:

Parametername	Typ	Beschreibung
Name	String	Physikalischer Name des gewünschten Feldes in der verwendeten/aktiven Eingabemaske.
Contents	String	(Neuer) Feldinhalt Der Inhalt von Datumsfeldern muss im Format YYYYMMDD übergeben werden, bei Datumsfeldern mit Zeitanteil ist es YYYYMMDDHHMMSS

Rückgabewert:

Bool

SetContentsValueByName

Beschreibung:

Legt den Inhalt des Feldes fest, dessen physikalischer Feldname übergeben wurde.

Hinweis: Wird nur ausgeführt, falls sich das InputForm-Objekt im Bearbeiten-Modus befindet!

Die übergebene Variable für den Inhalt kann dabei einen zum Feldtyp korrespondierenden Typ haben und muss nicht vorher in eine Zeichenkette umgewandelt werden. Somit werden bspw. Lokalisierungsprobleme (Komma oder Punkt als Dezimalzeichen? Datumsformatierung?) bei der Weiterverarbeitung des Wertes vermieden.

Das Setzen eines Primärschlüssels ist möglich, sofern das Schreiben erlaubt ist.

Hinweis: Bitte prüfen Sie den Rückgabewert der Methode, um sicherzustellen, dass das Setzen des neuen Inhalts funktioniert hat.

Wird ein formatiertes Notizenfeld gesetzt, ist folgendes zu beachten: fängt der Inhalt mit \\plaintext: an, dann wird Klartext angenommen, fängt er mit \\html: an, dann wird HTML Inhalt angenommen, wird kein Präfix übergeben, so wird HTML angenommen.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Physikalischer Name des gewünschten Feldes.
Contents	Variant	(Neuer) Feldinhalt

Rückgabewert:

Bool

Beispiel:

Hinweis: Bitte beachten Sie, dass "" nicht gleich NULL ist. NULL Werte können wie folgt geprüft werden.

Für ein Beispiel siehe Methode GetContentsValueByName des Record-Objektes.

SetFocusToField

Beschreibung:

Diese Methode setzt den Eingabecursor in ein bestimmtes Feld.

Hinweis: Wird nur ausgeführt, falls das InputForm-Objekt sich im Bearbeiten-Modus befindet!

Parameter:

Parametername	Typ	Beschreibung
FieldName	String	Name des gewünschten Feldes.
SelectText	Long	Wenn SelectText auf 1 (True) gesetzt wurde, wird der bereits existierende Text in dem Feld auch selektiert.

Rückgabewert:

Bool

ShowField

Beschreibung:

Stellt das übergebene Feld dar, sofern es in der Eingabemaske platziert wurde, ggf. wird hierzu auch eine Karteikarten-Seite umgeschaltet. Falls das Feld mehrfach platziert wurde, wird das erste Vorkommen dargestellt.

Parameter:

Parametername	Typ	Beschreibung
FieldName	String	Name des gewünschten Feldes. Zum Anzeigen von Containern muss der Feldname in der Form 'Schlüsselfeld.Zielansicht.Fremdschlüsselfeld' übergeben werden, also z.B. 'ID.Kontakte.CompanyID' für den Container mit den Kontakten der Firma.

Rückgabewert:

Bool

UpdateAllContainers

Beschreibung:

Führt für alle sichtbaren Container eine Aktualisierung in der Ansicht durch. Der Datensatz der Ansicht selbst wird dabei nicht explizit aktualisiert (siehe View.Update). Der aktuelle Filter und die aktuelle Position innerhalb der Datensätze der Ansicht werden somit nicht verändert.

Einsatzbereich: In einem Script wird ein relationaler Datensatz erzeugt (z.B. Kontaktcontainer-Eintrag), dessen Existenz im Container jetzt durch das Script visuell forciert werden soll, ohne Filter und Position innerhalb der Datensätze der Ansicht zu verändern.

3.16 Link Objekt

3.16.1 Eigenschaften

DisplayText, read-only

Beschreibung:

Liefert den Anzeigenamen der Verknüpfung zurück.

Rückgabewert:

String

PrimaryKeyFldContent, read-only

Beschreibung:

Liefert den Inhalt des Primärschlüsselfelds der Verknüpfung zurück.

Rückgabewert:

String

PrimaryKeyFldName, read-only

Beschreibung:

Liefert den Name des Primärschlüsselfelds der Verknüpfung zurück.

Rückgabewert:

String

Project, read-only

Beschreibung:

Liefert das Projekt der Verknüpfung zurück.

Rückgabewert:

String

ViewName, read-only

Beschreibung:

Liefert den Namen der Ansicht der Verknüpfung zurück.

Rückgabewert:

String

ViewFamily, read-only

Beschreibung:

Liefert den Familiennamen der Ansicht der Verknüpfung zurück.

Rückgabewert:

String

3.16.2 Methoden

GetLinkAsString

Beschreibung:

Liefert die Verknüpfung als Zeichenfolge zurück.

Rückgabewert:

String

SetLinkFromString

Beschreibung:

Setzt einen Link anhand einer Zeichenfolge.

Parameter:

Parametername	Typ	Beschreibung
Link	String	Link-Zeichenfolge. Einzelne Parameter des Links werden wie folgt durch '|' getrennt: <Pro-ectID>|<ViewName>|<ViewFamily>|<PKFldName>|<PKConten t>|<DisplayText>

Beispiel VBScript:

```
Dim oContactRecord, oAppointment, oLink, oViewConfig, sLinkString, sPKField

Set oContactRecord =
crm.CurrentProject.ActiveViews.ActiveView.CurrentRecordSet.CurrentRecordSynchroniz
ed

Set oAppointment = crm.CurrentProject.timemanager.Appointments.Add

Set oLink = oAppointment.Links.NewLink

Set oViewConfig = crm.CurrentProject.ActiveViews.ActiveView.Config

sPKField = oViewConfig.PrimaryKeyFldName

sLinkString = CStr(crm.CurrentProject.ID + "|" +
crm.CurrentProject.ActiveViews.ActiveView.Name + "|" + oViewConfig.FamilyName +
"|" + sPKField + "|" + oContactRecord.GetContentsByName(CStr(sPKField)) + "|"  +
oContactRecord.GetContentsByName("Name") + ", " +
oContactRecord.GetContentsByName("Firstname"))

MsgBox sLinkString

oLink.SetLinkFromString(sLinkString)

oAppointment.Links.Add(oLink)

oAppointment.Display
```

Beispiel C# Script:

```
Record contactRecord =
cRM.CurrentProject.ActiveViews.ActiveView.CurrentRecordSet.CurrentRecordSynchroniz
ed;

Appointment appointment = cRM.CurrentProject.TimeManager.Appointments.Add();

Link link = appointment.Links.NewLink();

ViewConfig viewConfig = cRM.CurrentProject.ActiveViews.ActiveView.Config;

string primaryKeyField = viewConfig.PrimaryKeyFldName;

string linkString = cRM.CurrentProject.ID + "|" +
cRM.CurrentProject.ActiveViews.ActiveView.Name + "|" + viewConfig.FamilyName + "|"
+ primaryKeyField + "|" + contactRecord.GetContentsByName(primaryKeyField) + "|" +
contactRecord.GetContentsByName("Name") + ", " +
contactRecord.GetContentsByName("Firstname");
```

```
MessageBox.Show(linkString, cRM.AppTitle, MessageBoxButtons.OK);

link.SetLinkFromString(linkString);

appointment.Links.Add(link);

appointment.Display(true);
```

3.17 Links Objekt

3.17.1 Eigenschaften

Count, read-only

Beschreibung:

Liefert die Anzahl der Einträge in der Sammlung zurück.

Rückgabewert:

Long

3.17.2 Methoden

Add

Beschreibung:

Fügt eine neue Verknüpfung zu einer Aufgabe/einem Termin hinzu.

Parameter:

Parametername	Typ	Beschreibung
Link	Link	Link Objekt.

Rückgabewert:

Bool

Item

Beschreibung:

Gibt eine Verknüpfung zurück. Es muss die Index-Nummer der Verknüpfung übergeben werden. Der Index geht von 1 bis Count.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Index-Nummer.

Rückgabewert:

Link
NULL (wenn die Verknüpfung nicht existiert)

NewLink

Beschreibung:

Liefert eine neue Verknüpfung vom Typ Link zurück.

Rückgabewert:

Link

Remove

Beschreibung:

Löscht eine Verknüpfung. Es muss die Index-Nummer der Verknüpfung übergeben werden. Der Index geht von 1 bis Count.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Index-Nummer.

Rückgabewert:

Bool

3.18 ListCodeDefinitions Objekt

Collection von Codedefinitionen eines Feldes.

3.18.1 Methoden

Count

Beschreibung:

Liefert die Anzahl der Codedefinitionen eines Feldes.

Rückgabewert:

Integer

Item

Beschreibung:

Liefert die Codebeschreibung für den Eintrag mit dem übergebenen Index zurück. Der Index geht von 1 bis Count.

Parameter:

Parametername	Typ	Beschreibung
Index	Integer	Index-Nummer.

Rückgabewert:

String

Beispiel VBScript:

```
Set oCodeDefinitions =
cRM.CurrentProject.ViewConfigs.ItemByName("Firmen").CodeDefinitions("Category")

For i=1 To oCodeDefinitions.Count

    MsgBox oCodeDefinitions.Item(i)

Next
```

Beispiel C# Script:

```
ListCodeDefinitions codeDefinitions =
cRM.CurrentProject.ViewConfigs.ItemByName("Firmen").CodeDefinitions("Category");

for (int i = 1; i <= codeDefinitions.Count; i++)

{

    MessageBox.Show(codeDefinitions.Item(i).ToString(), cRM.AppTitle, MessageBox-
Buttons.OK);

}
```

3.19 ListContainers Objekt

Liste aller Container einer Ansicht.

3.19.1 Methoden

ItemByName

Beschreibung:

Liefert ein Objekt vom Typ **Container** mit dem angegebenen Namen zurück.

> Beim Erzeugen des **ListContainers**-Objekts ist darauf zu achten, dass dabei **CurrentInputForm(2)** verwendet wird. Damit wird der aktuelle Modus des Containers samt aktuell selektiertem Datensatz beibehalten.

Parameter:

Parametername	Typ	Beschreibung
ContainerName	String	Name des gewünschten Containers. Der Name setzt sich dabei wie folgt zusammen: Vollständige Relation (Achtung! Hier darf kein Relationsalias verwendet werden. Korrekt wäre bspw. *ID.Aktivitäten.ContactID* für die Relation zwischen Aktivitäten und Kontakten) + # + ID des entsprechenden Containers (bspw. {*B3C0768A-5599-44B5-B4F2-7D31A6C10EC5*}). Der korrekte Parameter für den Aktivitätencontainer der Ansicht Kontakte in der mitgelieferten Large Solution sähe dann wie folgt aus: *ID.Aktivitäten.KontaktID#{B3C0768A-5599-44B5-B4F2-7D31A6C10EC5}*

Rückgabewert:

Container

3.20 ListWebElements Objekt

Liste aller Web-Elemente einer Ansicht.

3.20.1 Methoden

ItemByName

Beschreibung:

Liefert ein Objekt vom Typ WebElement mit der angegebenen ID zurück.

Parameter:

Parametername	Typ	Beschreibung
WebElementID	String	ID des gewünschten Web-Elements. Die ID kann dem Eigenschaftsdialog des jeweiligen Web-Elements entnommen werden. Der korrekte Parameter für das Web-Element "Google Maps" der Ansicht WebElemente in der mitgelieferten Large Solution sähe dann wie folgt aus: {*64BBFFD7-EA32-4358-BBFC-744D6A94291D*}

Rückgabewert:

WebElement

3.21 ListRelations Objekt

Liste aller konfigurierten Relationen (Menüpunkt: Ansichteneigenschaft der betreffenden Ansicht > Relationen) einer Ansicht.

3.21.1 Eigenschaften

Count, read-only

Beschreibung:

Liefert die Anzahl der konfigurierten Relationen der Ansicht zurück.

Typ:

Long

3.21.2 Methoden

Item

Beschreibung:

Gibt eine Relation entsprechend dem übergebenen Index zurück. Der Index geht von 1 bis Count.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Index-Nummer.

Rückgabewert:

Relation

ItemByName

Beschreibung:

Gibt eine Relation entsprechend dem übergebenen Namen zurück.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Name der Relation im Format *ID.Aktivitäten.ContactID*.

Rückgabewert:

Relation

3.22 ListViewConfigs Objekt

3.22.1 Eigenschaften

Count, read-only

Beschreibung:

Liefert die Anzahl der konfigurierten Ansichten.

Typ:

Long

3.22.2 Methoden

Item

Beschreibung:

Liefert eine konfigurierte Ansicht als Objekt zurück. Es muss die Index-Nummer der Ansicht übergeben werden. Der Index geht von 1 bis Count.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Index-Nummer.

Rückgabewert:

ViewConfig

ItemByName

Beschreibung:

Liefert eine Ansicht mit dem übergebenen Namen als Objekt zurück.

Parameter:

Parametername	Typ	Beschreibung
ViewName	String	Name der Ansicht.

Rückgabewert:

ViewConfig

3.23 ListViews Objekt

3.23.1 Eigenschaften

Count, read-only

Beschreibung:

Liefert die Anzahl der aktiven Ansichten.

Typ:

Long

3.23.2 Methoden

ActiveView

Beschreibung:

Liefert ein Objekt vom Typ View, der aktiven (offen und im Vordergrund befindlichen) Ansicht in der combit Anwendung.

Rückgabewert:

View

CloseActiveView

Beschreibung:

Schließt eine aktive Ansicht.

Rückgabewert:

Bool

Wichtig: Ruft man CloseActiveView auf, während die aktive Ansicht die Ansicht ist, in der das Script läuft, kann dies zu Problemen führen. Dies sollte daher vermeiden werden.

CloseView

Beschreibung:

Schließt eine geöffnete (konfigurierte) Ansicht. Es muss die Index-Nummer der Ansicht übergeben werden. Der Index geht von 1 bis Count.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Index-Nummer.

Rückgabewert:

Bool

> **Wichtig**: Ruft man CloseView mit dem Index der aktuellen Ansicht auf (in der das Script läuft), kann dies zu Problemen führen. Dies sollte daher vermeiden werden.

Item

Beschreibung:

Liefert eine aktive Ansicht als Objekt zurück. Es muss die Index-Nummer der Ansicht übergeben werden. Der Index geht von 1 bis Count.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Index-Nummer.

Rückgabewert:

View

ItemByName

Beschreibung:

Liefert eine Ansicht mit dem übergebenen Namen als Objekt zurück.

Parameter:

Parametername	Typ	Beschreibung
ViewName	String	Name der Ansicht

Rückgabewert:

View

ItemByTagID

Beschreibung:

Liefert eine aktive Ansicht über die TagID (Identifiername) als Objekt zurück.

Parameter:

Parametername	Typ	Beschreibung
TagID	String	Identifiername

Rückgabewert:

View

3.24 OLEError Objekt

Gibt Fehlermeldungen als Fehlercode oder Text zurück.

3.24.1 Eigenschaften

ErrorCode, read-only

Beschreibung:

Gibt den Fehlercode zurück. Welche Fehlercodes das sein können, hängt von den aufrufenden Objekten ab (siehe dort).

Typ:

Long

ErrorText, read-only

Beschreibung:

Gibt die Fehlermeldung als Text zurück.

Typ:

String

NativeErrorCode, read-only

Beschreibung:

Gibt die Fehlermeldung des Datenbankservers, -clients oder des Betriebssystems zurück.

Typ:

Long

3.25 phonemanager Objekt

3.25.1 Eigenschaften

CallList, read-only

Beschreibung:

Liefert ein Objekt vom Typ CallList (s.u.) mit der Anrufliste des phone manager zurück.

Typ:

CallList

3.25.2 Methoden

Activate

Beschreibung:

Aktiviert den phone manager.

Dial

Beschreibung:

Wählt die übergebene Telefonnummer analog zum Kontextmenüpunkt "Anrufen" eines Telefonfeldes.

Parameter:

Parametername	Typ	Beschreibung
Number	String	Die zu wählende Rufnummer.

InvokeMenu

Beschreibung:

Ruft einen Menüeintrag des phone manager auf. Neben der ID des Menüeintrages wird angegeben, ob das Script solange warten soll, bis der Befehl abgearbeitet wurde (und evtl. Dialoge geschlossen wurden) oder ob das Script direkt weiterlaufen soll. Die Menü-IDs des phone manager finden Sie im Kapitel "Menü-IDs".

Hinweis: Es werden nur Menü-IDs von direkt sichtbaren Menüs unterstützt, d.h. Kontextmenüs können nicht verwendet werden.

Parameter:

Parametername	Typ	Beschreibung
MenuID	Long	Die ID des Menüeintrages.
Synchron	Bool	True: synchrone Ausführung False: asynchrone Ausführung

Rückgabewert:

Bool

Wert	Beschreibung
True	Befehl wurde ausgeführt.
False	Befehl konnte nicht ausgeführt werden, dies kann z.B. der Fall sein, wenn der Befehl derzeit nicht zur Verfügung steht.

3.26 Project Objekt

3.26.1 Eigenschaften

ActiveViews, read-only

Beschreibung:

Gibt die aktiven geöffneten Ansichten als Objekt vom Typ ListViews zurück.

> Wichtig: ListViews wird beim Erzeugen initialisiert. Um den aktuellen Stand zu erhalten muss das Objekt neu erzeugt werden.

Typ:

ListViews

CompanyInfo, read-only

Beschreibung:

Liefert die Firmenstammdaten zurück.

Typ:

CompanyInfo

CurrentUser, read-only

Beschreibung:

Liefert den aktiven Benutzer der combit Anwendung.

Typ:

String

DatabaseName, read-only

Beschreibung:

Gibt den Datenbanknamen des Projektes zurück.

Typ:

String

DBSystemType, read-only

Beschreibung:

Liefert das eingestellte Datenbanksystem zurück.

Typ:

Long

Wert	Beschreibung
0	keine Verbindung aktiv
3	Microsoft SQL Server
10	PostgreSQL

DBSystemVersion, read-only

Beschreibung:

Rückgabe der Datenbankserver-Version der installierten combit Anwendung.

Typ:

Long

Wert	Beschreibung
HIWORD	Hauptversion (Dividiert durch 65535)
LOWORD	Nebenversion (Modulo 0xFFFF0000)

Beispiel VBScript:

```
Dim nMajorVersion, nMinorVersion

nMajorVersion = CInt(cRM.CurrentProject.DBSystemVersion / 65535)

nMinorVersion = CInt(cRM.CurrentProject.DBSystemVersion Mod &HFFFF0000)
```

Beispiel C# Script:

```
long majorVersion = (cRM.CurrentProject.DBSystemVersion / 65535);

long minorVersion = (cRM.CurrentProject.DBSystemVersion / 0xFFFF0000);
```

Description, read-only

Beschreibung:

Gibt die Beschreibung des Projektes zurück.

Typ:

String

DocMngr

Beschreibung:

Mit Hilfe des DocMngr-Objekts können Dateien im konfigurierten Dokumentencontainer bzw. Dokumentenfeldern abgelegt werden.

Typ:

DocMngr

FilePath, read-only

Beschreibung:

Gibt den Verzeichnispfad des Projektes zurück.

z.B. C:\Programme\combit\cRM\Solutions\myproject.crm

Typ:

String

ID, read-only

Beschreibung:

Gibt die eindeutige ID des Projektes zurück.

Typ:

String

Name, read-only

Beschreibung:

Gibt den Projektnamen des aktiven Projektes zurück.

Typ:

String

ProjectDir, read-only

Beschreibung:

Gibt das Projektverzeichnis des aktiven Projektes mit abschließendem Backslash zurück.

Typ:

String

Users, read-only

Beschreibung:

Liefert das Users-Objekt für den Zugriff auf die Benutzerstammdaten zurück.

Typ:

Users

timemanager

Beschreibung:

Liefert ein Termin-Objekt vom Typ timemanager zurück.

Typ:

timemanager

ViewConfigs

Beschreibung:

Gibt die konfigurierten Ansichten im aktuellen Projekt als Objekt vom Typ ListViewConfigs zurück.

Typ:

ListViewConfigs

3.26.2 Methoden

CheckVATID

Beschreibung:

Führt online beim Bundeszentralamt für Steuern eine qualifizierte Bestätigungsanfrage einer ausländischen Umsatzsteuer-Identifikationsnummer durch (vgl. §6a Ziff. 3 UstG).

Im Bestätigungsverfahren erhalten anfrageberechtigte Unternehmer vom Bundeszentralamt für Steuern die Auskunft, ob eine ausländische USt-IdNr. gültig ist und ob die von Ihnen mitgeteilten Angaben zu Firmenname, Firmenort, Postleitzahl und Straße mit den in der Unternehmerdatei des jeweiligen EU-Mitgliedstaates registrierten Daten übereinstimmen. (Quelle: http://www.bzst.bund.de/003_menue_links/005_ustidnr/index.html)

Wichtig: Zum Aufruf der Methode wird eine bestehende Internetverbindung vorausgesetzt.

Parameter:

Parametername	Typ	Beschreibung	Pflicht
Parameter für die Abfrage			
VATID_1	String	Ihre deutsche USt-IdNr.	Ja
VATID_2	String	Anzufragende ausländische USt-IdNr.	Ja
Company	String	Name der anzufragenden Firma einschl. Rechtsform	Ja
City	String	Ort der anzufragenden Firma	Ja
ZIP	String	Postleitzahl der anzufragenden Firma	Nein
Street	String	Strasse und Hausnummer der anzufragenden Firma	Nein
Print	Bool	True: Mit schriftlicher amtlicher Bestätigungsmitteilung	Nein

Wichtig: Wird für den Parameter Print True übergeben, wird die amtliche Bestätigungsmitteilung schriftlich per Post an die Adresse Ihrer deutschen USt-IdNr. versendet.

Parametername	Typ	Beschreibung
Parameter, die mit Ergebnissen gefüllt werden		
Date	String	Datum der Anfrage (Format: tt.mm.jjjj)
Time	String	Uhrzeit der Anfrage (Format: hh:mm:ss)
ErrorCode	String	Fehlernummer der Anfrage (Übersicht der ErrorCodes)
ResultName	String	Ergebnis für den angefragten Namen der Firma
ResultCity	String	Ergebnis für den angefragten Ort der Firma
ResultZIP	String	Ergebnis für die angefragte Postleitzahl der Firma
ResultStreet	String	Ergebnis für die angefragte Straße der Firma
ValidFrom	String	Wird nur bei ErrorCode 203 bzw. 204 angegeben. Beginn der Gültigkeit der ausländischen USt-IdNr. (Format: tt.mm.jjjj)
ValidTo	String	Wird nur bei ErrorCode 204 angegeben. Ende der Gültigkeit der ausländischen USt-IdNr. (Format: tt.mm.jjjj)
ConnectionError	String	Enthält eine Fehlermeldung, wenn es zu einem Fehler während der HTTP-Anfrage gekommen ist
ResponseStream	String	Enhält die komplette Antwort, die vom Server zurückgegeben wird

Folgende Werte sind als Ergebnis für die Parameter **ResultName, ResultCity, ResultZIP** und **ResultStreet** möglich:

A = stimmt überein

B = stimmt nicht überein

C = nicht angefragt

D = vom EU-Mitgliedsstaat nicht mitgeteilt

Rückgabewert:

Bool

True: Wird zurückgeliefert, wenn

1. der ErrorCode den Wert 200 hat und die Parameter **Company**, **City**, **ZIP** und **Street** übergeben wurden und die Ergebnisparameter **ResultName**, **ResultCity**, **ResultZIP** und **ResultStreet** den Wert 'A' haben.
2. der ErrorCode den Wert 200 hat und die Ergebnisparameter **ResultName** und **ResultCity** den Wert 'A' haben und **ResultZIP** und **ResultStreet** den Wert 'C' haben, da sie nicht übergeben wurden.

ExecuteInstantReportByDescription

Beschreibung:

Führt einen abgespeicherten Sofortbericht anhand seiner Beschreibung aus.

Wichtig: Sofortberichte, die als Datenbasis einen einzelnen Datensatz haben, können mit dieser Methode nicht exportiert werden. In diesem Fall muss die Methode **ExecuteInstantReportByName** aus dem **RecordSet** Objekt verwendet werden. Bei Sofortberichten, die als Datenbasis "aktueller Filter" haben, werden alle Datensätze gedruckt.

Wenn für den Parameter Media der Wert "PRV" und für den Parameter OutputFileName ein leerer String übergeben wird, wird der Sofortbericht in die Berichtsansicht gedruckt. Wird in OutputFile-

Name aber ein Name und Pfad übergeben, wird ein einfacher Export nach *.ll durchgeführt und das Ergebnis nicht angezeigt.

Parameter:

Parametername	Typ	Beschreibung
Media	String	Ausgabe-Medium. Zur Verfügung stehen: PRN: Drucker PRV: Vorschau PDF: Adobe PDF Format XHTML: XHTML/CSS Format MHTML: Multi-Mime HTML Format XLS: Microsoft Excel Format DOCX: Microsoft Word Format RTF: Rich Text Format (RTF) XPS: Microsoft XPS Format (Erfordert ein auf dem Arbeitsplatz installiertes .NET Framework 3.5 oder neuer) PICTURE_MULTITIFF: Multi-TIFF-Grafik PICTURE_TIFF: TIFF-Grafik PICTURE_PNG: PNG-Grafik PICTURE_PNG_CROP: Automatisch um überflüssigen weißen Rahmen beschnittene PNG-Grafik PICTURE_JPEG: JPG-Grafik PICTURE_JPEG_CROP: Automatisch um überflüssigen weißen Rahmen beschnittene JPG-Grafik PICTURE_BMP: Bitmap-Grafik PICTURE_EMF: Metafile-Grafik (EMF) PICTURE_EMF_CROP: Automatisch um überflüssigen weißen Rahmen beschnittene EMF-Grafik FILE: Datei HTML: HTML-Format JQM: HTML jQuery Mobile Format (nur für Listen) TTY: Nadeldrucker (TTY) TXT: Text (CSV) Format TXT_LAYOUT: Text (Layout) Format XML: XML Format
OutputFileName	String	Evtl. Name und Pfad der zu erzeugenden Ausgabe-Datei bei Ausgabe-Medien wie z.B. "RTF" oder "HTML". In allen anderen Fällen kann ein leerer String übergeben werden. Bei der Erzeugung mehrerer Dateien in der Ausgabe (z.B. Bildexport) kann "%d" als Platzhalter für die Seitennummer verwenden werden.
ReportName	String	Beschreibung des Sofortberichts. Groß- / Kleinschreibung wird beachtet! Die Beschreibung ist dabei die im Feld "Beschreibung" in der Sofortberichte-Konfiguration definierte.
Silent	Bool	Optional. Legt fest, ob der Druck (nach Möglichkeit) ohne Benutzerinteraktion erfolgen soll. Voreinstellung: True

Typ:

Bool

ExecuteInstantReportByName

Beschreibung:

Führt einen abgespeicherten Sofortbericht anhand seines Namens aus.

Wichtig: Sofortberichte, die als Datenbasis einen einzelnen Datensatz haben, können mit dieser Methode nicht exportiert werden. In diesem Fall muss die Methode ExecuteInstantReportByName aus dem RecordSet Objekt verwendet werden. Bei Sofortberichten, die als Datenbasis "aktueller Filter" haben, werden alle Datensätze gedruckt.

Wenn für den Parameter Media der Wert "PRV" und für den Parameter OutputFileName ein leerer String übergeben wird, wird der Sofortbericht in die Berichtsansicht gedruckt. Wird in OutputFileName aber ein Name und Pfad übergeben, wird ein einfacher Export nach *.ll durchgeführt und das Ergebnis nicht angezeigt.

Parameter:

Parametername	Typ	Beschreibung
Media	String	Ausgabe-Medium. Zur Verfügung stehen: PRN: Drucker PRV: Vorschau PDF: Adobe PDF Format XHTML: XHTML/CSS Format MHTML: Multi-Mime HTML Format XLS: Microsoft Excel Format DOCX: Microsoft Word Format RTF: Rich Text Format (RTF) XPS: Microsoft XPS Format (Erfordert ein auf dem Arbeitsplatz installiertes .NET Framework 3.5 oder neuer) PICTURE_MULTITIFF: Multi-TIFF-Grafik PICTURE_TIFF: TIFF-Grafik PICTURE_PNG: PNG-Grafik PICTURE_PNG_CROP: Automatisch um überflüssigen weißen Rahmen beschnittene PNG-Grafik PICTURE_JPEG: JPG-Grafik PICTURE_JPEG_CROP: Automatisch um überflüssigen weißen Rahmen beschnittene JPG-Grafik PICTURE_BMP: Bitmap-Grafik PICTURE_EMF: Metafile-Grafik (EMF) PICTURE_EMF_CROP: Automatisch um überflüssigen weißen Rahmen beschnittene EMF-Grafik FILE: Datei HTML: HTML-Format JQM: HTML jQuery Mobile Format (nur für Listen) TTY: Nadeldrucker (TTY)

		TXT: Text (CSV) Format TXT_LAYOUT: Text (Layout) Format XML: XML Format
OutputFileName	String	Evtl. Name und Pfad der zu erzeugenden Ausgabe-Datei bei Ausgabe-Medien wie z.B. "RTF" oder "HTML". In allen anderen Fällen kann ein leerer String übergeben werden. Bei der Erzeugung mehrerer Dateien in der Ausgabe (z.B. Bildexport) kann "%d" als Platzhalter für die Seitennummer verwenden werden.
ReportName	String	Name des Sofortberichts. Groß- / Kleinschreibung wird beachtet! Der Name ist dabei der im Feld "Name für Scripte/Workflows" in der Sofortberichte-Konfiguration definierte.
Silent	Bool	Optional. Legt fest, ob der Druck (nach Möglichkeit) ohne Benutzerinteraktion erfolgen soll. Voreinstellung: True

Typ:

Bool

ExecuteScriptByCode

Beschreibung:

Führt den übergebenen Scriptcode als internes Script aus.

Parameter:

Parametername	Typ	Beschreibung
sCode	String	Scriptcode

Rückgabewert:

Bool

ExecuteScriptByFilename

Beschreibung:

Führt das übergebene Script aus.

Parameter:

Parametername	Typ	Beschreibung
sFilePath	String	Dateiname inkl. Pfad des Scripts

Rückgabewert:

Bool

ExecuteWorkflowByFilename

Beschreibung:

Führt den übergebenen Workflow aus.

Wichtig: Diese Methode steht erst ab der cRM Professional Edition zur Verfügung.

Parameter:

Parametername	Typ	Beschreibung
sFilePath	String	Dateiname inkl. Pfad des Workflows

Rückgabewert:

Bool

FetchConfigFile

Beschreibung:

Holt die Konfigurationsdatei.

Parameter:

Parametername	Typ	Beschreibung
sSrcFilePath	String	Dateiname in der cmbt_Files Tabelle in der System-Datenbank
sDestFilePath	String	Dateiname, unter dem die Datei abgespeichert werden soll.

Rückgabewert:

Bool

GetGlobalProperty

Beschreibung:

Liest eine globale Script-Einstellung aus, die zuvor mit SetGlobalProperty gesetzt wurde oder gibt den Standardwert zurück.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Der Name der Einstellung.
Default	String	Der Standardwert der Einstellung, der zurückgegeben wird, wenn keine Einstellung mit dem angegebenen Namen existiert.

Rückgabewert:

String

GetSessionProperty

Beschreibung:

Liest eine Session-Variable aus, die zuvor mit SetSessionProperty gesetzt wurde oder gibt den Standardwert zurück.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Der Name der Einstellung.
Default	String	Der Standardwert der Einstellung, der zurückgegeben wird, wenn keine Einstellung mit dem angegebenen Namen existiert.

Rückgabewert:

String

GetUserProperty

Beschreibung:

Liest eine benutzerspezifische Script-Einstellung aus, die zuvor mit SetUserProperty gesetzt wurde oder gibt den Standardwert zurück.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Der Name der Einstellung.
Default	String	Der Standardwert der Einstellung, der zurückgegeben wird, wenn keine Einstellung mit dem angegebenen Namen existiert.

Rückgabewert:

String

OpenActiveViewByName

Beschreibung:

Liefert eine aktive Ansicht über den Ansichtennamen.

Parameter:

Parametername	Typ	Beschreibung
ViewName	String	Name der aktiven Ansicht in der combit Anwendung

Rückgabewert:

View

OpenNewViewByName

Beschreibung:

Öffnet eine neue Ansicht über den Ansichtennamen.

Parameter:

Parametername	Typ	Beschreibung
ViewName	String	Name der neuen Ansicht

Rückgabewert:

View

SetGlobalProperty

Beschreibung:

Setzt eine globale Script-Einstellung. Die Speicherung erfolgt in der Datei "global.ini" in der "cmbt_Files" Tabelle.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Der Name der Einstellung.
String	String	Der (neue) Wert der Einstellung.

SetSessionProperty

Beschreibung:

Setzt eine Session-Variable im List & Label Formeleditor im Unterzweig "cRM.Project.Session".

Für die Verwendung in Datensatzrechten muss dies im "Projekt wurde geöffnet" Ereignis geschehen.

Parameter:

Parametername	Typ	Beschreibung
Property	String	Name der Session-Variable.
Value	String	Inhalt der Session-Variable.

SetUserProperty

Beschreibung:

Setzt eine benutzerspezifische Script-Einstellung. Die Speicherung erfolgt in der Datei "user.ini" in der "cmbt_Files" Tabelle.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Der Name der Einstellung.
String	String	Der (neue) Wert der Einstellung.

StoreConfigFile

Beschreibung:

Speichert die Konfigurationsdatei der combit Anwendung.

Parameter:

Parametername	Typ	Beschreibung
sFilePath	String	Dateiname unter dem die Datei in der cmbt_Files Tabelle in der System-Datenbank abgelegt werden soll.
sLocalFilePath	String	Dateiname der lokalen Datei, die verwendet werden soll.

Rückgabewert:

Bool

3.27 Record Objekt

3.27.1 Methoden

AddToFavorites

Beschreibung:

Fügt den aktuellen Datensatz den Favoriten hinzu.

Wichtig: Diese Methode darf nicht aufgerufen werden, wenn das zugrundeliegende Record Objekt über CurrentRecordBuffered erzeugt wurde.

Rückgabewert:

Bool

AddToHistory

Beschreibung:

Fügt den aktuellen Datensatz dem Verlauf hinzu.

Wichtig: Diese Methode darf nicht aufgerufen werden, wenn das zugrundeliegende Record Objekt über CurrentRecordBuffered erzeugt wurde.

Rückgabewert:

Bool

AddToPhoneManager

Beschreibung:

Fügt den aktuellen Datensatz dem phone manager hinzu.

Parameter:

Parametername	Typ	Beschreibung
Feldname	String	Feldname mit der Nummer, die hinzugefügt werden soll.

Rückgabewert:

Bool

Delete

Beschreibung:

Löscht einen Datensatz.

Rückgabewert:

Bool

DialNumber

Beschreibung:

Wählt die übergebene Telefonnummer und entspricht insofern der manuellen Wahl.

Wichtig: Diese Methode darf nicht aufgerufen werden, wenn das zugrundeliegende Record Objekt über CurrentRecordBuffered erzeugt wurde.

Parameter:

Parametername	Typ	Beschreibung
Feldname	String	Feldname mit der zu wählenden Nummer.

Hinweis: Enthält der Parameter 'Feldname' als erstes Zeichen keinen Buchstaben, so wird der Inhalt als direkt übergebene Telefonnummer interpretiert. Dies ermöglicht Datensatz-spezifische (notwendig für z.B. Autoprotokolle) Telefonwahl, ohne dass die Telefonnummer selbst in einem Datensatz-Feld stehen muss.

GetContentsByFormula

Beschreibung:

Liefert das Ergebnis einer List & Label Formel auf Basis des aktuellen Datensatzes zurück. Die Felder des Datensatzes stehen dabei als Variablen zur Verfügung.

Hinweis: Wird ein formatiertes Notizenfeld abgerufen, so wird immer der entsprechende HTML-Code zurückgeliefert.

Parameter:

Parametername	Typ	Beschreibung
Formula	String	Auszuwertende List & Label Formel. Weitere Informationen finden Sie in der List & Label Hilfe.

Hinweis: Für Feldnamen, die in einer List & Label Formel verwendet werden, muss der physikalische Feldname verwendet werden. Das Verwenden von Feldaliasen wird ausdrücklich nicht empfohlen, da eine Änderung des Feldalias (ggf. nur indirekt, z.B. durch die Änderung eines 1:1 Relationsalias) dazu führt, dass das Script nicht mehr korrekt funktioniert.

Tipp: Eine funktionierende Formel lässt sich bspw. im Eingabemaskendesigner bei der Tooltip-Eigenschaft eines Eingabefelds zusammenstellen. Geben Sie die gewünschte Formel ein und bestätigen Sie den Formel-Dialog mit "OK". Kopieren Sie anschließend die bereits automatisch auf physikalische Feldnamen umgestellte Formel und verwenden diese im Script.

Rückgabewert:

String

GetContentsByName

Beschreibung:

Liefert den Inhalt des Feldes als Zeichenkette zurück, dessen physikalischer Feldname übergeben wurde.

Hinweis: Wird ein formatiertes Notizenfeld abgerufen, so wird immer der entsprechende HTML-Code zurückgeliefert.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Physikalischer Name des gewünschten Feldes.

Rückgabewert:

String

GetContentsByNameToFile

Beschreibung:

Das angegebene BLOB-Feld wird ausgelesen und dessen Inhalt unter dem angegebenen Dateinamen gespeichert.

Parameter:

Parametername	Typ	Beschreibung
Feldname	String	Physikalischer Name des gewünschten (BLOB)-Feldes.
Path	String	Speicherpfad der angelegten Datei.

Rückgabewert:

Bool

GetContentsValueByFormula

Beschreibung:

Liefert das Ergebnis einer List & Label Formel auf Basis des aktuellen Datensatzes entsprechend des von List & Label zurückgelieferten Datentyps zurück. Die Felder des Datensatzes stehen dabei als Variablen zur Verfügung.

Hinweis: Wird ein formatiertes Notizenfeld abgerufen, so wird immer der entsprechende HTML-Code zurückgeliefert.

Parameter:

Parametername	Typ	Beschreibung
Formula	String	Auszuwertende List & Label Formel. Weitere Informationen finden Sie in der List & Label Hilfe.

Hinweis: Für Feldnamen, die in einer List & Label Formel verwendet werden, muss der physikalische Feldname verwendet werden. Das Verwenden von Feldaliasen wird ausdrücklich nicht empfohlen, da eine Änderung des Feldalias (ggf. nur indirekt, z.B. durch die Änderung eines 1:1 Relationsalias) dazu führt, dass das Script nicht mehr korrekt funktioniert.

Tipp: Eine funktionierende Formel lässt sich bspw. im Eingabemaskendesigner bei der Tooltip-Eigenschaft eines Eingabefelds zusammenstellen. Geben Sie die gewünschte Formel ein und bestätigen Sie den Formel-Dialog mit "OK". Kopieren Sie anschließend die bereits automatisch auf physikalische Feldnamen umgestellte Formel und verwenden diese im Script.

Rückgabewert:

Variant

Beispiel:

Hinweis: Bitte beachten Sie, dass "" nicht gleich NULL ist. NULL Werte können wie folgt geprüft werden.

Für ein Beispiel siehe Methode GetContentsValueByName.

GetContentsValueByName

Beschreibung:

Liefert den Inhalt entsprechend des Feldtyps des Feldes zurück, dessen Feldname übergeben wurde, z.B. Datumzeit-Typen als Datumsvariable, numerische Typen als numerische Variable etc. Somit werden bspw. Lokalisierungsprobleme (Komma oder Punkt als Dezimalzeichen? Datumsformatierung?) bei der Weiterverarbeitung des Wertes vermieden.

Hinweis: Wird ein formatiertes Notizenfeld abgerufen, so wird immer der entsprechende HTML-Code zurückgeliefert.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Physikalischer Name des gewünschten Feldes.

Rückgabewert:

Variant

Beispiel VBScript:

Hinweis: Bitte beachten Sie, dass "" nicht gleich NULL ist. NULL Werte können wie folgt geprüft werden.

```
Umsatz =
Company.GetContentsValueByName("ID.SalesUmsatzFirmenLJ.CompanyID.Turnover")

If VarType(Umsatz) = vbNull Then

    MsgBox "Umsatz == NULL"

else

    MsgBox Umsatz

End If

If  Umsatz > 5999 then

    ...
```

Beispiel C# Script:

Hinweis: Bitte beachten Sie, dass "" nicht gleich NULL ist. NULL Werte können wie folgt geprüft werden.

```
Record company =
cRM.CurrentProject.ActiveViews.ActiveView.CurrentRecordSet.CurrentRecordSynchroniz
ed;

var umsatz =
company.GetContentsValueByName("ID.SalesUmsatzFirmenLJ.CompanyID.Turnover");

if (umsatz.GetType() == typeof(long))

{

    if (umsatz == null)

    {

        MessageBox.Show("Umsatz ist NULL", cRM.AppTitle, MessageBoxButtons.OK);

    }

    else
```

```
        {
            MessageBox.Show("Umsatz beträgt: " + umsatz, cRM.AppTitle,
            MessageBoxButtons.OK);
            if (umsatz > 5999)
            {
                // ...
            }
        }
    }
```

Lock

Beschreibung:

Sperrt den aktuellen Datensatz für die Bearbeitung durch andere Benutzer und sollte vor dem Ändern eines Datensatzes durch SetContents... aufgerufen werden. Die Methode liefert im Erfolgsfall True, ansonsten False zurück. Letzteres kann bspw. dadurch bedingt sein, dass der Datensatz bereits gesperrt ist.

Wichtig: Für ein mit der Methode NewRecord erzeugtes Record Objekt dürfen lediglich die Methoden Lock, Get/SetContents... mit einem (einmaligen) abschließenden Save und einem etwaigen (einmaligen) Unlock verwendet werden. Um andere Methoden des Record Objektes verwenden zu können, muss das Record Objekt freigegeben, neu initialisiert und auf den soeben erzeugten Datensatz positioniert werden.

Rückgabewert:

Bool

PrintCard

Beschreibung:

Druckt ein Karteikartenprojekt aus. Neben dem Druckziel muss u.a. der Name des gewünschten Druckprojektes angegeben werden.

Wichtig: Diese Methode darf nicht aufgerufen werden, wenn das zugrundeliegende Record Objekt über CurrentRecordBuffered erzeugt wurde.

Parameter:

Parametername	Typ	Beschreibung
Media	String	Ausgabe-Medium. Zur Verfügung stehen: PRN: Drucker PRV: Vorschau PDF: Adobe PDF Format XHTML: XHTML/CSS Format MHTML: Multi-Mime HTML Format XLS: Microsoft Excel Format DOCX: Microsoft Word Format

		RTF: Rich Text Format (RTF) XPS: Microsoft XPS Format (Erfordert ein auf dem Arbeitsplatz installiertes .NET Framework 3.5 oder neuer) PICTURE_MULTITIFF: Multi-TIFF-Grafik PICTURE_TIFF: TIFF-Grafik PICTURE_PNG: PNG-Grafik PICTURE_PNG_CROP: Automatisch um überflüssigen weißen Rahmen beschnittene PNG-Grafik PICTURE_JPEG: JPG-Grafik PICTURE_JPEG_CROP: Automatisch um überflüssigen weißen Rahmen beschnittene JPG-Grafik PICTURE_BMP: Bitmap-Grafik PICTURE_EMF: Metafile-Grafik (EMF) PICTURE_EMF_CROP: Automatisch um überflüssigen weißen Rahmen beschnittene EMF-Grafik FILE: Datei HTML: HTML-Format JQM: HTML jQuery Mobile Format (nur für Listen) TTY: Nadeldrucker (TTY) TXT: Text (CSV) Format TXT_LAYOUT: Text (Layout) Format XML: XML Format
FileName	String	Dateiname inkl. Pfad des Druckprojektes.
Silent	Bool	Legt fest, ob der Druck (nach Möglichkeit) ohne Benutzerinteraktion erfolgen soll.
OutputFileName	String	Evtl. Name und Pfad der zu erzeugenden Ausgabe-Datei bei Ausgabe-Medien wie z.B. "RTF" oder "HTML". In allen anderen Fällen kann ein leerer String übergeben werden.
UseModalPreviewWindow	Bool	Optional. Legt fest, ob der Druck auf Vorschau (PRV) in das normale Vorschaufenster gedruckt wird (True, Voreinstellung) oder in die Berichtsansicht (False).

Rückgabewert:

Bool

PrintLabel

Beschreibung:

Druckt ein Etikettenprojekt aus. Neben dem Druckziel muss u.a. der Name des gewünschten Druckprojektes angegeben werden.

Wichtig: Diese Methode darf nicht aufgerufen werden, wenn das zugrundeliegende Record Objekt über CurrentRecordBuffered erzeugt wurde.

Parameter:

Parametername	Typ	Beschreibung
Media	String	Ausgabe-Medium. Zur Verfügung stehen: PRN: Drucker PRV: Vorschau PDF: Adobe PDF Format

		XHTML: XHTML/CSS Format MHTML: Multi-Mime HTML Format XLS: Microsoft Excel Format DOCX: Microsoft Word Format RTF: Rich Text Format (RTF) XPS: Microsoft XPS Format (Erfordert ein auf dem Arbeitsplatz installiertes .NET Framework 3.5 oder neuer) PICTURE_MULTITIFF: Multi-TIFF-Grafik PICTURE_TIFF: TIFF-Grafik PICTURE_PNG: PNG-Grafik PICTURE_PNG_CROP: Automatisch um überflüssigen weißen Rahmen beschnittene PNG-Grafik PICTURE_JPEG: JPG-Grafik PICTURE_JPEG_CROP: Automatisch um überflüssigen weißen Rahmen beschnittene JPG-Grafik PICTURE_BMP: Bitmap-Grafik PICTURE_EMF: Metafile-Grafik (EMF) PICTURE_EMF_CROP: Automatisch um überflüssigen weißen Rahmen beschnittene EMF-Grafik FILE: Datei HTML: HTML-Format JQM: HTML jQuery Mobile Format (nur für Listen) TTY: Nadeldrucker (TTY) TXT: Text (CSV) Format TXT_LAYOUT: Text (Layout) Format XML: XML Format
FileName	String	Dateiname inkl. Pfad des Druckprojektes.
Silent	Bool	Legt fest, ob der Druck (nach Möglichkeit) ohne Benutzerinteraktion erfolgen soll.
OutputFileName	String	Evtl. Name und Pfad der zu erzeugenden Ausgabe-Datei bei Ausgabe-Medien wie z.B. "RTF" oder "HTML". In allen anderen Fällen kann ein leerer String übergeben werden.
UseModalPreviewWindow	Bool	Optional. Legt fest, ob der Druck auf Vorschau (PRV) in das normale Vorschaufenster gedruckt wird (True, Voreinstellung) oder in die Berichtsansicht (False).

Rückgabewert:

Bool

PrintReport

Beschreibung:

Druckt ein Listenprojekt aus. Neben dem Druckziel muss u.a. der Name des gewünschten Druckprojektes angegeben werden.

Wichtig: Diese Methode darf nicht aufgerufen werden, wenn das zugrundeliegende Record Objekt über CurrentRecordBuffered erzeugt wurde.

Parameter:

Parametername	Typ	Beschreibung

Media	String	Ausgabe-Medium. Zur Verfügung stehen: PRN: Drucker PRV: Vorschau PDF: Adobe PDF Format XHTML: XHTML/CSS Format MHTML: Multi-Mime HTML Format XLS: Microsoft Excel Format DOCX: Microsoft Word Format RTF: Rich Text Format (RTF) XPS: Microsoft XPS Format (Erfordert ein auf dem Arbeitsplatz installiertes .NET Framework 3.5 oder neuer) PICTURE_MULTITIFF: Multi-TIFF-Grafik PICTURE_TIFF: TIFF-Grafik PICTURE_PNG: PNG-Grafik PICTURE_PNG_CROP: Automatisch um überflüssigen weißen Rahmen beschnittene PNG-Grafik PICTURE_JPEG: JPG-Grafik PICTURE_JPEG_CROP: Automatisch um überflüssigen weißen Rahmen beschnittene JPG-Grafik PICTURE_BMP: Bitmap-Grafik PICTURE_EMF: Metafile-Grafik (EMF) PICTURE_EMF_CROP: Automatisch um überflüssigen weißen Rahmen beschnittene EMF-Grafik FILE: Datei HTML: HTML-Format JQM: HTML jQuery Mobile Format (nur für Listen) TTY: Nadeldrucker (TTY) TXT: Text (CSV) Format TXT_LAYOUT: Text (Layout) Format XML: XML Format
FileName	String	Dateiname inkl. Pfad des Druckprojektes
Silent	Bool	Legt fest, ob der Druck (nach Möglichkeit) ohne Benutzerinteraktion erfolgen soll.
OutputFileName	String	Evtl. Name und Pfad der zu erzeugenden Ausgabe-Datei bei Ausgabe-Medien wie z.B. "RTF" oder "HTML". In allen anderen Fällen kann ein leerer String übergeben werden.
UseModalPreviewWindow	Bool	Optional. Legt fest, ob der Druck auf Vorschau (PRV) in das normale Vorschaufenster gedruckt wird (True, Voreinstellung) oder in die Berichtsansicht (False).

Rückgabewert:

Bool

Save

Beschreibung:

Speichert einen mit Hilfe von SetContents... geänderten Datensatz in die Datenbank; zuvor sollte der Datensatz immer mit Lock gesperrt werden.

Wichtig: Für ein mit der Methode NewRecord erzeugtes Record Objekt dürfen lediglich die Methoden Lock, Get/SetContents... mit einem (einmaligen) abschließenden Save und einem etwaigen (einmaligen) Unlock verwendet werden. Um andere Methoden des Record Objektes verwenden zu können, muss das Record Objekt freigegeben, neu initialisiert und auf den soeben erzeugten Datensatz positioniert werden.

Rückgabewert:

Bool

SaveRecordRef

Beschreibung:

Speichert einen Datensatzverweis.

Parameter:

Parametername	Typ	Beschreibung
FileName	String	Pfad + Dateiname + Dateiendung des zu speichernden Adressverweises.

Rückgabewert:

Bool

Beispiel VBScript:

```
Dim oCurrentRecord

Set oCurrentRecord =
cRM.CurrentProject.ActiveViews.ActiveView.CurrentRecordSet.CurrentRecord

oCurrentRecord.SaveRecordRef("C:\temp\RecordRef.crx")
```

Beispiel C# Script:

```
Record currentRecord =
cRM.CurrentProject.ActiveViews.ActiveView.CurrentRecordSet.CurrentRecordSynchroniz
ed;

currentRecord.SaveRecordRef(@"C:\temp\RecordRef.crx");
```

SendMailDirect

Beschreibung:

Sendet eine eMail mit Anhang entsprechend den Einstellungen unter "Konfigurieren > Allgemein" ohne Benutzer-Interaktion. Die automatische eMail-Ablage wird bei dieser Methode aktiviert. Die eMail wird, sofern konfiguriert, im Kontext des aktuellen Datensatz hinterlegt, bzw. verknüpft.

Wichtig: Diese Methode darf nicht aufgerufen werden, wenn das zugrundeliegende Record Objekt über CurrentRecordBuffered erzeugt wurde.

Parameter:

Parametername	Typ	Beschreibung

EMailAddress	String	Empfänger eMail-Adresse.
Subject	String	Betreff der eMail.
Contents	String	Text der eMail.
Files	String	Kann eine Liste von eMail-Anhängen beinhalten. Diese müssen dann per Semikolon separiert übergeben werden. z.B. C:\MyFiles\Report.pdf;C:\Info.doc

Rückgabewert:

Bool

SendRecordRef

Beschreibung:

Sendet eine eMail mit dem Datensatzverweis des aktuellen Datensatzes.

Wichtig: Diese Methode darf nicht aufgerufen werden, wenn das zugrundeliegende Record Objekt über CurrentRecordBuffered erzeugt wurde.

Parameter:

Parametername	Typ	Beschreibung
EmailAddress	String	Empfänger eMail-Adresse.
Subject	String	Betreff der eMail.
Contents	String	Text der eMail.

Rückgabewert:

Bool

SendSingleMail

Beschreibung:

Sendet eine Einzelmail ohne Dialog. Es wird der vollständige Pfad der Vorlagedatei übergeben.

Wichtig: Diese Methode darf nicht aufgerufen werden, wenn das zugrundeliegende Record Objekt über CurrentRecordBuffered erzeugt wurde.

Parameter:

Parametername	Typ	Beschreibung
TemplatePath	String	Vollständiger Pfad der Vorlagedatei.
Files	String	Kann eine Liste von eMail-Anhängen beinhalten. Diese müssen dann per Semikolon separiert übergeben werden. z.B. C:\MyFiles\Report.pdf;C:\Info.doc

Rückgabewert:

Bool

SendSingleMailDialog

Beschreibung:

Sendet eine Einzelmail mit vorherigem Dialog. Es wird der vollständige Pfad der Vorlagedatei übergeben. Bevor der Anwender die Mail verschickt, kann er noch Änderungen/Ergänzungen vornehmen, die danach aber nicht zwingend in der Vorlage abgespeichert werden müssen.

Wichtig: Diese Methode darf nicht aufgerufen werden, wenn das zugrundeliegende Record Objekt über CurrentRecordBuffered erzeugt wurde.

Parameter:

Parametername	Typ	Beschreibung
TemplatePath	String	Vollständiger Pfad der Vorlagedatei.
Files	String	Optional. Kann eine Liste von eMail-Anhängen beinhalten. Diese müssen dann per Semikolon separiert übergeben werden. z.B. C:\MyFiles\Report.pdf;C:\Info.doc

Rückgabewert:

Bool

SetContentsByFormula

Beschreibung:

Legt den Inhalt des Feldes mit einer List & Label Formel fest, dessen Feldname übergeben wurde. Das Setzen eines Primärschlüssels ist möglich, sofern das Schreiben erlaubt ist.

Hinweis: Bitte prüfen Sie den Rückgabewert der Methode, um sicherzustellen, dass das Setzen des neuen Inhalts funktioniert hat.

Wird ein formatiertes Notizenfeld gesetzt, ist folgendes zu beachten: fängt der Inhalt mit \\plaintext: an, dann wird Klartext angenommen, fängt er mit \\html: an, dann wird HTML Inhalt angenommen, wird kein Präfix übergeben, so wird HTML angenommen.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Physikalischer Name des gewünschten Feldes.
Formula	String	(Neuer) Feldinhalt aus einer List & Label Formel.

Hinweis: Für Feldnamen, die in einer List & Label Formel verwendet werden, muss der physikalische Feldname verwendet werden. Das Verwenden von Feldaliasen wird ausdrücklich nicht

empfohlen, da eine Änderung des Feldalias (ggf. nur indirekt, z.B. durch die Änderung eines 1:1 Relationsalias) dazu führt, dass das Script nicht mehr korrekt funktioniert.

Tipp: Eine funktionierende Formel lässt sich bspw. im Eingabemaskendesigner bei der Tooltip-Eigenschaft eines Eingabefelds zusammenstellen. Geben Sie die gewünschte Formel ein und bestätigen Sie den Formel-Dialog mit "OK". Kopieren Sie anschließend die bereits automatisch auf physikalische Feldnamen umgestellte Formel und verwenden diese im Script.

Rückgabewert:

Bool

SetContentsByName

Beschreibung:

Legt den Inhalt des Feldes fest, dessen physikalischer Feldname übergeben wurde. Das Setzen eines Primärschlüssels ist möglich, sofern das Schreiben erlaubt ist.

Hinweis: Bitte prüfen Sie den Rückgabewert der Methode, um sicherzustellen, dass das Setzen des neuen Inhalts funktioniert hat.

Wird ein formatiertes Notizenfeld gesetzt, ist folgendes zu beachten: fängt der Inhalt mit \\plaintext: an, dann wird Klartext angenommen, fängt er mit \\html: an, dann wird HTML Inhalt angenommen, wird kein Präfix übergeben, so wird HTML angenommen.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Physikalischer Name des gewünschten Feldes.
Contents	String	(Neuer) Feldinhalt

Rückgabewert:

Bool

SetContentsByNameFromFile

Beschreibung:

Die angegebene Datei wird in das übergebene BLOB-Feld eingebettet.

Hinweis: Damit die Datei in der combit Anwendung korrekt angezeigt bzw. verwendet werden kann, muss zusätzlich die Dateiendung, sowie der Dateiname in die entsprechenden Felder (siehe Konfiguration der 'Dokumentinfos' der Ansicht) geschrieben werden.

Parameter:

Parametername	Typ	Beschreibung
FieldName	String	Physikalischer Name des gewünschten (BLOB)-Felds.
FileName	String	Dateiname

Rückgabewert:

Bool

Beispiel VBScript:

```
…
oRecord.Lock
oRecord.SetContentsByNameFromFile("Dokument", "c:\Exceldatei.xls")
oRecord.SetContentsByName "Dateiname", "Exceldatei.xls"
oRecord.SetContentsByName "Typ", "xls"
oRecord.Save
oRecord.Unlock
…
```

Beispiel C# Script:

```
Record record =
cRM.CurrentProject.ActiveViews.ActiveView.CurrentRecordSet.CurrentRecordSynchroniz
ed;
record.Lock();
record.SetContentsByNameFromFile("Dokument", @"C:\temp\Exceldatei.xls");
record.SetContentsByName("Dateiname", "Exceldatei.xls");
record.SetContentsByName("Typ", "xls");
record.Save();
record.Unlock();
```

SetContentsByNameToNull

Beschreibung:

Setzt Feld auf NULL. Bitte beachten Sie, dass die Methode nicht prüft, ob ein Feld in der Datenbank überhaupt NULL sein darf!

Hinweis: Die Methode liefert beim Setzen eines Primärschlüssels False zurück.

Parameter:

Parametername	Typ	Beschreibung
FieldName	String	Physikalischer Name des Feldes.

Rückgabewert:

Bool

SetContentsValueByFormula

Beschreibung:

Legt den Inhalt des Feldes mit einer List & Label Formel fest, dessen Feldname übergeben wurde. Das Setzen eines Primärschlüssels ist möglich, sofern das Schreiben erlaubt ist.

Hinweis: Bitte prüfen Sie den Rückgabewert der Methode, um sicherzustellen, dass das Setzen des neuen Inhalts funktioniert hat.

Wird ein formatiertes Notizenfeld gesetzt, ist folgendes zu beachten: fängt der Inhalt mit \\plaintext: an, dann wird Klartext angenommen, fängt er mit \\html: an, dann wird HTML Inhalt angenommen, wird kein Präfix übergeben, so wird HTML angenommen.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Physikalischer Name des gewünschten Feldes.
Formula	String	(Neuer) Feldinhalt aus einer List & Label Formel.

Hinweis: Für Feldnamen, die in einer List & Label Formel verwendet werden, muss der physikalische Feldname verwendet werden. Das Verwenden von Feldaliasen wird ausdrücklich nicht empfohlen, da eine Änderung des Feldalias (ggf. nur indirekt, z.B. durch die Änderung eines 1:1 Relationsalias) dazu führt, dass das Script nicht mehr korrekt funktioniert.

Tipp: Eine funktionierende Formel lässt sich bspw. im Eingabemaskendesigner bei der Tooltip-Eigenschaft eines Eingabefelds zusammenstellen. Geben Sie die gewünschte Formel ein und bestätigen Sie den Formel-Dialog mit "OK". Kopieren Sie anschließend die bereits automatisch auf physikalische Feldnamen umgestellte Formel und verwenden diese im Script.

Rückgabewert:

Bool

Beispiel:

Hinweis: Bitte beachten Sie, dass "" nicht gleich NULL ist. NULL Werte können wie folgt geprüft werden.

Für ein Beispiel siehe Methode GetContentsValueByName.

SetContentsValueByName

Beschreibung:

Legt den Inhalt des Feldes fest, dessen physikalischer Feldname übergeben wurde.

Die übergebene Variable für den Inhalt kann dabei einen zum Feldtyp korrespondierenden Typ haben und muss nicht vorher in eine Zeichenkette umgewandelt werden. Somit werden bspw. Lokalisierungsprobleme (Komma oder Punkt als Dezimalzeichen? Datumsformatierung?) bei der Weiterverarbeitung des Wertes vermieden.

Das Setzen eines Primärschlüssels ist möglich, sofern das Schreiben erlaubt ist.

Hinweis: Bitte prüfen Sie den Rückgabewert der Methode, um sicherzustellen, dass das Setzen des neuen Inhalts funktioniert hat.

Wird ein formatiertes Notizenfeld gesetzt, ist folgendes zu beachten: fängt der Inhalt mit \\plaintext: an, dann wird Klartext angenommen, fängt er mit \\html: an, dann wird HTML Inhalt angenommen, wird kein Präfix übergeben, so wird HTML angenommen.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Physikalischer Name des gewünschten Feldes.
Contents	Variant	(Neuer) Feldinhalt

Rückgabewert:

Bool

Beispiel:

Hinweis: Bitte beachten Sie, dass "" nicht gleich NULL ist. NULL Werte können wie folgt geprüft werden.

Für ein Beispiel siehe Methode GetContentsValueByName.

TransferData

Beschreibung:

Führt die Einzelübernahme mit einer als Parameter übergebenen Übernahmemaske durch.

Hinweis: Falls es sich um einen nicht-visuellen Record handelt, wird das Autoprotokoll ("Einzelübernahme") nicht ausgelöst.

Parameter:

Parametername	Typ	Beschreibung
TransferTemplate	String	Pfad und Dateiname der Übernahmemaske. Die Platzhalter %APPDIR% und %PRJDIR% werden unterstützt.

Beispiel VBScript:

```
cRM.CurrentProject.ActiveViews.ActiveView.CurrentRecordSet.CurrentRecord.TransferD
ata("%PRJDIR%\Übernahmemasken\Kontakte - Word 2010.xfx")
```

Beispiel C# Script:

```
cRM.CurrentProject.ActiveViews.ActiveView.CurrentRecordSet.CurrentRecordSynchroniz
ed.TransferData(@"%PRJDIR%\Übernahmemasken\Kontakte - Word 2010.xfx");
```

Rückgabewert:

Bool

Unlock

Beschreibung:

Entsperrt einen zuvor mit Lock gesperrten Datensatz wieder und ermöglicht somit die Änderung durch andere Benutzer.

Wichtig: Für ein mit der Methode NewRecord erzeugtes Record Objekt dürfen lediglich die Methoden Lock, Get/SetContents... mit einem (einmaligen) abschließenden Save und einem etwaigen (einmaligen) Unlock verwendet werden. Um andere Methoden des Record Objektes verwenden zu können, muss das Record Objekt freigegeben, neu initialisiert und auf den soeben erzeugten Datensatz positioniert werden.

Rückgabewert:

Bool

3.28 RecordSet Objekt

Wichtig: Bitte beachten Sie, dass wenn das RecordSet-Objekt basierend auf einem ViewConfig-Objekt erzeugt wurde, sämtliche Methodenaufrufe (wie z.B. Druck oder eMail-Versand) keinerlei Autoprotokolle, Auto-Emailablage o.ä. auslösen!

3.28.1 Eigenschaften

RecCount, read-only

Beschreibung:

Gibt die Anzahl der Datensätze in der aktuellen Ansicht / Filter zurück.

Typ:

Long

SortOrder

Beschreibung:

Setzt eine definierte Sortierung über den Index oder gibt die aktuell gesetzte Sortierung zurück.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Index der Sortierung 0 = <ohne Sortierung> Eine absteigende Sortierung kann durch Setzen des entsprechenden negativen Werts erreicht werden.

Rückgabewert:

Long

3.28.2 Methoden

CurrentRecord

Beschreibung:

Liefert den aktuellen Datensatz als Objekt vom Typ **Record** zurück. Wenn z.B. keine Datensätze im **RecordSet** enthalten sind, liefert diese Methode kein **Record** Objekt zurück.

Wichtig: Im Gegensatz zur Methode **CurrentRecordBuffered** aktualisiert sich ein so erzeugter Record immer automatisch, wenn anschließend **Move**...-Befehle für den zugehörigen **RecordSet** aufgerufen werden.

Da das komplette Puffern der Feldinhalte entfällt, ist diese Methode deutlich performanter als **CurrentRecordBuffered** und stellt den empfohlenen Weg dar, um auf Feldinhalte von Datensätzen zuzugreifen.

Mit **CurrentRecord** erzeugte **Record**-Objekte können dadurch jedoch nicht als Variablen für unterschiedliche Datensätze benutzt werden (vgl. Beispiel zu **CurrentRecordBuffered**)! Ist dies erforderlich, so muss anstattdessen **CurrentRecordBuffered** benutzt werden.

Wichtiger Hinweis zu geändertem Verhalten: CurrentRecord verhält sich ab der Version 8 des cRM wie ein **CurrentRecordSynchronized**. (Aus Kompatibilitätsgründen vorher wie ein **CurrentRecordBuffered**). Wir empfehlen die eingesetzten Scripte sorgfältig auf die Verwendung von **CurrentRecord** zur prüfen. Ein typisches Beispiel für die Verwendung ist ein **Record**-Objekt, welches per **CurrentRecord** geholt wird und danach im zugehörigen **RecordSet** eine **Goto**.../**SetFilter**... Methode aufgerufen wird und danach trotzdem auf den zuvor geholten (d.h. gewollt absichtlich "alten") **Record** zugegriffen wird:

```
RecordSet.GotoFirst

Set oRecord = RecordSet.CurrentRecord

oRecord.GetContentsByName "Company"  'liefert Firmenname von Firmendatensatz #1

RecordSet.GotoNext

oRecord.GetContentsByName "Company"  'cRM7: Firmenname (weiterhin) von Firmendatensatz #1 / cRM8:
Firmenname von Firmendatensatz #2
```

Wenn das cRM7 Verhalten der letzten Code-Zeile oben ZWINGEND erforderlich sein sollte, dann schafft Abhilfe entweder die Verwendung eines explizit gepufferten Records, d.h.

```
Set oRecord = RecordSet.CurrentRecordBuffered
```

oder das Setzen eines Kompatibilitätsschalters in der Solution Projekt-Datei (wirkt sich global auf alle **CurrentRecord** Stellen innerhalb der Solution aus). Erstellen Sie dazu in der Solution Projekt-Datei eine Untersektion namens "ExtendedSettings" und setzen darin die Option "COMCurrentRecordAlwaysAsBuffered" auf 1:

...

<!-- DATA -->

```
<profile>

<list name="">

  <list name="ExtendedSettings">

   <item name="COMCurrentRecordAlwaysAsBuffered">1</item>

  </list>

...
```

Beachten Sie dabei aber, dass die letztere Variante nur als Notlösung eingesetzt wird, da die problematische Codestelle z.B. in einem verschlüsselten Script liegt und daher nicht selbst angepasst werden kann. Wir empfehlen grundsätzlich möglichst ohne gepufferte Records auszukommen!

Rückgabewert:

Record

Beispiel VBScript:

```
...
If oRecordSet.MoveFirst Then
    Set oRecord = oRecordSet.CurrentRecord
    Do
        nUmsatz = nUmsatz + CDbl(oRecord.GetContentsByName("Turnover"))
    Loop Until Not oRecordSet.MoveNext
    Set oRecord = Nothing
End If
...
```

Beispiel C# Script:

```
RecordSet recordSet =
cRM.CurrentProject.ViewConfigs.ItemByName("Firmen").CreateRecordSet();
Record record;
var gesamtUmsatz = 0;
if (recordSet.MoveFirst() == true)
{
    record = recordSet.CurrentRecord;
    while (recordSet.MoveNext() == true)
    {
        gesamtUmsatz = gesamtUmsatz +
Convert.ToInt32(record.GetContentsValueByName("Turnover"));
    }
```

```
}
```

Siehe auch:

CurrentRecordBuffered

CurrentRecordBuffered

Beschreibung:

Liefert den aktuellen Datensatz als Objekt vom Typ Record zurück. Diese Methode funktioniert nur, wenn zuvor über eine der Move...-Methoden auf einen gültigen Datensatz gesprungen wurde. Wenn z.B. keine Datensätze im RecordSet enthalten sind, liefert diese Methode kein Record Objekt zurück.

Wichtig: Werden anschließend Move...-Befehle für den zugehörigen RecordSet aufgerufen, so aktualisiert sich das Record-Objekt nicht. Es behält die zum Zeitpunkt seiner Erzeugung aktuellen Feldinhalte. Um die neuen Inhalte zu bekommen, muss ein neues Objekt von RecordSet durch erneuten Aufruf von CurrentRecordBuffered erzeugt werden.

Diese Methode ist deutlich weniger performant als CurrentRecord und sollte nur verwendet werden, wenn unterschiedliche Record-Objekte als Variablen für unterschiedliche Datensätze benutzt werden müssen (vgl. Beispiel).

Rückgabewert:

Record

Beispiel VBScript:

```
...

Set oRecord1 = oRecordSet.CurrentRecordBuffered

oRecordSet.MoveNext

Set oRecord2 = oRecordSet.CurrentRecordBuffered

If CDbl(oRecord1.GetContentsByName("Turnover")) > _
               CDbl(oRecord2.GetContentsByName("Turnover")) Then

    ...

Set oRecord2 = Nothing

Set oRecord1 = Nothing

...
```

Beispiel C# Script:

```
RecordSet recordSet =
cRM.CurrentProject.ViewConfigs.ItemByName("Firmen").CreateRecordSet();

Record record1 = recordSet.CurrentRecordBuffered;

recordSet.MoveNext();

Record record2 = recordSet.CurrentRecordBuffered;
```

```
if (Convert.ToInt32(record1.GetContentsValueByName("Turnover")) > Con-
vert.ToInt32(record2.GetContentsValueByName("Turnover")))

{

    //...

}
```

Siehe auch:

CurrentRecord

CurrentRecordSynchronized

Beschreibung:

Liefert den aktuellen Datensatz als Objekt vom Typ Record zurück. Wenn z.B. keine Datensätze im RecordSet enthalten sind, liefert diese Methode kein Record Objekt zurück. Diese Methode ist aus Kompatibilitätsgründen enthalten, siehe die Methode CurrentRecord für den empfohlenen Weg.

Wichtig: Im Gegensatz zur Methode CurrentRecordBuffered aktualisiert sich ein so erzeugter Record immer automatisch, wenn anschließend Move...-Befehle für den zugehörigen RecordSet aufgerufen werden.

Da das komplette Puffern der Feldinhalte entfällt, ist diese Methode deutlich performanter als CurrentRecordBuffered und stellt den empfohlenen Weg dar, um auf Feldinhalte von Datensätzen zuzugreifen.

Mit CurrentRecordSynchronized erzeugte Record-Objekte können dadurch jedoch nicht als Variablen für unterschiedliche Datensätze benutzt werden (vgl. Beispiel zu CurrentRecordBuffered)! Ist dies erforderlich, so muss anstattdessen CurrentRecordBuffered benutzt werden.

Rückgabewert:

Record

Beispiel:

```
...

If oRecordSet.MoveFirst Then

    Set oRecord = oRecordSet.CurrentRecordSynchronized

    Do

        nUmsatz = nUmsatz + CDbl(oRecord.GetContentsByName("Turnover"))

    Loop Until Not oRecordSet.MoveNext

    Set oRecord = Nothing

End If

...
```

Beispiel C# Script:

```
RecordSet recordSet =
cRM.CurrentProject.ViewConfigs.ItemByName("Firmen").CreateRecordSet();

Record record;

var gesamtUmsatz = 0;

if (recordSet.MoveFirst() == true)

{

    record = recordSet.CurrentRecordSynchronized;

    while (recordSet.MoveNext() == true)

    {

        gesamtUmsatz = gesamtUmsatz +
Convert.ToInt32(record.GetContentsValueByName("Turnover"));

    }

}
```

Siehe auch:

CurrentRecordBuffered, CurrentRecord

DeleteAllRecords

Beschreibung:

Löscht alle Datensätze im Filter des aktuellen RecordSet-Objektes.

Rückgabewert:

Bool

Wert	Beschreibung
True	Alle Datensätze wurden gelöscht.
False	Befehl konnte nicht ausgeführt werden.

DialogSelectRecord

Beschreibung:

Ermöglicht die Auswahl eines Datensatzes basierend auf dem aktuellen Filter des RecordSet Objektes. Das Ergebnis ist ein Objekt vom Typ Record.

Wenn kein Datensatz ausgewählt wurde, wird NULL zurückgegeben. Das Objekt ist dann somit ungültig!

Hinweis: DialogSelectRecord speichert die Spalten/Layoutkonfiguration der Datensatzübersichtsliste ab (Voreinstellung: in der Registry). Der Name der Sektion der Konfiguration lautet "DialogSelect_"+<Ansichtenname>. Dieser kann jedoch individuell anderweitig vorgegeben wer-

den, indem man beim Funktionsaufruf einen eigenen Namen im Parameter für den Fenstertitel durch TAB (chr$(9)) getrennt mit übergibt.

Parameter:

Parametername	Typ	Beschreibung
WindowTitle	String	Fenstertitel des erzeugten Dialoges. Der Titel kann auch leer übergeben werden, dann wird eine Grundeinstellung genommen.
AllowUserDefined-SortOrder	Bool	True: Eine selbst zu definierende Sortierung wird verwendet. False: Die in der Ansicht definierte Sortierung wird verwendet.

Typ:

Record

DialogSelectRecordMultiple

Beschreibung:

Ermöglicht die Auswahl mehrerer Datensätze basierend auf dem aktuellen Filter des RecordSet Objektes. Das Ergebnis ist ein neues unabhängiges Objekt vom Typ RecordSet, das die gefiltertern Datensätze enthält.

Wenn kein Datensatz ausgewählt wurde, wird NULL zurückgegeben. Das Objekt ist dann somit ungültig!

Wichtig: Hierfür wird ein Feld vom internen Feldtyp 'Datensatz-ID' benötigt, welches mit entsprechenden global eindeutigen IDs gefüllt ist. Falls es sich nicht um ein Feld vom externen SQL Feldtyp 'uniqueidentifier' sondern 'nvarchar' handelt, muss dieses mindestens 32 Zeichen lang sein und die GUID in der cRM-internen Darstellung enthalten, z.B. "ce05a0f11dbf464f8a7f14666ddd0f3a".

Hinweis: DialogSelectRecordMultiple speichert die Spalten/Layoutkonfiguration der Datensatzübersichtsliste ab (Voreinstellung: in der Registry). Der Name der Sektion der Konfiguration lautet "DialogSelect_"+<Ansichtenname>. Dieser kann jedoch individuell anderweitig vorgegeben werden, indem man beim Funktionsaufruf einen eigenen Namen im Parameter für den Fenstertitel durch TAB (chr$(9)) getrennt mit übergibt.

Parameter:

Parametername	Typ	Beschreibung
WindowTitle	String	Fenstertitel des erzeugten Dialoges. Der Titel kann auch leer übergeben werden, dann wird eine Grundeinstellung genommen.
AllowUserDefined-SortOrder	Bool	True: Eine selbst zu definierende Sortierung wird verwendet. False: Die in der Ansicht definierte Sortierung wird verwendet.

Typ:

RecordSet

ExecuteInstantReportByName

Beschreibung:

Führt einen abgespeicherten Sofortbericht anhand seines Namens aus.

Hinweis: Es wird nur in der aktuell aktiven, d.h. sichtbaren, Ansicht nach dem Sofortbericht gesucht, d.h. gibt man einen Namen eines Sofortberichts an, der in einer anderen Ansicht definiert ist, schlägt die Methode fehl.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Name des Sofortberichts. Groß- / Kleinschreibung wird beachtet!

Typ:

Bool

Export

Beschreibung:

Startet den Export-Assistenten und führt das übergebene Benutzerformat aus. Es werden alle Einstellungen dieses Formates verwendet.

Parameter:

Parametername	Typ	Beschreibung
Format	String	Name des für den aktuellen Benutzer angelegten Benutzerformates. Hierbei handelt es sich um den Namen des Benutzerformates wie es im Assistenten gespeichert wurde, nicht um den Dateinamen der Vorlagendatei. Beachten Sie auch die Groß-/Kleinschreibung.

Rückgabewert:

Bool

FindRecordByEmail

Beschreibung:

Sucht alle Datensätze, bei denen in einem eMail-Feld eine bestimmte eMail-Adresse vorhanden ist. Diese unscharfe Suche findet auch eMail-Adressen derselben Domain falls kein direkter Treffer vorhanden ist.

Hinweis: Falls der COM-RecordSet auf einer sichtbaren Ansicht basiert, so wechselt die Ansicht bei mehreren Treffern automatisch in die Listenübersicht. Bei lediglich einem einzelnen Treffer erfolgt der Wechsel in die Eingabemasken-Ansicht.

Parameter:

Parametername	Typ	Beschreibung
sEmail	String	Die eMail-Adresse, nach der in allen eMail-Feldern gesucht werden soll.
bReduceEmail	Bool	Wenn True, dann wird die eMail-Adresse schrittweise in bis zu vier Durchläufen durch Auslassen von Top-Level-Domain, Domain, Lokalteil gesucht, falls es nicht bereits Treffer gab. So können auch eMails unterschiedlicher Domains gefunden werden.
bUseCurrentQuery	Bool	Optional (Voreinstellung: False), gibt an, ob lediglich innerhalb des aktuellen Filters gesucht werden soll.

Rückgabewert:

Bool

Wert	Beschreibung
True	Es gab Treffer, für den RecordSet ist nun der entsprechende Filter aktiv.
False	Es gab keine Treffer.

FindRecordByPhoneNumber

Beschreibung:

Sucht alle Datensätze, bei denen in einem Telefon-Feld eine bestimmte Telefonnummer vorhanden ist, analog zur Rufnummernerkennung durch den phone manager.

Hinweis: Falls der COM-RecordSet auf einer sichtbaren Ansicht basiert, so wechselt die Ansicht bei mehreren Treffern automatisch in die Listenübersicht. Bei lediglich einem einzelnen Treffer erfolgt der Wechsel in die Eingabemasken-Ansicht.

Parameter:

Parametername	Typ	Beschreibung
sPhoneNo	String	Die Telefonnummer, nach der in allen Telefonfeldern gesucht werden soll. Die Telefonnummer darf telefonspez. Sonderzeichen wie zum Beispiel '/', '+', '-' enthalten.
bReducePhoneNo	Bool	Wenn True, dann wird die Telefonnummer schrittweise um bis zu 4 Stellen verkürzt, falls es nicht bereits Treffer gab. So können auch unterschiedliche Durchwahl-Nummern gefunden werden.
bUseCurrentQuery	Bool	Optional (Voreinstellung: False), gibt an, ob lediglich innerhalb des aktuellen Filters gesucht werden soll.

Rückgabewert:

Bool

Wert	Beschreibung
True	Es gab Treffer, für den RecordSet ist nun der entsprechende

	Filter aktiv.
False	Es gab keine Treffer.

Import

Beschreibung:

Startet den Import-Assistenten und führt das übergebene Benutzerformat aus. Es werden alle Einstellungen dieses Formates verwendet.

Wichtig: Wenn ein Format zum automatischen Ausführen übergeben wird, oder die Methoden für einen unsichtbaren RecordSet aufgerufen werden, dann ist ein Import von unbekannten Codefeldern - im Gegensatz zu interaktivem Importieren - nicht möglich, diese werden dann ignoriert.

Parameter:

Parametername	Typ	Beschreibung
Format	String	Name des für den aktuellen Benutzer angelegten Benutzerformates.

Rückgabewert:

Bool

MoveFirst

Beschreibung:

Bewegt den Datensatz-Zeiger auf den Anfang des RecordSet.

Rückgabewert:

Bool

MoveLast

Beschreibung:

Bewegt den Datensatz-Zeiger auf das Ende der Datenbank.

Rückgabewert:

Bool

MoveNext

Beschreibung:

Bewegt den Datensatz-Zeiger um einen Datensatz vorwärts.

Rückgabewert:

Bool

MovePrevious

Beschreibung:

Bewegt den Datensatz-Zeiger um einen Datensatz rückwärts.

Rückgabewert:

Bool

NewRecord

Beschreibung:

Erzeugt einen neuen Datensatz und liefert diesen als Objekt vom Typ Record zurück.

Wichtig: Hierbei ist zu beachten, dass über SetContents... mindestens ein Feldinhalt gesetzt werden muss. Der neu angelegte Record muss nach dem Speichern nicht mit Unlock freigegeben werden (da er ja eben erst erzeugt wurde).

Für ein mit dieser Methode erzeugtes Record Objekt dürfen lediglich die Methoden Lock, Get/SetContents... mit einem (einmaligen) abschließenden Save und einem etwaigen (einmaligen) Unlock verwendet werden. Um andere Methoden des Record Objektes verwenden zu können, muss das Record Objekt freigegeben, neu initialisiert und auf den soeben erzeugten Datensatz positioniert werden.

Rückgabewert:

Record

PrintCard

Beschreibung:

Druckt ein Karteikartenprojekt aus. Neben dem Druckziel muss u.a. der Name des gewünschten Druckprojektes angegeben werden.

Parameter:

Parametername	Typ	Beschreibung
Media	String	Ausgabe-Medium. Zur Verfügung stehen: PRN: Drucker PRV: Vorschau PDF: Adobe PDF Format XHTML: XHTML/CSS Format MHTML: Multi-Mime HTML Format XLS: Microsoft Excel Format DOCX: Microsoft Word Format RTF: Rich Text Format (RTF) XPS: Microsoft XPS Format (Erfordert ein auf dem Arbeitsplatz installiertes .NET Framework 3.5 oder neuer) PICTURE_MULTITIFF: Multi-TIFF-Grafik PICTURE_TIFF: TIFF-Grafik PICTURE_PNG: PNG-Grafik PICTURE_PNG_CROP: Automatisch um überflüssigen weißen

		Rahmen beschnittene PNG-Grafik PICTURE_JPEG: JPG-Grafik PICTURE_JPEG_CROP: Automatisch um überflüssigen weißen Rahmen beschnittene JPG-Grafik PICTURE_BMP: Bitmap-Grafik PICTURE_EMF: Metafile-Grafik (EMF) PICTURE_EMF_CROP: Automatisch um überflüssigen weißen Rahmen beschnittene EMF-Grafik FILE: Datei HTML: HTML-Format JQM: HTML jQuery Mobile Format (nur für Listen) TTY: Nadeldrucker (TTY) TXT: Text (CSV) Format TXT_LAYOUT: Text (Layout) Format XML: XML Format
FileName	String	Dateiname inkl. Pfad des Druckprojektes.
Silent	Bool	Legt fest, ob der Druck (nach Möglichkeit) ohne Benutzerinteraktion erfolgen soll.
Range	Bool	True: Es werden alle Datensätze im aktuellen Filter/Datenbank ausgegeben, entspricht dem Seriendruck. False: Es wird nur der aktuelle Datensatz ausgegeben, entspricht dem Einzeldruck.
OutputFileName	String	Evtl. Name und Pfad der zu erzeugenden Ausgabe-Datei bei Ausgabe-Medien wie z.B. "RTF" oder "HTML". In allen anderen Fällen kann ein leerer String übergeben werden.
UseModalPreviewWindow	Bool	Optional. Legt fest, ob der Druck auf Vorschau (PRV) in das normale Vorschaufenster gedruckt wird (True, Voreinstellung) oder in die Berichtsansicht (False).

Rückgabewert:

Bool

Hinweis: Druckt man über ein RecordSet-Objekt, welches aus einer ViewConfig erstellt wurde, stehen momentan die Variablen *cRM.Project.?* und *cRM.View.?* nicht zur Verfügung.

PrintLabel

Beschreibung:

Druckt ein Etikettenprojekt aus. Neben dem Druckziel muss u.a. der Name des gewünschten Druckprojektes angegeben werden.

Parameter:

Parametername	Typ	Beschreibung
Media	String	Ausgabe-Medium. Zur Verfügung stehen: PRN: Drucker PRV: Vorschau PDF: Adobe PDF Format XHTML: XHTML/CSS Format MHTML: Multi-Mime HTML Format XLS: Microsoft Excel Format

		DOCX: Microsoft Word Format RTF: Rich Text Format (RTF) XPS: Microsoft XPS Format (Erfordert ein auf dem Arbeitsplatz installiertes .NET Framework 3.5 oder neuer) PICTURE_MULTITIFF: Multi-TIFF-Grafik PICTURE_TIFF: TIFF-Grafik PICTURE_PNG: PNG-Grafik PICTURE_PNG_CROP: Automatisch um überflüssigen weißen Rahmen beschnittene PNG-Grafik PICTURE_JPEG: JPG-Grafik PICTURE_JPEG_CROP: Automatisch um überflüssigen weißen Rahmen beschnittene JPG-Grafik PICTURE_BMP: Bitmap-Grafik PICTURE_EMF: Metafile-Grafik (EMF) PICTURE_EMF_CROP: Automatisch um überflüssigen weißen Rahmen beschnittene EMF-Grafik FILE: Datei HTML: HTML-Format JQM: HTML jQuery Mobile Format (nur für Listen) TTY: Nadeldrucker (TTY) TXT: Text (CSV) Format TXT_LAYOUT: Text (Layout) Format XML: XML Format
FileName	String	Dateiname inkl. Pfad des Druckprojektes.
Silent	Bool	Legt fest, ob der Druck (nach Möglichkeit) ohne Benutzerinteraktion erfolgen soll.
Range	Bool	True: Es werden alle Datensätze im aktuellen Filter/Datenbank ausgegeben, entspricht dem Seriendruck. False: Es wird nur der aktuelle Datensatz ausgegeben, entspricht dem Einzeldruck.
OutputFileName	String	Evtl. Name und Pfad der zu erzeugenden Ausgabe-Datei bei Ausgabe-Medien wie z.B. "RTF" oder "HTML". In allen anderen Fällen kann ein leerer String übergeben werden.
UseModalPreviewWindow	Bool	Optional. Legt fest, ob der Druck auf Vorschau (PRV) in das normale Vorschaufenster gedruckt wird (True, Voreinstellung) oder in die Berichtsansicht (False).

Rückgabewert:

Bool

Hinweis: Druckt man über ein RecordSet Objekt, welches aus einer ViewConfig erstellt wurde, stehen momentan die Variablen *cRM.Project.?* und *cRM.View.?* nicht zur Verfügung.

PrintReport

Beschreibung:

Druckt ein Listenprojekt aus. Neben dem Druckziel muss u.a. der Name des gewünschten Druckprojektes angegeben werden.

Parameter:

Parametername	Typ	Beschreibung

Media	String	Ausgabe-Medium. Zur Verfügung stehen: PRN: Drucker PRV: Vorschau PDF: Adobe PDF Format XHTML: XHTML/CSS Format MHTML: Multi-Mime HTML Format XLS: Microsoft Excel Format DOCX: Microsoft Word Format RTF: Rich Text Format (RTF) XPS: Microsoft XPS Format (Erfordert ein auf dem Arbeitsplatz installiertes .NET Framework 3.5 oder neuer) PICTURE_MULTITIFF: Multi-TIFF-Grafik PICTURE_TIFF: TIFF-Grafik PICTURE_PNG: PNG-Grafik PICTURE_PNG_CROP: Automatisch um überflüssigen weißen Rahmen beschnittene PNG-Grafik PICTURE_JPEG: JPG-Grafik PICTURE_JPEG_CROP: Automatisch um überflüssigen weißen Rahmen beschnittene JPG-Grafik PICTURE_BMP: Bitmap-Grafik PICTURE_EMF: Metafile-Grafik (EMF) PICTURE_EMF_CROP: Automatisch um überflüssigen weißen Rahmen beschnittene EMF-Grafik FILE: Datei HTML: HTML-Format JQM: HTML jQuery Mobile Format (nur für Listen) TTY: Nadeldrucker (TTY) TXT: Text (CSV) Format TXT_LAYOUT: Text (Layout) Format XML: XML Format
FileName	String	Dateiname inkl. Pfad des Druckprojektes
Silent	Bool	Legt fest, ob der Druck (nach Möglichkeit) ohne Benutzerinteraktion erfolgen soll.
Range	Bool	True: Es werden alle Datensätze in der aktuellen Ansicht ausgegeben, entspricht dem Seriendruck. False: Es wird nur der aktuelle Datensatz ausgegeben, entspricht dem Einzeldruck.
OutputFileName	String	Evtl. Name und Pfad der zu erzeugenden Ausgabe-Datei bei Ausgabe-Medien wie z.B. "RTF" oder "HTML". In allen anderen Fällen kann ein leerer String übergeben werden.
UseModalPreviewWindow	Bool	Optional. Legt fest, ob der Druck auf Vorschau (PRV) in das normale Vorschaufenster gedruckt wird (True, Voreinstellung) oder in die Berichtsansicht (False).

Rückgabewert:

Bool

Hinweis: Druckt man über ein RecordSet Objekt, welches aus einer ViewConfig erstellt wurde, stehen momentan die Variablen *cRM.Project.?* und *cRM.View.?* nicht zur Verfügung.

RelationalAppend

Beschreibung:

Führt eine abgespeicherte Vorlage für relationales Ergänzen über den Namen aus. Diese kann über den Menüpunkt **Organisieren** > **Relational Ergänzen** > **Allgemein** erzeugt werden.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Name der abgespeicherten Vorlage. Hinweis: Bitte beachten Sie, dass nicht die Bezeichnung des Filterausdrucks verwendet wird, sondern der 'Name für Scripte'.

Rückgabewert:

Bool

RelationalAppendDirect

Beschreibung:

Direktes relationales Ergänzen in einer Relationsansicht. Die Übergabe der Werte erfolgt in folgendem Format:

```
<FeldName1>=<Inhalt/Formel1>\t<FeldName2>=<Inhalt/Formel2>\t...
```

Parameter:

Parametername	Typ	Beschreibung
Relation	Objekt	Relationsobjekt
Parameter	String	Wertepaar

Rückgabewert:

Bool

Beispiel VBScript:

```
Set oRelations = oViewConfig.Relations

Set oRelation = oRelations.ItemByName("ID.Aktivitäten.ContactID")

oRecordSet.RelationalAppendDirect oRelation, " Direction =2" & vbTab & " ActivityType
=1" & vbTab & " Comment=«AskString$(""Kontaktbetreff?"", .F., "")»"
```

Beispiel C# Script:

```
ViewConfig viewConfig = cRM.CurrentProject.ViewConfigs.ItemByName("Kontakte");

ListRelations relations = viewConfig.Relations;

Relation relation = relations.ItemByName("ID.Aktivitäten.ContactID");

RecordSet recordSet = viewConfig.CreateRecordSet();

recordSet.RelationalAppendDirect(relation, "Direction=2" + "\t" + "ActivityType=1" +
"\t" + "Comment=«AskString$(\"Kontaktbetreff?\", .F., \"\")»");
```

SendBulkMail

Beschreibung:

Sendet eine Serien-eMail. Der Mailversand über diese Methode verhält sich analog zum Serien-Mailversand in der Anwendung.

Wichtig: Diese Methode steht nur in der cRM Enterprise Edition zur Verfügung.

Parameter:

Parametername	Typ	Beschreibung
TemplatePath	String	Pfad zur Mailvorlage
ErrorTagFilePath	String	Pfad zu .tag-Datei für manuelles Filtern. Diese wird erstellt wenn es im RecordSet Datensätze mit leeren eMail-Feldern gibt und enthält alle diese Datensätze. Falls ein leerer String übergeben wird, wird eine Datei im temp-Verzeichnis mit Standardnamen angelegt.

Rückgabewert:

Bool

Wert	Beschreibung
True	Der Versand war erfolgreich **und** es waren keine Datensätze mit leeren eMail-Feldern enthalten.
False	Der Versand ist fehlgeschlagen **oder** es war mindestens ein Datensatz mit leerem eMail-Feld enthalten.

SendMailDialog

Beschreibung:

Sendet eine eMail. Der Mailversand über diese Methode verhält sich analog zum Kontextmenü.

Parameter:

Parametername	Typ	Beschreibung
EmailAddress	String	Empfänger eMail-Adresse.
Subject	String	Betreff der eMail.
Contents	String	Text der eMail.
Files	String	Kann eine Liste von eMail-Anhängen beinhalten. Diese müssen dann per Semikolon separiert übergeben werden. z.B. C:\MyFiles\Report.pdf;C:\Info.doc

Rückgabewert:

Bool

SendMAPIMail

Beschreibung:

Sendet eine eMail über die MAPI-Schnittstelle.

Hinweis: Die eMail wird nicht automatisch abgelegt.

Parameter:

Parametername	Typ	Beschreibung
EMailAddress	String	Empfänger eMail-Adresse. SendMAPIMail unterstützt auch die Auflistung mehrerer Empfänger per Semikolon getrennt. Die Empfänger können dabei einen Prefix ("TO:", "CC:", "BCC:") haben, der bestimmt, um welche Art von Empfänger es sich handelt (Voreinstellung: "TO:", wenn kein Präfix angeben ist).
Subject	String	Betreff der eMail.
Contents	String	Text der eMail.
Files	String	Kann eine Liste von eMail-Anhängen beinhalten. Diese müssen dann per Semikolon separiert übergeben werden. z.B. C:\MyFiles\Report.pdf;C:\Info.doc
ShowDialog	Long	Anzeige eines Bestätigungsdialoges. 0 = Dialog wird nicht angezeigt. 1 = Dialog wird angezeigt.

Rückgabewert:

Bool

Beispiel VBScript:

```
oCurrentRecordSet.SendMAPIMail(CStr("info@relationship-
manager.net;CC:support@combit.net;BCC:webmaster@combit.net;TO: info@combit.net"),
CStr("Test"), CStr("Mailtext."), CStr(""), 1)
```

Beispiel C# Script:

```
RecordSet currentRecordSet =
cRM.CurrentProject.ActiveViews.ActiveView.CurrentRecordSet;

currentRecordSet.SendMAPIMail("info@relationship-
manager.net;CC:support@combit.net;BCC:webmaster@combit.net;TO:info@combit.net",
"Test", "Mailtext.", "", 1);
```

SetFilter

Beschreibung:

Erstellt einen Filter auf Basis des übergebenen Filter-Ausdruckes. Die Methode entspricht technisch **Filtern > Allgemein**. Übergeben Sie einen leeren String ("""), so wird ein bestehender Filter aufgehoben.

Der Filter kann immer nur für die jeweilige Ansicht/RecordSet ausgeführt werden. Für einen relationalen Filterausdruck können Sie entweder einen abgespeicherten Filter (**SetFilterByName**) oder **SetFilterDirectSQL** verwenden.

Wichtig: Es sollten zuvor alle etwaig geholten Record-Objekte auf Nothing gesetzt werden, keinesfalls dürfen sie anschließend noch verwendet werden.

Hinweis: Aus Performance-Gründen (mind. bei PostgreSQL) empfehlen wir Ihnen für eine Überprüfung des Filters *RecCount* zu vermeiden. Einen konkreten Lösungsansatz finden Sie im Scriptbeispiel "Filter-Ergebnis überprüfen".

Parameter:

Parametername	Typ	Beschreibung
FilterContents	String	Filterausdruck, entspricht den Formeln unter Filtern > Allgemein. Weitere Informationen finden Sie dort.
bUseCurrentQuery	Variant	True: Der Filter basiert auf dem aktuell bestehenden Filter. False (Voreinstellung): Es wird ein neuer Filter erstellt.

Rückgabewert:

Bool

Wert	Beschreibung
True	Der Filterausdruck wurde korrekt ausgeführt. Alle gefilterten Datensätze sind ab diesem Zeitpunkt im verwendeten **RecordSet**-Objekt enthalten.
False	Filterausdruck konnte nicht ausgeführt werden.

C# Script:

```
// Aktives Projekt

Project currentProject = cRM.CurrentProject;

combit.cRM.COM.View activeView = currentProject.ActiveViews.ActiveView;



InputForm currentInputForm = activeView.CurrentInputForm(0);



// Feld "BonitätOK" in der Ansicht "Firmen" - Registerkarte "Vertrieb" akti-
vieren:

currentInputForm.SetContentsByName("CreditOK", "T");

MessageBox.Show("Die Checkbox wurde aktiviert: " +
currentInputForm.GetContentsByName("CreditOK"), cRM.AppTitle,
MessageBoxButtons.OK);



// Feld "BonitätOK" in der Ansicht "Firmen" - Registerkarte "Vertrieb" wieder
deaktivieren:

currentInputForm.SetContentsByName("CreditOK", "F");
```

SetFilterByCurrentSortOrder

Beschreibung:

Erzeugt einen Filter basierend auf den Feldern der aktuellen Sortierung.

Wichtig: Das **RecordSet**-Objekt muss auf einem **View** basieren! Es sollten zuvor alle etwaig geholten **Record-** Objekte auf Nothing gesetzt werden, keinesfalls dürfen sie anschließend noch verwendet werden.

Parameter:

Parametername	Typ	Beschreibung
FilterContents	String	Das Suchkriterium.
SearchType	Long	0 Normal 1 Exakt 2 Wildcard 3 Phonetisch
CaseSensitive	Long	Gross-/Kleinschreibung beachten: 0 nicht CaseSensitive 1 CaseSensitive
bUseCurrentQuery	Variant	True: Der Filter basiert auf dem aktuell bestehenden Filter. False (Voreinstellung): Es wird ein neuer Filter erstellt.

Rückgabewert:

Bool

Wert	Beschreibung
True	Der Filterausdruck wurde korrekt ausgeführt. Alle gefilterten Datensätze sind ab diesem Zeitpunkt im verwendeten **RecordSet**-Objekt enthalten.
False	Filter konnte nicht ausgeführt werden.

SetFilterByFieldName

Beschreibung:

Setzt einen Filter anhand eines Feldinhalts.

Wichtig: Es sollten zuvor alle etwaig geholten **Record**-Objekte auf Nothing gesetzt werden, keinesfalls dürfen sie anschließend noch verwendet werden.

Parameter:

Parametername	Typ	Beschreibung
FieldName	String	Name des Feldes.
Contents	String	Feldinhalt
SearchType	Long	0 Normal 1 Exakt

		2 Wildcard 3 Phonetisch
CaseSensitive	Long	Gross-/Kleinschreibung beachten: 0 nicht CaseSensitive 1 CaseSensitive
bUseCurrentQuery	Variant	True: Der Filter basiert auf dem aktuell bestehenden Filter. False (Voreinstellung): Es wird ein neuer Filter erstellt.

Rückgabewert:

Bool

Wert	Beschreibung
True	Filter konnte erstellt werden. Alle gefilterten Datensätze sind ab diesem Zeitpunkt im verwendeten RecordSet-Objekt enthalten.
False	Filter konnte nicht ausgeführt werden.

SetFilterByName

Beschreibung:

Führt einen abgespeicherten Filterausdruck über den Namen aus. Dieser kann über den Menüpunkt Filtern > Filterausdrücke verwalten definiert werden.

Wichtig: Es sollten zuvor alle etwaig geholten Record-Objekte auf Nothing gesetzt werden, keinesfalls dürfen sie anschließend noch verwendet werden.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Name des abgespeicherten Filters. Hinweis: Bitte beachten Sie, dass nicht die Bezeichnung des Filterausdrucks verwendet wird, sondern der "Name für Scripte".
bUseCurrentQuery	Variant	True: Der Filter basiert auf dem aktuell bestehenden Filter. False (Voreinstellung): Es wird ein neuer Filter erstellt.

Rückgabewert:

Bool

Wert	Beschreibung
True	Filter konnte erstellt werden. Alle gefilterten Datensätze sind ab diesem Zeitpunkt im verwendeten RecordSet-Objekt enthalten.
False	Filter konnte nicht ausgeführt werden.

SetFilterByPrimaryKey

Beschreibung:

Setzt einen Filter basierend auf dem Primärschlüssel des RecordSets.

Diese Methode funktioniert nur, wenn ein Primärschlüssel vorhanden ist, der genau aus einem Feld besteht.

Wichtig: Es sollten zuvor alle etwaig geholten Record-Objekte auf Nothing gesetzt werden, keinesfalls dürfen sie anschließend noch verwendet werden.

Parameter:

Parametername	Typ	Beschreibung
PrimaryKeyContent	String	Inhalt des Primärschlüsselfeldes.

Rückgabewert:

Bool

SetFilterDirectSQL

Beschreibung:

Erstellt einen Filter auf Basis des übergebenen freien Filterausdruckes.

Hinweis: In der Abfrage muss generell im SELECT-Teil die Primärschlüsselspalte der zum RecordSet gehörenden Tabelle selektiert werden, der restliche Aufbau der Abfrage ist beliebig.

Wichtig: Es sollten zuvor alle etwaig geholten Record-Objekte auf Nothing gesetzt werden, keinesfalls dürfen sie anschließend noch verwendet werden.

Parameter:

Parametername	Typ	Beschreibung
sSQLQuery	String	Freier SQL Filterausdruck.
bUseCurrentQuery	Variant	True: Der Filter basiert auf dem aktuell bestehenden Filter. False (Voreinstellung): Es wird ein neuer Filter erstellt.

Rückgabewert:

Bool

Wert	Beschreibung
True	Filter konnte erstellt werden. Alle gefilterten Datensätze sind ab diesem Zeitpunkt im verwendeten RecordSet-Objekt enthalten.
False	Filter konnte nicht ausgeführt werden.

Beispiel VBScript:

```
oRecordSet.SetFilterDirectSQL "SELECT Contacts.ID FROM Contacts WHERE Salutation =
'Herrn', True"
```

Beispiel C# Script:

```
RecordSet recordSet = cRM.CurrentProject.ActiveViews.ActiveView.CurrentRecordSet;

recordSet.SetFilterDirectSQL("SELECT Contacts.ID FROM Contacts WHERE Salutation =
\'Herrn\'", true);
```

Synchronize

Beschreibung:

Startet den Abgleich-Assistenten und führt das übergebene Benutzerformat aus. Es werden alle Einstellungen dieses Formates verwendet.

Wichtig: Bitte beachten Sie, dass nach dem Aufruf dieser Methode View.Update aufgerufen werden muss.

Es wird dabei das benutzerspezifische Format vor dem globalen Format verwendet. Ist dies nicht gewünscht, kann der Dateiname inkl. Pfad des gewünschten Formats angegeben werden.

Wenn ein Format zum automatischen Ausführen übergeben wird, oder die Methoden für einen unsichtbaren RecordSet aufgerufen werden, dann ist ein Abgleich von unbekannten Codefeldern - im Gegensatz zum interaktiven Abgleich - nicht möglich, diese werden dann ignoriert.

Parameter:

Parametername	Typ	Beschreibung
Format	String	Name des für den aktuellen Benutzer angelegten Benutzerformates.

Rückgabewert:

Bool

3.29 RecurrencePattern Objekt

Dieses Objekt ähnelt dem von Outlook verwendeten Objekt in seiner Schnittstelle. Es dient zur Speicherung der Eigenschaften bzw. Einstellungen von Serienterminen.

3.29.1 Eigenschaften

DayOfMonth

Beschreibung:

Setzt den Tag des Monats (1-31) an dem ein Termin auftritt. Wird bei olRecursMonthly(2) und olRecursYearly(5) verwendet.

Parameter:

Long

Rückgabewert:

Long

DayOfWeekMask

Beschreibung:

Gibt die Wochentage für wiederkehrende Termine an. Wird bei olRecursDaily(0), olRecursMonthNth(3), olRecursWeekly(1) und olRecursYearNth(6) verwendet.

Parameter:

Long

Folgende Werte können miteinander ver'oder't werden:

Konstante	Wert	Beschreibung
OlSunday	1	Sonntag
OlMonday	2	Montag
OlTuesday	4	Dienstag
OlWednesday	8	Mittwoch
OlThursday	16	Donnerstag
OlFriday	32	Freitag
OlSaturday	64	Samstag

Rückgabewert:

Long

Duration

Beschreibung:

Dauer eines Termins in Minuten.

Parameter:

Long

Rückgabewert:

Long

EndTime

Beschreibung:

Liefert/setzt die Endzeit für ein Serienmuster.

Parameter:

Date

Rückgabewert:

Date

Instance

Beschreibung:

Dieser Wert ist nur für olRecursMonthNth(3) und olRecursYearNth(6) gültig und ermöglicht die Definition eines Serienmusters, das nur für das n-te Vorkommen gilt, wie z. B. ein Muster der Form "der zweite Sonntag im März".

Die Zahl wird numerisch festgelegt: 1 für das erste, 2 für das zweite und so weiter bis 5 für das letzte.

Parameter:

Long

Rückgabewert:

Long

Interval

Beschreibung:

Gibt die Anzahl von Einheiten zwischen dem Auftreten zweier Serientermine an, also z.B. Interval = 2 bei Terminen alle zwei Wochen.

Wird bei olRecursDaily(0), olRecursWeekly(1), olRecursMonthly(2), olRecursMonthNth(3), verwendet.

Parameter:

Long

Rückgabewert:

Long

MonthOfYear

Beschreibung:

Gibt den Monat im Jahr (1-12) für einen Serientermin an.

Wenn diese Eigenschaft z.B. auf 5 gesetzt wird und RecurrenceType olRecursYearly (5) ist, so wird dieser Termin jedes Jahr im Mai auftreten.

Parameter:

Long

Rückgabewert:

Long

NoEndDate

Beschreibung:

Ist True, wenn ein Serientermin kein Enddatum besitzt.

Parameter:

Bool

Rückgabewert:

Bool

PatternEndDate

Beschreibung:

Enddatum eines Serientermins. Die NoEndDate-Eigenschaft wird dabei zurückgesetzt.

Parameter:

Date

Rückgabewert:

Date

PatternStartDate

Beschreibung:

Anfangsdatum eines Serientermins.

Parameter:

Date

Rückgabewert:

Date

RecurrenceType

Beschreibung:

Gibt die Art des Serientermins an, die Konstanten sind analog zur Outlook-Schnittstelle.

Konstante	Wert	Beschreibung
olRecursDaily	0	Tägliche Wiederholung
olRecursWeekly	1	Wöchentliche Wiederholung
olRecursMonthly	2	Monatliche Wiederholung

olRecursMonthNth	3	Monatliche Wiederholung spezieller Art
olRecursYearly	5	Jährliche Wiederholung
olRecursYearNth	6	Jährliche Wiederholung spezieller Art

Parameter:

Long

Rückgabewert:

Long

Hinweis: Diese Eigenschaft muss als erste Eigenschaft des RecurrencePattern-Objektes gesetzt werden.

StartTime

Beschreibung:

Startzeit für ein Serienmuster.

Parameter:

Date

Rückgabewert:

Date

3.30 Relation Objekt

3.30.1 Eigenschaften

Alias

Beschreibung:

Liefert den Relationsalias zurück.

Rückgabewert:

String

FieldName

Beschreibung:

Liefert den Namen des Feldes, von dem die Relation weg führt.

Rückgabewert:

String

ForeignViewFieldName

Beschreibung:

Liefert den Namen des Feldes, auf das die Relation zeigt (Fremdschlüssel).

Rückgabewert:

String

ForeignViewName

Beschreibung:

Liefert den Namen der Ansicht, zu der die Relation hin führt.

Rückgabewert:

String

RelType

Beschreibung:

Liefert den Typ der Relation zurück.

Rückgabewert:

Long

Wert	Beschreibung
1	1:1 Relation
2	1:N Relation
3	N:M Relation

3.31 SQLShell Objekt

Führt eine SQL-Anweisung direkt aus und liefert die Anzahl Datensätze zurück, die davon betroffen sind.

Das Objekt kann nicht von außen instanziert werden und steht daher in dieser Form nur in Scripten zur Verfügung, die innerhalb des Programms aufgerufen werden.

3.31.1 Eigenschaften

LastError, read-only

Beschreibung:

Liefert ein OLEError-Objekt zurück.

Typ:

OLEError

Wert	Beschreibung
0	Kein Fehler aufgetreten.

1	Rückgabe der Fehlermeldung des Servers.

3.31.2 Methoden

ExecuteCommandRaw

Beschreibung:

Führt eine SQL-Anweisung direkt aus. Es wird dabei die aktuelle Verbindung zur Datenbank der Solution verwendet.

Parameter:

Parametername	Typ	Beschreibung
SQLCommand	String	SQL-Anweisung

Hinweis: Es macht keinen Sinn, hierüber eine SELECT Anweisung abzusetzen, da man hierüber das Ergebnis nicht abfragen/durchlaufen kann. Einsatzzweck sind UPDATE, INSERT, DELETE Anweisungen oder ggf. das Aufrufen von Stored Procedures.

Rückgabewert:

Long

Wert	Beschreibung
>= 0	Anzahl der betroffenen Datensätze bei UPDATE oder DELETE Anweisungen.
0	Rückgabewert bei INSERT Anweisungen oder dem Aufruf von Stored Procedures.
-1	Fehler

Beispiel VBScript:

```
nResult = SQLShell.ExecuteCommandRaw("UPDATE ""FIRMEN"" SET ""ABC"" = 'A'")

If nResult = -1 Then

    MsgBox SQLShell.LastError.ErrorCode & ": " & SQLShell.LastError.ErrorText

Else

    MsgBox nResult & " Datensätze betroffen."

End If
```

Beispiel C# Script:

```
long result = SQLShell.ExecuteCommandRaw("UPDATE \"Companies\" SET \"ABC\" =
'A'");

if (result == -1)

{

     MessageBox.Show(SQLShell.LastError.ErrorCode + ": " +
SQLShell.LastError.ErrorText, cRM.AppTitle, MessageBoxButtons.OK);

}

else
```

```
{
    MessageBox.Show(result.ToString() + " Datensätze betroffen", cRM.AppTitle,
MessageBoxButtons.OK);
}
```

3.32 timemanager Objekt

3.32.1 Eigenschaften

ActiveView

Beschreibung:

Legt die aktuelle Ansicht fest oder gibt diese zurück. Es muss daher eine Konstante aus dem Wertebereich der TMViewConstants übergeben werden.

Konstanten der TMViewConstants:

Konstante	Wert	Beschreibung
TM_VIEW_DAY	0	Tagesansicht
TM_VIEW_WEEK	1	Wochenansicht
TM_VIEW_MONTH	2	Monatsansicht
TM_VIEW_APPLIST	3	Terminliste
TM_VIEW_TODO	4	Aufgaben

Rückgabewert:

TMViewConstants

Appointments, read-only

Beschreibung:

Gibt in einem Objekt vom Typ Appointments (s.u.) eine Sammlung aller Termine zurück.

Rückgabewert:

Appointments

CurrentUser

Beschreibung:

Name des aktuellen Benutzers.

Rückgabewert:

String

HostID

Beschreibung:

Eindeutige ID der aktuellen Host Applikation. Dieser Wert korrespondiert zu der HostID-Eigenschaft der ToDo- und Appointment-Objekte, s.u.

Rückgabewert:

String

ShowReminder

Beschreibung:

Diese Eigenschaft ermöglicht es (bei False) das Erscheinen von Termin-Erinnerungen zu unterdrücken.

Rückgabewert:

Bool

ToDos

Beschreibung:

Gibt in einem Objekt vom Typ **ToDos** (s.u.) eine Sammlung aller Aufgaben zurück.

Rückgabewert:

ToDos

ViewDate

Beschreibung:

Gibt das aktuelle Datum der Ansichten zurück oder legt dieses fest.

Parameter:

Parametername	Typ	Beschreibung
Date	Date	Datum der Ansicht.

Rückgabewert:

Date

Visible

Beschreibung:

Setzt den Anzeigezustand der Ansicht *Termine & Aufgaben* oder gibt diesen zurück.

Parameter:

Typ	Beschreibung
Bool	Setzt den Anzeigezustand.

Rückgabewert:

Bool

3.32.2 Methoden

UpdateViews

Beschreibung:

Aktualisiert alle Ansichten.

Wichtig: Muss nach der Neuanlage eines Termins/einer Aufgabe für die Aktualisierung der Anzeige des neu angelegten Termins/Aufgabe aufgerufen werden!

3.33 ToDo Objekt

Hinweis: Wenn man die Datumswerte auf einen ungültigen Wert setzt (hier 0), so wird automatisch 31.12.1900 0 Uhr eingetragen.

Wenn man die Werte NICHT setzt, so wird das heutige Datum genommen.

3.33.1 Eigenschaften

ActionID

Beschreibung:

Eindeutige ID der zugeordneten Aktion.

Konstante	Beschreibung
TMMAIL	Senden einer eMail-Benachrichtigung.
TMSHEX	Datei/Dokument ausführen bzw. öffnen.
DIAL_CRM	Telefonanruf ausführen.

Parameter:

String

Rückgabewert:

String

Attendees, read-only

Beschreibung:

Liefert alle Teilnehmer der Aufgabe als Objekt vom Typ **Attendees** zurück.

Rückgabewert:

Attendees

Body

Beschreibung:

Inhalt/Text der Aufgabe (mit RTF-Formatierung).

Parameter:

String

Rückgabewert:

String

BodyPlain

Beschreibung:

Inhalt/Text der Aufgabe (ohne RTF-Formatierung).

Parameter:

String

Rückgabewert:

String

Categories, read-only

Beschreibung:

Liefert alle Kategorien der Aufgabe als Objekt vom Typ **Categories** zurück.

Rückgabewert:

Categories

ChangeDate

Beschreibung:

Letztes Änderungsdatum der Aufgabe.

Parameter:

Date

Rückgabewert:

Date

ChangeUser

Beschreibung:

Letzter Änderungsbenutzer der Aufgabe.

Parameter:

String

Rückgabewert:

String

Contact

Beschreibung:

Kontakt, mit dem die Aufgabe stattfindet ("Mit").

Rückgabewert:

String

CreationDate

Beschreibung:

Erfassungsdatum der Aufgabe.

Parameter:

Date

Rückgabewert:

Date

CreationUser

Beschreibung:

Erfassungsbenutzer der Aufgabe.

Parameter:

String

Rückgabewert:

String

End

Beschreibung:

Enddatum und -zeit der Aufgabe/Fälligkeitsdatum.

Parameter:

Date

Rückgabewert:

Date

ExtUserData1

Beschreibung:

Ermöglicht das Hinterlegen von Zusatzinformationen in Form von Zeichenketten (Strings).

Hinweis: Die Zeichenketten können maximal jeweils 250 Zeichen lang werden.

Rückgabewert:

String

ExtUserData2

Beschreibung:

Ermöglicht das Hinterlegen von Zusatzinformationen in Form von Zeichenketten (Strings).

Hinweis: Die Zeichenketten können maximal jeweils 250 Zeichen lang werden.

Rückgabewert:

String

HostDatabase

Beschreibung:

Eindeutige Projekt-ID. Der Aufbau ist folgender:

<ProjektID>|<Ansichtenname>|<Ansichtenfamilienname(optional)>|<Primärschlüsselfeldname>|<Primärschlüsselwert>

Rückgabewert:

String

Beispiel VBScript:

```
Dim sRecordRef

sRecordRef = CStr(cRM.CurrentProject.ID) & "|" & _
```

```
CStr(cRM.CurrentProject.ActiveViews.ActiveView.Name) & "|" & _

CStr(oViewConfig.FamilyName) & "|" & _

CStr(oViewConfig.PrimaryKeyFldName) & "|" & _

CStr(oRecord.GetContentsByName(Ctr(sPKFieldName)))

ToDo.HostDatabase = sRecordRef
```

Beispiel C# Script:

```
ViewConfig viewConfig =
cRM.CurrentProject.ViewConfigs.ItemByName("<Ansichtsname>");

Record record = viewConfig.CreateRecordSet().CurrentRecordSynchronized;

ToDo todo = cRM.CurrentProject.TimeManager.ToDos.Item(0);

string recordRef = cRM.CurrentProject.ID + "|" +
cRM.CurrentProject.ActiveViews.ActiveView.Name + "|" + viewConfig.FamilyName + "|"
+ viewConfig.PrimaryKeyFldName + "|" +

record.GetContentsByName(viewConfig.PrimaryKeyFldName);

todos.HostDatabase = recordRef;
```

HostID

Beschreibung:

Eindeutige ID des (optional) zugeordneten Mutterprogrammes (HOST) der Aufgabe, wie z.B. der Adressverwaltung.

Parameter:

String

Rückgabewert:

String

HostRecordID

Beschreibung:

Korrespondierend zu **HostDatabase** die eindeutige Datensatznummerierung des (optional) zugeordneten Datensatzes.

Parameter:

String

Rückgabewert:

String

Importance

Beschreibung:

Priorität der Aufgabe (1 bis 5).

Parameter:

Long

Rückgabewert:

Long

Links, read-only

Beschreibung:

Liefert alle Verknüpfungen der Aufgabe als Objekt vom Typ Links zurück.

Rückgabewert:

Links

PercentComplete

Beschreibung:

Prozentzahl, zu der die Aufgabe als "erledigt" gilt.

Parameter:

Long

Rückgabewert:

Long

Private

Beschreibung:

Legt fest, ob die Aufgabe als "Privat" markiert werden soll.

Parameter:

Bool

Rückgabewert:

Bool

Reminder

Beschreibung:

Legt fest, ob an diese Aufgabe erinnert werden soll.

Parameter:

Bool

Rückgabewert:

Bool

ReminderMinutesBeforeStart

Beschreibung:

Gibt die Anzahl der Minuten zurück, die vor der Fälligkeit der Aufgabe an diese erinnert werden soll.

Parameter:

Long

Rückgabewert:

Long

Start

Beschreibung:

Startdatum und -zeit der Aufgabe.

Parameter:

Date

Rückgabewert:

Date

Subject

Beschreibung:

Beschreibung der Aufgabe.

Parameter:

String

Rückgabewert:

String

TimeStamp

Beschreibung:

Zeichenkette, die aus dem Änderungsdatum und einem fortlaufenden Zahlenwert (als String) besteht. Der Zahlenwert wird bei jeder Datensatzänderung verändert. Durch diesen Wert kann man auch feststellen, wenn ein Datensatz sich an einem Tag mehrfach geändert hat.

Format:YYYYMMDD:HHMMSS
z.B. 20070919:100120

Parameter:

String

Rückgabewert:

String

UniqueID, read-only

Beschreibung:

Eindeutige RecordID der Aufgabe.

Parameter:

String

User

Beschreibung:

Eigentümer der Aufgabe.

Parameter:

String

Rückgabewert:

String

3.33.2 Methoden

Display

Beschreibung:

Zeigt einen Dialog zur Aufgabe an.

Parameter:

Parametername	Typ	Beschreibung
bModal	Bool	Display-Methode kann sich über einen Parameter gesteuert wahlweise modal

		verhalten. Dies geht jedoch momentan nur bei Neuanlage. z.B. oTask.Display(True).

Rückgabewert:

Bool

ExportAsICal

Beschreibung:

Exportiert die Aufgabe in das iCal-Format.

Parameter:

Parametername	Typ	Beschreibung
sFileName	String	Pfad und Dateiname der zu exportierenden Datei.

Rückgabewert:

Bool

Wert	Beschreibung
True	Datei wurde exportiert.
False	Beim Exportieren der Datei ist ein Fehler aufgetreten.

Remove

Beschreibung:

Löscht die Aufgabe.

Rückgabewert:

Bool

Save

Beschreibung:

Speichert die Einstellungen der Aufgabe.

Rückgabewert:

Bool

Wichtig: Um die neu angelegte Aufgabe in der Ansicht gleich angezeigt zu bekommen, muss UpdateViews nach der Speicherung aufgerufen werden.

3.34 ToDos Objekt

3.34.1 Eigenschaften

Count, read-only

Beschreibung:

Liefert die Anzahl der Einträge in der Sammlung zurück.

Rückgabewert:

Long

3.34.2 Methoden

Add

Beschreibung:

Legt eine neue Aufgabe an und liefert diese als Objekt vom Typ ToDo zurück.

Rückgabewert:

ToDo

Item

Beschreibung:

Gibt eine Aufgabe zurück. Es muss die Index-Nummer der Aufgabe übergeben werden. Der Index geht von 1 bis Count.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Index-Nummer.

Rückgabewert:

ToDo

ItemByUniqueID

Beschreibung:

Gibt eine Aufgabe anhand der eindeutigen RecordID zurück.

Parameter:

Parametername	Typ	Beschreibung
UniqueID	String	Eindeutige RecordID der Aufgabe.

Rückgabewert:

ToDo

Remove

Beschreibung:

Löscht eine Aufgabe. Es muss die Index-Nummer der Aufgabe übergeben werden. Der Index geht von 1 bis Count.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Index-Nummer.

Rückgabewert:

Bool

RemoveAll

Beschreibung:

Löscht alle Aufgaben.

Rückgabewert:

Bool

SetFilter

Beschreibung:

Filtert die Anzeige auf Aufgaben, die bestimmte Eigenschaften erfüllen. Die Filterung kann ausgehend von Datenbank oder Datensatz der Host-Applikation erfolgen oder auf Basis des Benutzers.

Parameter:

Parametername	Typ	Beschreibung
FilterType	TMListFilter Constants	Art der Filterung.
HostDatabase	String	Datenbank (Pfad+Name).
HostRecordID	String	Eindeutige Datensatznummer des zugeordneten Datensatzes.
UserName	String	Name des Benutzers.

Konstanten der TMListFilterConstants:

Konstante	Wert	Beschreibung
TM_FILTER_HOSTDB	1	Filterung aller Aufgaben zu einer bestimmten Datenbank, korrespondierend zu der Eigenschaft HostDatabase des Appointment-

		Objektes.
TM_FILTER_HOSTRECID	2	Filterung aller Aufgaben zu einem bestimmten Datensatz, korrespondierend zu der Eigenschaft HostRecordID des Appointment-Objektes. Impliziert die Verwendung eines Filters auf Datenbank-Ebene, vgl. TM_FILTER_HOSTDB.
TM_FILTER_USER	4	Filterung aller Aufgaben zu einem bestimmten Benutzer.

Rückgabewert:

Bool

3.35 User Objekt

Zugriff auf die Stammdaten eines Benutzers der combit Anwendung. Das User-Objekt erhält man über CurrentUser oder ItemByName der Users-Collection, z.B.

Beispiel VBScript:

```
MsgBox cRM.CurrentProject.Users.CurrentUser.FirstName

MsgBox cRM.CurrentProject.Users.ItemByName("Administrator").LastName
```

Beispiel C# Script:

```
MessageBox.Show(cRM.CurrentProject.Users.CurrentUser.FirstName, cRM.AppTitle, Mes-
sageBoxButtons.OK);

MessageBox.Show(cRM.CurrentProject.Users.ItemByName("Administrator").LastName,
cRM.AppTitle, MessageBoxButtons.OK);
```

3.35.1 Eigenschaften

Department, read-only

Beschreibung:

Liefert die Abteilung zurück.

Rückgabewert:

String

Email, read-only

Beschreibung:

Liefert eMail zurück.

Rückgabewert:

String

Email2, read-only

Beschreibung:

Liefert eMail2 zurück.

Rückgabewert:

String

Fax, read-only

Beschreibung:

Liefert Fax zurück.

Rückgabewert:

String

FirstName, read-only

Beschreibung:

Liefert den Vornamen zurück.

Rückgabewert:

String

GroupNames, read-only

Beschreibung:

Liefert in einem String kommasepariert alle Gruppen (jeweils in einfache Anführungszeichen eingeschlossen), in denen der Benutzer Mitglied ist, z.B. 'Vertrieb', 'Administratoren'.

Typ:

String

IsLoggedIn, read-only

Beschreibung:

Liefert True zurück, wenn der Benutzer gerade eingeloggt ist.

Rückgabewert:

Bool

LastName, read-only

Beschreibung:

Liefert den Nachnamen zurück.

Rückgabewert:

String

LoginName, read-only

Beschreibung:

Liefert den Loginnamen zurück.

Rückgabewert:

String

Mobile, read-only

Beschreibung:

Liefert Mobiltelefon zurück.

Rückgabewert:

String

Phone, read-only

Beschreibung:

Liefert Telefon zurück.

Rückgabewert:

String

Phone2, read-only

Beschreibung:

Liefert Telefon2 zurück.

Rückgabewert:

String

ShortName, read-only

Beschreibung:

Liefert den Kurznamen zurück.

Rückgabewert:

String

Signature, read-only

Beschreibung:

Liefert die Unterschrift (Dateipfad) zurück.

Rückgabewert:

String

Title, read-only

Beschreibung:

Liefert die Anrede zurück.

Rückgabewert:

String

UserDefined1, read-only

Beschreibung:

Liefert Zusatz1 zurück.

Rückgabewert:

String

UserDefined2, read-only

Beschreibung:

Liefert Zusatz2 zurück.

Rückgabewert:

String

3.36 Users Objekt

Bietet den Zugriff auf die konfigurierten Benutzer der combit Anwendung.

> Hinweis: Für den Zugriff auf andere Benutzer als den eigenen, wird das Recht 'Andere Benutzer sehen' benötigt. Ist dieses nicht gewährt, so liefert Count() immer 1 und der Zugriff auf Item(1) liefert immer den eigenen Benutzer. ItemByName funktioniert in diesem Falle auch nur, wenn der eigene Benutzername übergeben wird.

3.36.1 Eigenschaften

Count, read-only

Beschreibung:

Liefert die Anzahl der konfigurierten Benutzer der combit Anwendung zurück.

Typ:

Long

3.36.2 Methoden

CurrentUser

Beschreibung:

Liefert angemeldeten Benutzer der combit Anwendung als Objekt vom Typ User zurück.

Rückgabewert:

User

Item

Beschreibung:

Gibt einen Benutzer zurück. Es muss die Index-Nummer des Eintrages übergeben werden. Der Index geht von 1 bis Count.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Index-Nummer.

Rückgabewert:

User

ItemByName

Beschreibung:

Liefert einen Benutzer mit dem übergebenen Namen als Objekt zurück.

Parameter:

Parametername	Typ	Beschreibung
ViewName	String	Name des Benutzers

Rückgabewert:

User

SyncWithActiveDirectory

Beschreibung:

Synchronisiert die Benutzerverwaltung der combit Anwendung mit dem konfigurierten Active Directory.

Wichtig: Diese Methode steht erst ab der cRM Professional Edition zur Verfügung.

Rückgabewert:

Long

Wert	Beschreibung
0	Synchronisation erfolgreich.
1	Keine Konfiguration gefunden.
2	Fehler während der Synchronisation.
3	Mehrere Benutzer mit gleichem Login Namen oder Windows Login Namen sind nicht erlaubt.
4	Benutzer ist weder Administrator noch hat er administrative Rechte.

3.37 View Objekt

Repräsentiert eine aktive (geöffnete) Ansicht in der combit Anwendung.

Hinweis: Unter C# kann es bei der Initalisierung des Objektes zu einem Konflikt mit einem weiteren View-Objekt des Namespaces System.Windows.Forms kommen. Wir empfehlen daher den kompletten Klassennamen zu verwenden:

Beispiel:
combit.cRM.COM.View activeView = cRM.CurrentProject.ActiveViews.ActiveView;

3.37.1 Eigenschaften

Config, read-only

Beschreibung:

Erzeugt ein Objekt vom Typ ViewConfig.

Typ:

ViewConfig

Editmode

Beschreibung:

Liefert oder verändert den Status des Bearbeitungsmodus in der aktuellen Ansicht.

Typ:

Bool

FilterActive, read-only

Beschreibung:

Liefert True zurück, wenn in der Ansicht ein vom Benutzer angelegter Filter aktiv ist.

Typ:

Bool

FilterRecCount, read-only

Beschreibung:

Gibt die Anzahl der Datensätze im aktuellen Filter zurück. Ist kein Filter aktiv, so wird RecCount zurückgeliefert.

Typ:

Long

Name, read-only

Beschreibung:

Liefert den Namen der Ansicht zurück.

Typ:

String

RecCount, read-only

Beschreibung:

Gibt die Anzahl der Datensätze zurück. Ein etwaiger Filter wird dabei ignoriert. Soll dieser berücksichtigt werden, verwenden Sie stattdessen FilterRecCount.

Typ:

Long

Wert	Beschreibung
Long	Anzahl der Datensätze
- 1	Fehler

TagID

Beschreibung:

Mit Hilfe dieser Eigenschaft können Sie eine eigene TagID (Identifiername) für eine aktive Ansicht übergeben bzw. auslesen. Ein Objekt vom Typ View (aktive Ansicht) kann anschließend über ItemByTagID erzeugt werden.

Parameter:

Parametername	Typ	Beschreibung
TagID	String	Identifiername der geöffneten Ansicht

Typ:

String

ViewMode

Beschreibung:

Gibt den Modus der betreffenden Ansicht zurück oder setzt diesen.

Typ:

Integer

Wert	Beschreibung
1	Karteikartenansicht
2	Listenansicht
3	Webansicht

3.37.2 Methoden

Activate

Beschreibung:

Aktiviert die Ansicht und bringt sie in den Vordergrund.

Rückgabewert:

Bool

Close

Beschreibung:

Close versucht die Ansicht zu schließen.

Wichtig: Ruft man Close in der aktuellen Ansicht auf (in der das Script läuft), kann dies zu Problemen führen. Dies sollte daher vermeiden werden.

Rückgabewert:

-

CurrentInputForm

Beschreibung:

Diese Methode wechselt in der aktuellen Ansicht in den Bearbeitungsmodus und liefert das InputForm Objekt der aktuellen Eingabemaske zurück.

Parameter:

Parametername	Typ	Beschreibung
nMode	Long	0 = Wechsel in Bearbeiten-Modus (kompatibel zu bisher) 1 = Neuanlage eines Datensatzes und Wechsel in Bearbeiten-Modus (kompatibel zu bisher) 2 = Beibehalten des aktuellen Modus

Typ:

InputForm

CurrentRecordSet

Beschreibung:

Liefert ein Objekt vom Typ RecordSet zurück.

Rückgabewert:

RecordSet

EditMailTemplate

Beschreibung:

Diese Methode öffnet eine Mailvorlage im Mail-Editor.

Parameter:

Parametername	Typ	Beschreibung
TemplateFileName	String	Pfad der zu öffnenden Mailvorlage. Wenn kein Pfad übergeben wird, dann erfolgt die Dateiauswahl interaktiv.

EditReportTemplate

Beschreibung:

Diese Methode öffnet eine Druckvorlage im Druckvorlagen-Designer.

Parameter:

Parametername	Typ	Beschreibung

TemplateFileName	String	Pfad der zu öffnenden Druckvorlage. Wenn kein Pfad übergeben wird, dann erfolgt die Dateiauswahl interaktiv.

InvokeMenu

Beschreibung:

Ruft einen Menüeintrag der combit Anwendung auf. Neben der ID des Menüeintrages wird angegeben, ob das Script solange warten soll, bis der Befehl abgearbeitet wurde (und evtl. Dialoge geschlossen wurden) oder ob das Script direkt weiterlaufen soll. Die Menü-IDs der combit Anwendung finden Sie im Kapitel "Menü-IDs".

Hinweis: Es werden nur Menü-IDs von direkt sichtbaren Menüs unterstützt, d.h. Kontextmenüs können nicht verwendet werden.

Parameter:

Parametername	Typ	Beschreibung
MenuID	Long	Die ID des Menüeintrages.
Synchron	Bool	True: synchrone Ausführung False: asynchrone Ausführung

Rückgabewert:

Bool

Wert	Beschreibung
True	Befehl wurde ausgeführt.
False	Befehl konnte nicht ausgeführt werden, dies kann z.B. der Fall sein, wenn der Befehl derzeit nicht zur Verfügung steht.

StartEditNew

Beschreibung:

Legt in der aktuellen Ansicht einen neuen Datensatz an und wechselt in den Änderungsmodus (analog zum Menüpunkt 'Neuer Datensatz anlegen').

Rückgabewert:

Bool

Wert	Beschreibung
True	Der Datensatz konnte angelegt werden.
False	Das Ausführen der Methode ist fehlgeschlagen.

Update

Beschreibung:

Aktualisiert die aktuelle Ansicht. Dabei wird der aktuelle Datensatz neu dargestellt, und die Statusleiste wird ebenfalls aktualisiert. Es sollten zuvor sowohl alle etwaigen von der View geholten RecordSet-Objekte als auch alle etwaig geholten Record-Objekte auf Nothing gesetzt werden, keinesfalls dürfen sie anschließend noch benutzt werden.

Rückgabewert:

Bool

3.38 ViewConfig Objekt

3.38.1 Eigenschaften

DBTableName, read-only

Beschreibung:

Liefert den Datenbank-Tabellennamen der Ansicht zurück.

Typ:

String

FamilyName, read-only

Beschreibung:

Liefert den definierten Familiennamen der Ansicht zurück.

Typ:

String

FldCount, read-only

Beschreibung:

Liefert die Anzahl der Felder zurück.

Typ:

Long

Name, read-only

Beschreibung:

Liefert den Namen der definierten Ansicht zurück.

Typ:

String

PrimaryKeyFldName, read-only

Beschreibung:

Liefert den Namen des Primärschlüsselfeldes, sofern der Primärschlüssel aus genau einem Feld besteht.

Typ:

String

Relations

Beschreibung:

Liefert ein Objekt vom Typ ListRelations zurück.

Typ:

ListRelations

SortOrderCount, read-only

Beschreibung:

Liefert die Anzahl der definierten Sortierungen zurück.

Typ:

Long

3.38.2 Methoden

CodeDefinitions

Beschreibung:

Liefert für das übergebene Feld eine Collection ListCodeDefinitions der dafür hinterlegten Codedefinitionen.

Parameter:

Parametername	Typ	Beschreibung
FieldName	String	Physikalischer Feldname des gewünschten Feldes.

Rückgabewert:

ListCodeDefinitions

CreateRecordSet

Beschreibung:

Erzeugt ein Objekt vom Typ RecordSet. Es wird daher keine Ansicht visuell geöffnet.

Rückgabewert:

RecordSet

FldAlias

Beschreibung:

Liefert den Aliasnamen des Feldes zurück, das als fortlaufende Nummer übergeben wurde. Ist der Aliasname leer, wird der entsprechende physikalische Feldname zurückgegeben.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Die fortlaufende Nummer des gewünschten Feldes in der Datenbankstruktur von 1 bis FldCount.

Rückgabewert:

String

FldAliasByName

Beschreibung:

Liefert den Aliasnamen des Feldes zurück, dessen physikalischer Feldname übergeben wurde. Ist der Aliasname leer, wird der entsprechende physikalische Feldname zurückgegeben.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Physikalischer Feldname des gewünschten Feldes.

Rückgabewert:

String

FldDec

Beschreibung:

Liefert die Anzahl der Dezimalstellen des Feldes zurück, das als fortlaufende Nummer übergeben wurde.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Die fortlaufende Nummer des gewünschten Feldes in der Datenbankstruktur von 1 bis FldCount.

Rückgabewert:

Long

FldExists

Beschreibung:

Überprüft, ob ein Feld in der Ansicht existiert.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Übergabe des zu überprüfenden Feldnamens.

Rückgabewert:

Bool

FldName

Beschreibung:

Liefert den Namen des Feldes zurück, das als fortlaufende Nummer übergeben wurde.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Die fortlaufende Nummer des gewünschten Feldes in der Datenbankstruktur von 1 bis FldCount.

Rückgabewert:

String

FldLen

Beschreibung:

Liefert die Zeichenlänge des Feldes zurück, das als fortlaufende Nummer übergeben wurde.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Die fortlaufende Nummer des gewünschten Feldes in der Datenbankstruktur von 1 bis FldCount.

Rückgabewert:

Long

FldReadOnly

Beschreibung:

Überprüft, ob ein Feld gesetzt (verändert) werden darf.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Die fortlaufende Nummer des gewünschten Feldes in der Datenbankstruktur von 1 bis FldCount.

Rückgabewert:

Bool

Wert	Beschreibung
True	Das Feld ist schreibgeschützt und kann nicht überschrieben werden.
False	Das Feld ist nicht schreibgeschützt und kann überschrieben werden.

FldType

Beschreibung:

Liefert den internen Typ des Feldes zurück, das als fortlaufende Nummer übergeben wurde.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Die fortlaufende Nummer des gewünschten Feldes in der Datenbankstruktur von 1 bis FldCount.

Rückgabewert:

Long

Interne Feldtypen der combit Anwendung:

Wert	Feldtyp intern
0	Unbekannt
1	Logisch
2	Zeichen
3	Notizen
4	Numerisch
5	Datum mit Zeit
6	Datum
7	Zeit
8	Numerisch binary
9	Telefon
10	Mobiltelefon
11	Internet
12	eMail
13	Code
14	Global eindeutige ID
15	Postleitzahl
16	Straße

17	Ort
18	Land
19	Bundesland
20	Bankleitzahl
21	Datensatz-ID
22	Grafik
23	Datei
24	Automatische Nr.
-	reserviert
27	Telefax
28	Postfach Postleitzahl
29	Postfach
30	Erfassungsbenutzer
31	Erfassungsdatum
32	Änderungsbenutzer
33	Änderungsdatum
34	Auto Sequenz
35	Symbol
-	Internal
37	Eingebettete Datei
38	Eingebettete Grafik
39	DMS Dokument
40	intern
41	Belegverweis (FM) – Format: fmw://<Mandantname>/<Belegtyp>/<BelegNummer>"
43	Notizen formatiert

FldTypePhys

Beschreibung:

Liefert den physikalischen Feldtyp zurück.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Die fortlaufende Nummer des gewünschten Feldes in der Datenbankstruktur von 1 bis FldCount.

Rückgabewert:

Long

Wert	Feldtyp
0	Unbekannt
1	Logisch
2	Ganzzahl kurz
3	Ganzzahl lang
4	Rationale Zahl
5	Numerisch
6	Datum mit Zeit

7	Zeichen
8	Bytes
9	Binär lang
10	Zeichen lang
11	BLOB
12	CLOB

RowIDFldName

Beschreibung:

Liefert die echte RowID-Spalte zurück.

Rückgabewert:

String

SortOrderName

Beschreibung:

Liefert anhand der übergebenen Sortierung den Namen der Sortierung zurück.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Übergabe der Sortierung 0 = <ohne Sortierung>

Rückgabewert:

String

3.39 WebElement Objekt

Bietet Zugriff auf ein Web-Element.

3.39.1 Eigenschaften

ID

Beschreibung:

Liefert die ID des Web-Elements zurück.

Rückgabewert:

String

3.39.2 Methoden

IE

Beschreibung:

Liefert das Interface des Internet Explorers als Objekt vom Typ InternetExplorer zurück.

Siehe http://msdn.microsoft.com/en-us/library/aa752084%28v=vs.85%29.aspx

Rückgabewert:

InternetExplorer

Beispiel VBScript:

```
cRM.CurrentProject.ActiveViews.ActiveView.CurrentInputForm(2).WebElements.ItemByNa
me("{64BBFFD7-EA32-4358-BBFC-744D6A94291D}").IE.Navigate "http://www.combit.net"
```

Beispiel C# Script:

```
cRM.CurrentProject.ActiveViews.ActiveView.CurrentInputForm(2).WebElements.ItemByNa
me("{64BBFFD7-EA32-4358-BBFC-
744D6A94291D}").IE().Navigate("http://www.combit.net");
```

Hinweis: combit kann nicht gewährleisten, dass alle Eigenschaften und Methoden des Internet Explorers im Kontext der Verwendung als Control innerhalb der Eingabemaske (hier als Web-Element) wirklich unterstützt werden. Es wird lediglich das Interface nach außen zur Verfügung gestellt.

3.40 WScript Objekt

Das WScript-Objekt beinhaltet Eigenschaften und Methoden der Scripting-Engine sowie von der Datenbank oder dem aktuellen Status unabhängige Funktionalität.

Das Objekt kann nicht von außen instanziert werden und steht daher in dieser Form nur in Scripten zur Verfügung, die innerhalb des Programms aufgerufen werden.

3.40.1 Eigenschaften

ClipboardText

Beschreibung:

Setzt oder liest den Textinhalt der Zwischenablage.

Typ:

String

Event

Beschreibung:

Bei durch Ereignisse ausgelösten Scripten steht ein zusätzliches Objekt zur Verfügung:

WScript.Event

Ausführliche Informationen dazu erhalten Sie im Kapitel "Event Objekt".

Typ:

Event

Priority

Beschreibung:

Legt die Priorität des Scripts fest oder gibt diese zurück. Je höher die Priorität, desto häufiger bekommt das Script Rechenzeit zugeteilt und umgekehrt. Die möglichen Werte liegen zwischen –2 und +2, wobei –2 die niedrigste und +2 die höchste Priorität darstellt.

Hinweis: Kann nur in asynchronen Scripts verwendet werden.

Typ:

Long

RightClicked, read-only

Beschreibung:

Gibt True zurück, wenn das Script in der Eingabemaske über einen Button ausgeführt wurde und dieser mit der rechten Maustaste geklickt wurde. Ermöglicht die Ausführung zweier verschiedener Funktionen über einen Button.

Typ:

Bool

Terminate, read-only

Beschreibung:

Gibt True zurück, wenn die combit Anwendung gerade beendet wird bzw. das aktive Script zum Beenden aufgefordert wird. Die Eigenschaft kann und sollte unbedingt in Verbindung mit asynchronen Scripts verwendet werden und ermöglicht ein sauberes Verlassen des Scripts.

Bitte beachten Sie dabei, dass die noch laufenden, asynchronen Scripte insgesamt 4 Sekunden Zeit haben, sich zu beenden, bevor die Beendigung erzwungen wird. Sleeps und andere Aktionen im Script sollten also keinesfalls länger als 3 Sekunden dauern, bevor wieder die .Terminate Eigenschaft geprüft und das Script dann ggf. sich so schnell wie möglich kontrolliert beenden muss.

Typ:

Bool

3.40.2 Methoden

ActivateTitle

Beschreibung:

Aktiviert ein Windows-Fenster über dessen Titel.

Parameter:

Parametername	Typ	Beschreibung
WindowTitle	String	Der Titel (Name) des zu aktivierenden Fensters. Wenn Sie als erstes Zeichen ein "$" übergeben, muss der übergebene Titel lediglich enthalten sein, ansonsten genau übereinstimmen.

ActivateTitleByClassname

Beschreibung:

Aktiviert ein Windows-Fenster über dessen Klassennamen.

Parameter:

Parametername	Typ	Beschreibung
Classname	String	Der Klassenname des zu aktivierenden Fensters.

Beep

Beschreibung:

Gibt den Standard Windows Warnton aus.

CheckAbortedWaitDlg

Beschreibung:

Die Methode muss zwischen StartWaitDlg und EndWaitDlg aufgerufen werden und gibt zurück, ob der Benutzer die Schaltfläche "Abbrechen" des Wartedialoges betätigt hat, sofern StartWaitDlg mit Abbruch-Button angezeigt wurde. Die Methode wird vorzugsweise in Ausführungsschleifen verwendet.

Rückgabewert:

Bool

EndWaitDlg

Beschreibung:

Blendet einen zuvor mit StartWaitDlg angezeigten Wartedialog aus.

GetGlobalProperty

Beschreibung:

Liest eine globale Script-Einstellung aus, die zuvor mit SetGlobalProperty gesetzt wurde oder gibt den Standardwert zurück.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Der Name der Einstellung.
Default	String	Der Standardwert der Einstellung, der zurückgegeben wird, wenn keine Einstellung mit dem angegebenen Namen existiert.

Rückgabewert:

String

GetUserProperty

Beschreibung:

Liest eine benutzerspezifische Script-Einstellung aus, die zuvor mit SetUserProperty gesetzt wurde oder gibt den Standardwert zurück.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Der Name der Einstellung.
Default	String	Der Standardwert der Einstellung, der zurückgegeben wird, wenn keine Einstellung mit dem angegebenen Namen existiert.

Rückgabewert:

String

SetGlobalProperty

Beschreibung:

Setzt eine globale Script-Einstellung. Die Speicherung erfolgt in der Datei "gloabl.ini" in der "cmbt_Files" Tabelle.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Der Name der Einstellung.
String	String	Der (neue) Wert der Einstellung.

Rückgabewert:

Bool

SetUserProperty

Beschreibung:

Setzt eine benutzerspezifische Script-Einstellung. Die Speicherung erfolgt in der Datei "user.ini" in der "cmbt_Files" Tabelle.

Parameter:

Parametername	Typ	Beschreibung
Name	String	Der Name der Einstellung.
String	String	Der (neue) Wert der Einstellung.

Rückgabewert:

Bool

SetWaitDlgText

Beschreibung:

Hiermit kann der Text in einem per StartWaitDlg gestarteten Wartedialog nachträglich geändert werden.

Parameter:

Parametername	Typ	Beschreibung
Text	String	Der anzuzeigende Informationstext.

Sleep

Beschreibung:

Hält die Ausführung des Scripts für eine anzugebende Zeit an.

Parameter:

Parametername	Typ	Beschreibung
Time	Long	Anzahl Millisekunden, die das Script angehalten werden soll. `Sleep 1000` hält die Ausführung bspw. für 1 Sekunde an.

StartWaitDlg

Beschreibung:

Zeigt einen Wartedialog mit einer Fortschrittsanimation sowie einem zu übergebenden Informationstext an. Die Ausführung des Scripts wird während der Anzeige fortgeführt. Die Ausblendung erfolgt mit EndWaitDlg. Über den zweiten Parameter kann bestimmt werden, ob der Dialog eine "Abbrechen"-Schaltfläche enthalten soll. Wenn ja, dann kann mit CheckAbortedWaitDlg geprüft werden, ob die Schaltfläche betätigt wurde.

Parameter:

Parametername	Typ	Beschreibung
Text	String	Der anzuzeigende Informationstext.
Cancel	Bool	Schaltfläche "Abbrechen" anzeigen?

WaitTitle

Beschreibung:

Methode wartet so lange, bis das Fenster mit dem übergebenen Titel gefunden wurde. Anschließend wird das gefundene Fenster aktiviert.

Parameter:

Parametername	Typ	Beschreibung
Titel	String	Der Titel (Name) des zu aktivierenden Fensters. Wenn Sie als erstes Zeichen ein "$" übergeben, muss der übergebene Titel lediglich enthalten sein, ansonsten genau übereinstimmen.
Time	Long	Max. Wartezeit in Sekunden. -1 bedeutet unbegrenzte Wartezeit.

Rückgabewert:

Bool

Wert	Beschreibung
True	Fenster wurde gefunden.
False	Kein Fenster gefunden.

4 Menü-IDs

4.1 Projekt-Menü

Befehl in Symbolleiste	Befehl in Menüband	ID
Datei > Neu	Datei > Neu	57600
Datei > Öffnen > Datei	Datei > Öffnen > Öffnen	57601
Datei > Öffnen > Solution-Center	Datei > Öffnen > Solution Center	33246
Datei > Speichern	Datei > Speichern	57603
Datei > Speichern unter	Datei > Speichern unter	57604
Datei > Reorganisieren	Datei > Information > Reorganisieren	33201
Datei > Für WebAccess bereitstellen	Datei > Information > Für WebAccess bereitstellen	33206
Datei > Beenden	Datei > Beenden	57665
Ansichten > Neu	Kontextmenü auf Ansicht im Projekt-Fenster > Neu	32800
Ansichten > Öffnen	Start > Öffnen > Ansicht bzw. Fenster > Wechseln zu > Ansicht bzw. Kontextmenü auf Ansicht im Projekt-Fenster > Öffnen	33186
Ansichten > Eigenschaften	Konfigurieren > Ansicht > Eigenschaften bzw. Kontext-menü auf Ansicht im Projekt-Fenster > Eigenschaften	32801
Ansichten > Löschen	Kontextmenü auf Ansicht im Projekt-Fenster > Löschen	32802
Ansichten > Info-Zentrale	Fenster > Wechseln zu > Info-Zentrale	33131
Ansichten > Termine & Aufgaben	Fenster > Wechseln zu > Termine und Aufgaben	33091
Ansichten > Anrufe	Fenster > Wechseln zu > Anrufe	33127
Fenster > Symbolleisten > Symbolleiste Projekt	*keine Entsprechung vorhanden*	59392
Fenster > Symbolleisten > Statusleiste	Fenster > Anzeigen > Statusleiste	59393
Fenster > Symbolleisten > Anpassen	*keine Entsprechung vorhanden*	33151
Fenster > Navigation	Fenster > Anzeigen > Navigationsbereich	59512
Fenster > Dubletten	Fenster > Anzeigen > Dubletten	59513
Fenster > Suchergebnis Globale Suche	Fenster > Anzeigen > Suchergebnisse	59525
Fenster > Tabs	*keine Entsprechung vorhanden*	33081
Fenster > Überlappend	*keine Entsprechung vorhanden*	57650
Fenster > Übereinander	*keine Entsprechung vorhanden*	57651
Fenster > Symbole anordnen	*keine Entsprechung vorhanden*	57649
Fenster > Alle schließen	Fenster > Fenster > Alle schließen	33148
Fenster > Fenster	Fenster > Fenster > Fenster wechseln	32933
Konfigurieren > Allgemein	Datei > Optionen > Allgemein	32915
Konfigurieren > Projekt > Eigenschaften	Datei > Information > Eigenschaften	32905
Konfigurieren > Projekt > Ereignisse	Datei > Information > Ereignisse	32987
Konfigurieren > Projekt > Navigationsstruktur	Datei > Information > Navigationsstruktur	33100
Konfigurieren > Projekt > Firmenstammdaten	Datei > Information > Firmenstammdaten	33129
Konfigurieren > Projekt > Importieren	Datei > Information > Projekt-Import	33243
Konfigurieren > Akustische Signale	Datei > Optionen > Akustische Signale	32914
Konfigurieren > Datenbank > Datenbankverbindung	Datei > Optionen > Datenbank Anmeldung	32899
Konfigurieren > Datenbank > Datenbankstruktur	Datei > Information > Datenbankstruktur	33016
Konfigurieren > Systemverwaltung > Benutzer	Datei > Optionen > Benutzerverwaltung	32805
Konfigurieren > Systemverwaltung > Automatische Zäh-ler	Datei > Information > Automatische Zähler	32897
Konfigurieren > Benutzerkennwort ändern	Datei > Optionen > Kennwort ändern	32937
Konfigurieren > DMS Anmeldeinformationen	Datei > Optionen > DMS Anmeldung	33042
? > Hilfethemen	Datei > Hilfe	57666
? > Kontakt	Datei > Hilfe > Kontakt	32236
? > Online-Registrierung	Datei > Hilfe > Online-Registrierung	32938
? > Online-Support > Knowledgebase	Datei > Hilfe > Support > Knowledgebase	32942

? > Online-Support > Foren	Datei > Hilfe > Support > Foren	32941
? > Online-Support > Support-Bereich	Datei > Hilfe > Support > Support-Bereich	32940
? > Online-Support > Aktualitätsprüfung im Internet	Datei > Hilfe > Aktualitätsprüfung im Internet	32939
? > Online-Support > An Online-Sitzung teilnehmen	Datei > Hilfe > Support > An Online-Sitzung teilnehmen	32229
? > Tools > address pick-up	Datei > Hilfe > Tools > address pick-up	33358
? > Tools > eMail Autopilot	Datei > Hilfe > Tools > eMail Autopilot	33359
? > Tools > LDAP Server	Datei > Hilfe > Tools > LDAP Server	33360
? > Tools > Workflow Server	Datei > Hilfe > Tools > Workflow Server	33361
? > Tools > Lizenz-Upgrade	Datei > Hilfe > Tools > Lizenz-Upgrade	33363
? > Tools > Solution-Distribution	Datei > Hilfe > Tools > Solution-Distribution	33364
? > Tools > Debug Tool	Datei > Hilfe > Tools > Debug Tool	33362
? > Über	Datei > Hilfe > Info über combit Relationship Manager	57664

4.2 Ansichten-Menü

Befehl in Symbolleiste	Befehl in Menüband	ID
Datensatz > Neu	Start > Datensatz > Neu	32781
Datensatz > Bearbeiten	Start > Datensatz > Bearbeiten	32807
Datensatz > In neuem Fenster bearbeiten	Start > Datensatz > Bearbeiten > In neuem Fenster bearbeiten	33202
Datensatz > Speichern	Bearbeiten Kontext > Speichern	32806
Datensatz > Speichern und Ansicht schließen	Bearbeiten Kontext > Speichern & Schließen	33163
Datensatz > Änderungen verwerfen	Bearbeiten Kontext > Abbrechen	33150
Datensatz > Löschen	Start > Datensatz > Löschen	32774
Datensatz > Alle Löschen	Start > Datensatz > Löschen > Alle löschen	32896
Datensatz > Duplizieren	Start > Datensatz > Neu > Duplizieren	32809
Datensatz > Zusammenführen	Start > Datensatz > Erweitert > Datensätze zusammenführen	33138
Datensatz > Gehe zu > Anfang	Gehe zu > Erster Datensatz bzw. Statusleiste	32787
Datensatz > Gehe zu > Vorwärts	Gehe zu > Nächster Datensatz bzw. Statusleiste	32789
Datensatz > Gehe zu > Rückwärts	Gehe zu > Vorheriger Datensatz bzw. Statusleiste	32788
Datensatz > Gehe zu > Ende	Gehe zu > Letzter Datensatz bzw. Statusleiste	32790
Datensatz > Verweis > Senden an	Start > Datensatz > Verweis > Senden an	32864
Datensatz > Verweis > Speichern als	Start > Datensatz > Verweis > Speichern als	32865
Datensatz > Verweis > In Zwischenablage	Start > Datensatz > Verweis > In Zwischenablage	32866
Datensatz > Verweis > Zu Favoriten hinzufügen	Start > Datensatz > Verweis > Zu Favoriten hinzufügen	33108
Datensatz > Überwachen	Start > Datensatz > Erweitert > Datensatz überwachen	33252
Datensatz > Anrufen	Start > Aktionen > Anrufen	33182
Datensatz > eMail senden	Start > Aktionen > eMail senden	33183
Bearbeiten > Rückgängig	Bearbeiten Kontext > Rückgängig	57643
Bearbeiten > Ausschneiden	Bearbeiten Kontext > Ausschneiden	57635
Bearbeiten > Kopieren	Bearbeiten Kontext > Kopieren	57634
Bearbeiten > Einfügen	Bearbeiten Kontext > Einfügen	57637
Suchen > Schnellsuche	Start > Suchen > Schnellsuche	32949
Suchen > Suchen	Start > Suchen > Suchen	32810
Suchen > Telefonnummer oder eMail	Start > Suchen > Spezialsuche	33177

Suchen > Aktuellen Datensatz in Online-Verzeichnissen	Start > Suchen > In Verzeichnissen > Datensatz in Online-Verzeichnissen und Social Media suchen	33254
Suchen > Telefonverzeichnis > Anhand aktuellem Datensatz	Start > Suchen > In Verzeichnissen > Datensatz in Telefonverzeichnis suchen	32816
Suchen > Telefonverzeichnis > Manuell übernehmen	Start > Suchen > In Verzeichnissen > Manuelle Übernahme aus einem Telefonverzeichnis	32817
Suchen > Routenplanung > Aktuellen Datensatz als Startpunkt setzen	Start > Aktionen > Routenplanung > Aktuellen Datensatz als Startpunkt setzen	33056
Suchen > Routenplanung > Aktuellen Datensatz als Zwischenstopp setzen	Start > Aktionen > Routenplanung > Aktuellen Datensatz als Zwischenstopp setzen	33057
Suchen > Routenplanung > Aktuellen Datensatz als Endpunkt setzen	Start > Aktionen > Routenplanung > Aktuellen Datensatz als Endpunkt setzen	33058
Suchen > Routenplanung > Alle Datensätze im Filter als Wegpunkte setzen	Start > Aktionen > Routenplanung > Alle Datensätze im Filter als Wegpunkte setzen	33069
Suchen > Routenplanung > Alle Wegpunkte löschen	Start > Aktionen > Routenplanung > Alle Wegpunkte löschen	33060
Suchen > Routenplanung > Routenplanung	Start > Aktionen > Routenplanung > Routenplanung	33059
Suchen > Routenplanung > Routenplanung zum aktuellen Datensatz	Start > Aktionen > Routenplanung > Routenplanung zum aktuellen Datensatz	33061
Suchen > Ersetzen	Start > Suchen > Ersetzen	32821
Filtern > Allgemein	Start > Filtern > Filter erstellen bzw. Filtern > Zusammenstellen > Filter Allgemein	32829
Filtern > Volltextrecherche	Filtern > Zusammenstellen > Volltextrecherche	32832
Filtern > Formularabfrage	Filtern > Zusammenstellen > Formularabfrage	32840
Filtern > Freie SQL-Abfrage	Filtern > Zusammenstellen > Freie SQL-Abfrage	32988
Filtern > Manuell > Zusammenstellen	Filtern > Manuell > Zusammenstellen	33089
Filtern > Manuell > Anwenden/Aufheben	Filtern > Manuell > Anwenden	33114
Filtern > Manuell > Datensatz einbeziehen	Filtern > Manuell > Datensatz einbeziehen	32892
Filtern > Manuell > Datensatz ausschließen	Filtern > Manuell > Datensatz ausschließen	32893
Filtern > Manuell > Alle Datensätze einbeziehen	Filtern > Manuell > Alle einbeziehen	32894
Filtern > Manuell > Alle Datensätze ausschließen	Filtern > Manuell > Alle ausschließen	32895
Filtern > Manuell > Datensatzauswahl laden	Filtern > Manuell > Datensatzauswahl > Manuellen Filter laden	33141
Filtern > Manuell > Datensatzauswahl speichern als	Filtern > Manuell > Datensatzauswahl > Manuellen Filter speichern	33143
Filtern > Manuell > Datensatzauswahl aufheben	Filtern > Manuell > Auswahl aufheben	33098
Filtern > Dubletten	Filtern > Zusammenstellen > Dublettenprüfung	32831
Filtern > Filterausdruck ausführen	Filtern > Zusammenstellen > Filterausdruck ausführen	32887
Filtern > Filterausdrücke verwalten	Konfigurieren > Verwalten > Filterausdrücke	32888
Filtern > Letzter Filter	Start > Filtern > Letzten Filter aktivieren bzw. Filtern > Zusammenstellen > Letzter Filter	32823
Filtern > Filter rückgängig	Filtern > Zusammenstellen > Filter rückgängig	32824
Filtern > Filter invertieren	Filtern > Zusammenstellen > Filter invertieren	33140
Filtern > Filter in phone manager übernehmen	Daten > Übernehmen > Filter in den phone manager übernehmen	32898
Filtern > Filter als manuellen Filter speichern	Filtern > Manuell > Datensatzauswahl > Aktuellen Filter als manuellen Filter speichern	33178
Organisieren > Sortieren	Konfigurieren > Verwalten > Sortierungen	32825
Organisieren > Importieren	Daten > Externe Daten > Importieren	32820
Organisieren > Exportieren	Daten > Externe Daten > Exportieren	32830
Hinweis: Bei der Verwendung des Ereignisses "Menübefehl wird ausgeführt" muss auch die ID 32930 berücksichtigt werden.		

Organisieren > Abgleichen	Daten > Externe Daten > Importieren mit Abgleich	32943
Organisieren > Anreichern aus Telefonverzeichnis	Daten > Bearbeiten > Datenanreicherung	32886
Organisieren > Wechseln	Daten > Bearbeiten > Wechseln	32814
Organisieren > Relational ergänzen > Allgemein	Daten > Bearbeiten > Relational ergänzen > Allgemein	32900
Organisieren > Relational ergänzen > Vorlage ausführen	Daten > Bearbeiten > Relational ergänzen > Vorlage ausführen	33134
Organisieren > Relational ergänzen > Vorlagen verwalten	Konfigurieren > Vorlagen > Relationales Ergänzen	33136
Ausgeben > Einzelübernahme	Ausgeben > Aktueller Datensatz > Übernahme in Textverarbeitung	32819
Hinweis: Bei der Verwendung des Ereignisses "Menübefehl wird ausgeführt" muss auch die ID 32945 berücksichtigt werden.		
Ausgeben > Einzel-eMail	Ausgeben > Aktueller Datensatz > eMail senden	32811
Ausgeben > Einzelbrief > Direkt	Ausgeben > Aktueller Datensatz > Brief drucken	32925
Ausgeben > Einzelbrief > Textverarbeitung	Ausgeben > Aktueller Datensatz > Übernahme in Textverarbeitung	32945
Hinweis: Bei der Verwendung des Ereignisses "Menübefehl wird ausgeführt" muss auch die ID 32819 berücksichtigt werden.		
Ausgeben > Einzeldruck > Allgemein	Ausgeben > Aktueller Datensatz > Drucken	32230
Ausgeben > Einzeldruck > Etikett	Ausgeben > Drucken > Aktueller Datensatz > Einzelnes Etikett drucken	32792
Ausgeben > Einzeldruck > Karteikarte	Ausgeben > Drucken > Aktueller Datensatz > Einzelne Karteikarte drucken	32793
Ausgeben > Einzeldruck > Liste	Ausgeben > Drucken > Aktueller Datensatz > Liste (Bericht) zum Datensatz drucken	32920
Ausgeben > Serien-eMail	Ausgeben > Alle Datensätze > Serien-eMail senden	32812
Ausgeben > Serienbrief > Direkt	Ausgeben > Alle Datensätze > Serienbriefe drucken	32923
Ausgeben > Serienbrief > Textverarbeitung	Ausgeben > Alle Datensätze > Serienbrief per Textverarbeitung	32930
Hinweis: Bei der Verwendung des Ereignisses "Menübefehl wird ausgeführt" muss auch die ID 32830 berücksichtigt werden.		
Ausgeben > Seriendruck > Allgemein	Ausgeben > Alle Datensätze > Seriendruck	32231
Ausgeben > Seriendruck > Etiketten	Ausgeben > Alle Datensätze > Seriendruck > Etiketten drucken	32794
Ausgeben > Seriendruck > Karteikarten	Ausgeben > Alle Datensätze > Seriendruck > Karteikarten drucken	32795
Ausgeben > Seriendruck > Liste	Ausgeben > Alle Datensätze > Seriendruck > Listen/Berichte drucken	32796
Ausgeben > Statistik	Ausgeben > Auswertungen > Statistik	32906
Ausgeben > Sofortbericht	Ausgeben > Auswertungen > Sofortbericht	33209
Termin > Neuer Termin zum Datensatz	Termine/Aufgaben > Neu > Termin zum Datensatz	32833
Termin > Termine zum Datensatz anzeigen	Termine/Aufgaben > Anzeigen > Termine zum Datensatz	32931
Termin > Terminübersicht	Termine/Aufgaben > Anzeigen > Terminübersicht	32835
Termin > Neue Aufgabe zum Datensatz	Termine/Aufgaben > Neu > Aufgabe zum Datensatz	32834
Termin > Aufgaben zum Datensatz anzeigen	Termine/Aufgaben > Anzeigen > Aufgaben zum Datensatz	32932
Termin > Aufgabenübersicht	Termine/Aufgaben > Anzeigen > Aufgabenübersicht	32836
Termin > Termine / Aufgaben anzeigen	Termine/Aufgaben > Wechseln zu > Termine und Aufgaben	32848
Ansicht > Übersichtsliste	Start > Anzeigen > Ansicht > Übersichtsliste	32782

Ansicht > Eingabemaske	Start > Anzeigen > Ansicht > Eingabemaske	32783
Ansicht > Web	Start > Anzeigen > Ansicht > Web	32989
Ansicht > Bericht	Start > Anzeigen > Ansicht > Bericht	32815
Ansicht > Symbolleisten > Standard	*keine Entsprechung vorhanden*	59500
Ansicht > Symbolleisten > Zwischenablage	*keine Entsprechung vorhanden*	59505
Ansicht > Symbolleisten > Suche	*keine Entsprechung vorhanden*	59524
Ansicht > Symbolleisten > Navigation	*keine Entsprechung vorhanden*	59501
Ansicht > Symbolleisten > Sortierung	*keine Entsprechung vorhanden*	59502
Ansicht > Symbolleisten > Filter	*keine Entsprechung vorhanden*	59526
Ansicht > Symbolleisten > Manueller Filter	*keine Entsprechung vorhanden*	59504
Ansicht > Symbolleisten > Routenplanung	*keine Entsprechung vorhanden*	59510
Ansicht > Symbolleisten > Alphabetischer Filter	*keine Entsprechung vorhanden*	59520
Ansicht > Symbolleisten > Statusleiste	Fenster > Anzeigen > Statusleiste	59393
Ansicht > Symbolleisten > Anpassen	Fenster > Anpassen > Menüband	33152
Ansicht > Wechseln zu > Ansicht öffnen	Fenster > Wechseln zu > Ansicht	33186
Ansicht > Wechseln zu > Info-Zentrale	Fenster > Wechseln zu > InfoZentrale	33131
Ansicht > Wechseln zu > Termine und Aufgaben	Fenster > Wechseln zu > Termine und Aufgaben	33091
Ansicht > Wechseln zu > Anrufe	Fenster > Wechseln zu > Anrufe	33127
Ansicht > Aktualisieren	Fenster > Ansicht > Aktualisieren	32784
Extras > Script > Ausführen	Start > Aktionen > Scripte & Workflows > Script ausführen bzw. Konfigurieren > Verwalten > Scripte >Ausführen	32934
Extras > Script > Bearbeiten	Konfigurieren > Verwalten > Scripte > Bearbeiten	32935
Extras > Script > Neu	Konfigurieren > Verwalten > Scripte > Neu	32936
Extras > Script > Verschlüsseln	Konfigurieren > Verwalten > Scripte > Verschlüsseln	33125
Extras > Workflow > Ausführen	Start > Aktionen > Scripte & Workflows > Workflow ausführen bzw. Konfigurieren > Verwalten > Workflows > Ausführen	33174
Extras > Workflow > Bearbeiten	Konfigurieren > Verwalten > Workflows > Bearbeiten	33175
Extras > Workflow > Neu	Konfigurieren > Verwalten > Workflows > Neu	33176
Extras > 01. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 01. Programm / Script / Workflow	32951
Extras > 02. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 02. Programm / Script / Workflow	32952
Extras > 03. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 03. Programm / Script / Workflow	32953
Extras > 04. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 04. Programm / Script / Workflow	32954
Extras > 05. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 05. Programm / Script / Workflow	32955
Extras > 06. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 06. Programm / Script / Workflow	32956
Extras > 07. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 07. Programm / Script / Workflow	32957
Extras > 08. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 08. Programm / Script / Workflow	32958
Extras > 09. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 09. Programm / Script / Workflow	32959
Extras > 10. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 10. Programm / Script / Workflow	32960

Extras > 11. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 11. Programm / Script / Workflow	32961
Extras > 12. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 12. Programm / Script / Workflow	32962
Extras > 13. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 13. Programm / Script / Workflow	32963
Extras > 14. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 14. Programm / Script / Workflow	32964
Extras > 15. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 15. Programm / Script / Workflow	32965
Extras > 16. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 16. Programm / Script / Workflow	32966
Extras > 17. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 17. Programm / Script / Workflow	32967
Extras > 18. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 18. Programm / Script / Workflow	32968
Extras > 19. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 19. Programm / Script / Workflow	32969
Extras > 20. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 20. Programm / Script / Workflow	32970
Extras > 21. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 21. Programm / Script / Workflow	32971
Extras > 22. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 22. Programm / Script / Workflow	32972
Extras > 23. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 23. Programm / Script / Workflow	32973
Extras > 24. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 24. Programm / Script / Workflow	32974
Extras > 25. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 25. Programm / Script / Workflow	32975
Extras > 26. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 26. Programm / Script / Workflow	32976
Extras > 27. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 27. Programm / Script / Workflow	32977
Extras > 28. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 28. Programm / Script / Workflow	32978
Extras > 29. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 29. Programm / Script / Workflow	32979
Extras > 30. Programm / Script / Workflow	Start > Aktionen > Scripte & Workflows > 30. Programm / Script / Workflow	32980
Konfigurieren > Ansicht	Konfigurieren > Ansicht > Eigenschaften	33156
Konfigurieren > Eingabemaske	Konfigurieren > Ansicht > Eingabemaske	32785
Konfigurieren > Eingabemaske WebAccess	Konfigurieren > Ansicht > Eingabemaske > Eingabemaske WebAccess	33208
Konfigurieren > Übernahmemaske > Auswählen	Konfigurieren > Vorlagen > Übernahmemasken > Auswählen	32924
Konfigurieren > Übernahmemaske > Bearbeiten	Konfigurieren > Vorlagen > Übernahmemasken > Bearbeiten	32863
Konfigurieren > eMail Vorlage	Konfigurieren > Vorlagen > eMails	32813
Konfigurieren > Telefon- & Online-Verzeichnisse / Routenplanung	Konfigurieren > Verwalten > Verzeichnisse+Routenplaner	32818
Konfigurieren > Druck > Etiketten	Konfigurieren > Vorlagen > Etiketten	32917
Konfigurieren > Druck > Karteikarten	Konfigurieren > Vorlagen > Karteikarten/Briefe	32918
Konfigurieren > Druck > Listen	Konfigurieren > Vorlagen > Listen/Berichte	32919
Konfigurieren > Druck > Direkter Einzel-/Serienbrief	Konfigurieren > Vorlagen > Direkte Einzel-/Serienbriefe	32921

Konfigurieren > Sofortberichte	Konfigurieren > Verwalten > Sofortberichte	33207

4.3 Termine & Aufgaben Menü

Befehl in Symbolleiste	Befehl in Menüband	ID
Datensatz > Neuer Termin	Start > Neu > Termin	33001
Datensatz > Neue Aufgabe	Start > Neu > Aufgabe	33002
Datensatz > Bearbeiten	Termin > Bearbeiten > Bearbeiten	33010
Datensatz > Delegieren	Termin > Bearbeiten > Delegieren	33012
Datensatz > Löschen	Termin > Bearbeiten > Löschen	33008
Datensatz > Alle Löschen > Termine	Start > Daten > Löschen > Alle Termine löschen	33046
Datensatz > Alle Löschen > Aufgaben	Start > Daten > Löschen > Alle Aufgaben löschen	33047
Datensatz > Gehe zu Datensatz	Start > Gehe zu > Datensatz	33014
Filtern > Benutzer	Start > Filtern > Benutzer	33027
Filtern > Termine	Start > Filtern > Termine	33043
Filtern > Aufgaben	Start > Filtern > Aufgaben	33044
Organisieren > Termine sortieren	Konfigurieren > Einstellungen > Sortierungen > Termine	33029
Organisieren > Aufgaben sortieren	Konfigurieren > Einstellungen > Sortierungen > Aufgaben	33030
Organisieren > Importieren	Start > Daten > Importieren	33191
Organisieren > Exportieren	Start > Daten > Exportieren	33192
Ausgeben > Termin-Liste	Ausgeben > Ausgeben > Listen > Termine	32997
Ausgeben > Aufgaben-Liste	Ausgeben > Ausgeben > Listen > Aufgaben	32998
Ausgeben > Termin-Karteikarten	Ausgeben > Ausgeben > Karteikarten > Termine	32999
Ausgeben > Aufgaben-Karteikarten	Ausgeben > Ausgeben > Karteikarten > Aufgaben	33000
Ausgeben > Kalenderansicht	Ausgeben > Ausgeben > Kalender	33195
Ansicht > Tagesansicht	Start > Ansicht > Tag	33006
Ansicht > Arbeitswochenansicht	Start > Ansicht > Arbeits-Woche	33005
Ansicht > Wochenansicht	Start > Ansicht > Woche	33085
Ansicht > Monatsansicht	Start > Ansicht > Monat	33004
Ansicht > Terminliste	Start > Ansicht > Terminliste	33009
Ansicht > Aufgabenliste	Start > Ansicht > Aufgaben	33003
Ansicht > Symbolleisten > Standard	*keine Entsprechung*	59507
Ansicht > Symbolleisten > Navigation	*keine Entsprechung*	59508
Ansicht > Symbolleisten > Filter	*keine Entsprechung*	59509
Ansicht > Symbolleisten > Statusleiste	*keine Entsprechung*	
Ansicht > Aktualisieren	Fenster > Ansicht > Aktualisieren	33007
Ansicht > Zu Heute wechseln	Start > Gehe zu > Heute	33011
Konfigurieren > Allgemein	Konfigurieren > Einstellungen > Optionen	33023
Konfigurieren > Ressourcen	Konfigurieren > Einstellungen > Ressourcen	33018
Konfigurieren > Kategorieren	Konfigurieren > Einstellungen > Kategorien	33250
Konfigurieren > Druck > Listen > Termine	Konfigurieren > Druckvorlagen > Listen > Termine	33019
Konfigurieren > Druck > Listen > Aufgaben	Konfigurieren > Druckvorlagen > Listen > Aufgaben	33020
Konfigurieren > Druck > Karteikarten > Termine	Konfigurieren > Druckvorlagen > Karteikarten > Termine	33021

Konfigurieren > Druck > Karteikarten > Aufgaben	Konfigurieren > Druckvorlagen > Karteikarten > Aufgaben	33022
Konfigurieren > Druck > Kalenderansicht	Konfigurieren > Druckvorlagen > Kalender	33199

4.4 Kontextmenüs

Kontextmenü - Container-Datensatz	**ID**
Neu	32845
Neues Dokument > Über Dokumentenverwaltung erzeugen	32903
Neues Dokument > Bestehende Datei	32904
Neues Dokument > Von Scanner einlesen	33028
Bearbeiten	32846
In neuem Fenster bearbeiten	33367
Löschen	32847
Feld in Zwischenablage kopieren	32858
Dokument mit verknüpfter Anwendung öffnen (Eingebettete Datei)	33034
Dokument mit verknüpfter Anwendung bearbeiten (Eingebettete Datei)	33033
Dokument mit verknüpfter Anwendung öffnen	32861
Dokumentenpfad in Zwischenablage kopieren	33167
Dokumentenverzeichnis öffnen	33169
Dokumentenverzeichnis in Zwischenablage kopieren	33168

Kontextmenü - Telefon-Feld	**ID**
Telefonwahl	32849
Aufnehmen in phone manager	32850
Telefontermin planen	32852

Kontextmenü - Mobiltelefon-Feld	**ID**
Telefonwahl	32849
Aufnehmen in phone manager	32850
Telefontermin planen	32852

Kontextmenü - Internet-Feld	**ID**
Internetbrowser aufrufen	32854

Kontextmenü - Mail-Feld	**ID**
eMail senden	32856
eMail-Termin planen	32855

Kontextmenü - Feld-Clipboard	**ID**
Suchen	32947
Ausschneiden	32857
Kopieren	32858
Einfügen	32859
Löschen	32860
Programmaufruf	33051

Häufigkeitsstatistik über Feld	33115

Kontextmenü - Eingebettete Datei-Feld	**ID**
Dokument einfügen	33032
Dokument mit verknüpfter Anwendung öffnen	33034
Dokument mit verknüpfter Anwendung bearbeiten	33033
Dokument löschen	33035

Kontextmenü - Grafik-Feld	**ID**
Grafik auswählen	32929
Grafik mit verknüpfter Anwendung öffnen	32928
Grafik entfernen	33258

Kontextmenü - Combobox-Feld	**ID**
Eintrag zur Auswahl hinzufügen	32944

Kontextmenü - Browse-Konfiguration	**ID**
Konfigurieren > Layout und Spalten	5012
Konfigurieren > Spalten-Eigenschaften	5011
Konfigurieren > Als Projekteinstellung für alle Anwender speichern	32984
Konfigurieren > Auf Projektvoreinstellung zurücksetzen	32985

Kontextmenü - RTF-Feld	**ID**
Rückgängig	32909
Ausschneiden	32910
Kopieren	32911
Einfügen	32912
Löschen	32913
Alles markieren	32908

Kontextmenü - Browse-Konfiguration time manager	**ID**
Konfigurieren > Layout und Spalten	5012
Konfigurieren > Spalten-Eigenschaften	5011

Kontextmenü - Datei-Feld	**ID**
Dokument auswählen	32862
Dokument mit verknüpfter Anwendung öffnen	32861
Dokumentenpfad in Zwischenablage kopieren	33167
Dokumentenverzeichnis öffnen	33169
Dokumentenverzeichnis in Zwischenablage kopieren	33168

Kontextmenü - Eingebettete Grafik-Feld	**ID**
Grafik einfügen	33032
Grafik mit verknüpfter Anwendung öffnen	33034
Grafik mit verknüpfter Anwendung bearbeiten	33033
Grafik löschen	33035

Kontextmenü - DMS-Feld	ID
Dokument in DMS einfügen	33032
DMS-Dokument mit verknüpfter Anwendung öffnen	33034
DMS-Dokument mit verknüpfter Anwendung bearbeiten	33033
Dokument im DMS löschen	33035
Verweis auf DMS-Dokument lösen	33045

Kontextmenü - Container-Konfiguration	ID
Konfigurieren > Layout und Spalten	5012
Konfigurieren > Spalten-Eigenschaften	5011
Konfigurieren > Als Projekteinstellung für alle Anwender speichern	32984
Konfigurieren > Auf Projektvoreinstellung zurücksetzen	32985
Sortieren > Aufsteigend	32868
Sortieren > Absteigend	32869
Sortieren > < ohne Sortierung >	32885
Sortieren > 1	32870
Sortieren > 2	32871
Sortieren > 3	32872
Sortieren > 4	32873
Sortieren > 5	32874
Sortieren > 6	32875
Sortieren > 7	32876
Sortieren > 8	32877
Sortieren > 9	32878
Sortieren > 10	32879
Sortieren > 11	32880
Sortieren > 12	32881
Sortieren > 13	32882
Sortieren > 14	32883
Sortieren > 15	32884
Sortieren > Nicht gruppieren	33348
Sortieren > Einfach gruppieren	33349
Sortieren > Mehrfach gruppieren	33350

Kontextmenü - Zusammenführen	ID
Kopieren	32858

Kontextmenü - Überwachungsergebnisse	ID
Gehe zu Datensatz	33121
Eintrag löschen	33247
Alle Einträge zu diesem Datensatz löschen	33248
Alle Einträge löschen	33249

4.5 phone manager Menü

Befehl in Symbolleiste	Befehl in Menüband	ID
Datei > Beenden	Datei > Beenden	32799
Bearbeiten > Eintrag löschen	Start > Bearbeiten > Eintrag löschen	32777
Bearbeiten > Eintrag verschieben	Start > Bearbeiten > Eintrag verschieben	32778
Bearbeiten > Alle Einträge löschen	Start > Bearbeiten > Eintrag löschen > Alle Einträge lö-schen	32782
Bearbeiten > Zur Anwendung wechseln	Start > Bearbeiten > Zur Anwendung wechseln	32781
Bearbeiten > Protokoll im Datensatz hinzufügen	Start > Bearbeiten > Protokoll im Datensatz hinzufügen	32800
Ansicht > Standard	*keine Entsprechung vorhanden*	59392
Wählen > Anrufen	Start > Wählen > Anrufen	32774
Wählen > Anhalten	Start > Wählen > Anhalten	32775
Wählen > Nächster	Start > Wählen > Nächster	32776
Wählen > Manuell	Start > Wählen > Manuell	32779
Wählen > Powerdialing	Start > Wählen > Powerdialing	32794
Ausgeben > Anrufliste	Start > Ausgeben > Anrufliste	106
Extras > Anrufsimulation	Datei > Optionen > Allgemein > Anrufsimulation	32787
Extras > Netzwerkfunktionen aktivieren	Datei > Optionen > Allgemein > Netzwerkfunktionen aktivieren	103
Konfigurieren > Wahlhilfe	Datei > Optionen > Allgemein > Wahlhilfe	32773
Konfigurieren > Telefonanlage	Datei > Optionen > Allgemein > Telefonanlage	32791
Konfigurieren > Wahlhilfeeinstellungen	Datei > Optionen > Allgemein > Wahlhilfeeinstellungen	32789
Konfigurieren > Anruferkennung	Datei > Optionen > Allgemein > Anruferkennung	32790
Konfigurieren > Akustische Signale	Datei > Optionen > Akustische Signale	32795
Konfigurieren > Ausgabe Anrufliste	Datei > Optionen > Allgemein > Ausgabe Anrufliste	101
? > Hilfe zum phone manager	Datei > Hilfe	32784
? > Über	*keine Entsprechung vorhanden*	57664

5 Ereignisse

Die Ereignis Verwaltung ermöglicht es auf verschiedene Ereignisse mit automatischen Aktionen zu reagieren. Hierzu können für jedes der angebotenen Ereignisse Scripte oder Workflows hinterlegt werden. Scripte und Workflows können per Datei verknüpft werden, alternativ können Scripte auch direkt im Projekt eingebettet werden.

Einschränkungen für diese Ereignis-Scripte:

Das Recht "Darf Scripte ausführen" wird bei diesen Scripten nicht abgefragt (ist also nicht wirksam), damit eine über Ereignis-Scripte konfigurierte Anwendungslogik in jedem Fall durchlaufen wird.

Es erfolgt keine Statusanzeige (Animation) in der Statusleiste.

Es gibt projektspezifische und ansichtsspezifische Ereignisse sowie Ereignisse für Termine und Aufgaben. Die nachfolgenden Ereignisse stehen Ihnen im Projekt unter Konfigurieren >Projekt > Ereignisse zur Verfügung.

> Wichtig: Bei aufwändigen Scripten (lange Laufzeit) empfiehlt es sich, asynchrone Scripte für die Ereignisse zu verwenden, da ansonsten die combit Anwendung zu lange blockiert wird.
>
> Ausgenommen hiervon sind jedoch die beiden Ereignisse "Projekt wird geschlossen" und "Ansicht wird geschlossen", hier muss ein synchrones Script verwendet werden. Generell müssen Sie bei asynchronen Scripten unbedingt berücksichtigen, dass der Anwender in der Zwischenzeit weiterarbeiten (und zum Beispiel den aktuellen Datensatz wechseln) kann. Dies kann je nach Script unvorhersehbare Effekte haben!

5.1 Event Objekt

Bei den ausgeführten Scripten steht ein neues Objekt Event zur Verfügung, welches Informationen über das aktuelle Ereignis zur Verfügung stellt. Auf dieses Objekt wird über das WScript Objekt zugegriffen:

```
WScript.Event
```

> Hinweis: Bei Workflows steht das Event Objekt nicht zur Verfügung.

5.1.1 Eigenschaften

Cancel

Beschreibung:

Sofern der das Ereignis auslösende Vorgang einen Abbruch unterstützt, kann dieser durch Setzen dieser Eigenschaft auf True vorgenommen werden. Eine genaue Beschreibung finden Sie in diesem Fall bei den jeweiligen Ereignissen.

Typ:

Bool

ChangedFields

Beschreibung:

Das Objekt ist für das Ereignis "Feldänderung" gültig.

Bei den Ereignissen "Feldänderung" und "Zeitüberschreitung" ist Record1 der betroffene Datensatz.

Typ:

ListEventFieldChange

Data

Beschreibung:

Enthält ggf. eine Zusatzinformation zum Ereignis.

Typ:

String

Project

Beschreibung:

Enthält das Project Objekt um das es im Ereignis geht.

Typ:

Project

Record1

Beschreibung:

Dieses Objekt zeigt ggf. auf den am Ereignis beteiligten Datensatz. Eine genaue Beschreibung finden Sie in diesem Fall bei den jeweiligen Ereignissen.

Typ:

Record

Record2

Beschreibung:

Dieses Objekt zeigt ggf. auf den am Ereignis beteiligten Datensatz. Eine genaue Beschreibung finden Sie in diesem Fall bei den jeweiligen Ereignissen.

Typ:

Record

View

Beschreibung:

Enthält das View Objekt, um das es im Ereignis geht (sofern beim Ereignis verfügbar).

Typ:

View

5.2 Projektspezifische Ereignisse

5.2.1 Projekt wurde geöffnet

Das Ereignis wird ausgelöst, wenn das Projekt geöffnet wurde.

5.2.2 Projekt wird geschlossen

Das Ereignis wird ausgelöst, bevor die Anwendung sich die aktuell offenen Ansichten merkt (um sie beim nächsten Start wiederherstellen zu können).

5.2.3 Eingehender Anruf

Das Ereignis wird ausgelöst, wenn ein Anruf eingeht.

- Data: enthält eingehende Nummer (ohne Sonderzeichen).

5.2.4 Eingehender Anruf wurde gesucht

Wird erst nach der durchgeführten Rufnummernsuche (phone manager) ausgelöst. Falls keine Datensätze gefunden wurden, hat WScript.Event.View den Wert Nothing, ansonsten zeigt es auf das View Objekt mit den Treffern. Falls ein derartiges Ereignis eingerichtet wurde, wird seitens der combit Anwendung keine "Datensatz wurde nicht gefunden" Meldung mehr ausgegeben, um Automatisierungen zu ermöglichen. Die Verwendung finden Sie im nachfolgenden Beispiel.

C# Script:

```
// Aktives Projekt

Project currentProject = cRM.CurrentProject;

combit.cRM.COM.View activeView = currentProject.ActiveViews.ActiveView;

ListViewConfigs viewConfigs = currentProject.ViewConfigs;

ViewConfig viewConfig = viewConfigs.ItemByName(activeView.Name);

RecordSet recordSet = viewConfig.CreateRecordSet();



// Filter erstellen

bool returnFilter = recordSet.SetFilter("upper(\"PLZ\") >= upper(\"70000\"
AND upper (\"PLZZ\") <= upper(\"80000\")", false);
```

```
if (returnFilter)

{

    // Überprüfung, ob Treffer

    if (recordSet.MoveFirst())

    {

        // Treffer vorhanden & Auswahl des Datensatzes durch den Benutzer

        Record record = recordSet.DialogSelectRecord("Bitte wählen Sie den
gewünschten Datensatz aus.");


        // Überprüfung, ob ein Datensatz ausgewählt wurde

        if (record != null)

        {

            MessageBox.Show("Es wurde folgender Datensatz ausgewählt:" + "\n"
+

                "Name: " + record.GetContentsByName("Name") + ", " +
record.GetContentsByName("Firstname") + "\n" +

                "PLZ: " + record.GetContentsByName("ZIP"), cRM.AppTitle,
MessageBoxButtons.OK);

        }

        else

        {

            MessageBox.Show("Es wurde kein Datensatz ausgewählt.",
cRM.AppTitle, MessageBoxButtons.OK);

        }

    }

    else

    {

        MessageBox.Show("Keine Datensätze für diesen Filter vorhanden!",
cRM.AppTitle, MessageBoxButtons.OK);

    }

}

else

{

    MessageBox.Show("Es konnte kein Filter erstellt werden!", cRM.AppTitle,
MessageBoxButtons.OK);

}
```

Verwendung des Events "Eingehender Anruf wurde gesucht"'.

- Data: enthält eingehende Nummer (ohne Sonderzeichen).

5.2.5 Menübefehl wird ausgeführt

Das Ereignis wird ausgeführt, wenn ein Menübefehl ausgeführt wird.

- Data: <MenuID> (siehe Kapitel "Projekt-Menü" für die entsprechenden Menü-IDs)

5.2.6 Menübefehl verstecken

Das Ereignis versteckt einen Menübefehl.

- Data: <MenuID> (siehe Kapitel "Projekt-Menü" für die entsprechenden Menü-IDs)

5.3 Termine und Aufgaben Ereignisse

5.3.1 Gemeinsame Ereignisse

Diese Ereignisse sollten nur einmal entweder für Termine oder Aufgaben definiert werden.

5.3.1.1 Menübefehl wird ausgeführt

Das Ereignis wird ausgeführt, wenn ein Menübefehl ausgeführt wird.

- Data: <MenuID> (siehe Kapitel "Termine & Aufgaben Menü" für die entsprechenden Menü-IDs)

5.3.1.2 Menübefehl verstecken

Das Ereignis versteckt einen Menübefehl.

- Data: <MenuID> (siehe Kapitel "Termine & Aufgaben Menü" für die entsprechenden Menü-IDs)

Beispiel zum Verstecken eines Menüpunkts aus einem Container-Kontextmenü

VBScript:

```
Option Explicit

Dim dataItems

dataItems = Split(WScript.Event.Data, "|", -1, 1)

Dim menuID

menuID = dataItems(0)

Dim relationContainerID

relationContainerID = dataItems(1)

If (relationContainerID = "ID.Aktivitäten.CompanyID#{45F5CAD2-73A1-4D48-B21E-
F38D27093D12}") Then

    MsgBox "Sie dürfen hier im Aktivitäten-Container keinen neuen Datensatz anlegen.
Daher wird die Aktion nun abgebrochen.", vbOKOnly + vbInformation, "Hinweis/Info"

    WScript.Event.Cancel = True

End If
```

C# Script:

```
string[] dataItems = WScript.Event.Data.Split('|');

if (dataItems.Length > 1)

{

    string menuID = dataItems[0];

    string relationContainerID = dataItems[1];



    if (relationContainerID == "ID.Aktivitäten.CompanyID#{45F5CAD2-73A1-4D48-B21E-
F38D27093D12}")

    {

        MessageBox.Show("Sie dürfen im Aktivitäten-Container keinen neuen Datensatz
anlegen. Daher wird die Aktion nun abgebrochen.", cRM.AppTitle + " - Hinweis/Info" ,
MessageBoxButtons.OK, MessageBoxIcon.Information);

        WScript.Event.Cancel = true;

    }

}
```

5.3.2 Termine

5.3.2.1 Termin wurde gespeichert

- Data: enthält die ID des Termins. Zugriff auf den Termin erhält man mit der Methode ItemByUniqueID des time manager Appointment Objekts.

Beispiel VBScript:

```
Dim oEvent, oTimeManager, oAppointment

Set oEvent = WScript.Event

Set oTimeManager = cRM.CurrentProject.timemanager

Set oAppointment = oTimeManager.Appointments.ItemByUniqueID(CStr(oEvent.Data))

MsgBox oAppointment.Start & ": " & oAppointment.Subject
```

Beispiel C# Script:

```
object appointmentSaved = WScript.Event;

TimeManager timeManager = cRM.CurrentProject.TimeManager;

Appointment appointment =
timeManager.Appointments.ItemByUniqueID(appointmentSaved.Data);

MessageBox.Show(appointment.Start.ToString() + ": " + appointment.Subject,
cRM.AppTitle, MessageBoxButtons.OK);
```

5.3.3 Aufgaben

5.3.3.1 Aufgabe wurde gespeichert / Aufgabe wurde erledigt

- Data: enthält die ID der Aufgabe. Zugriff auf die Aufgabe erhält man mit der Methode ItemByUniqueID des time manager ToDo Objekts.

Beispiel VBScript:

```
Dim oEvent, oTimeManager, oToDo

Set oEvent = WScript.Event

Set oTimeManager = cRM.CurrentProject.timemanager

Set oToDo = oTimeManager.ToDos.ItemByUniqueID(CStr(oEvent.Data))

MsgBox oToDo.Start & ": " & oToDo.Subject
```

Beispiel C# Script:

```
object toDoSaved = WScript.Event;

TimeManager timeManager = cRM.CurrentProject.TimeManager;

ToDo toDo = timeManager.ToDos.ItemByUniqueID(toDoSaved.Data);

MessageBox.Show(toDo.Start.ToString() + ": " + toDo.Subject, cRM.AppTitle, MessageBox-
Buttons.OK);
```

5.4 Ansichtenspezifische Ereignisse

5.4.1 Ansicht wurde geöffnet

Die Ansicht, in der das Ereignis konfiguriert wurde, wurde geöffnet.

5.4.2 Ansicht wird geschlossen

Die Ansicht wird geschlossen.

5.4.3 Datensatz wird gespeichert

Der Datensatz wird gespeichert (betrifft das interaktive Speichern eines Datensatzes in der Eingabemaske).

- Data: Wenn "0" dann ist es ein bestehender Datensatz, wenn "1" dann ist es ein neuer Datensatz.

Hinweis: WScript.Event.Record1 enthält den Datensatz kurz bevor er gespeichert wird. Über GetContents...() und SetContents...() Methoden können die Felder noch vor der tatsächlichen Speicherung gelesen/verändert werden. Lock(), Unlock(), Save() darf für den Record nicht aufgerufen werden.

Hinweis: Das Speichern eines Datensatzes kann durch Setzen der Eigenschaft Cancel des Event Objekts auf "True" unterbunden werden.

Wichtig: Ein Schreibzugriff auf CurrentRecord ist in diesem Event nicht möglich! Der Event wird nicht beim direkten Ändern in Übersichtsliste/Container ausgelöst.

5.4.4 Datensatz wurde gespeichert

Der Datensatz wurde gespeichert (betrifft das interaktive Speichern eines Datensatzes in der Eingabemaske).

- Data: Wenn "0" dann ist es ein bestehender Datensatz, wenn "1" dann ist es ein neuer Datensatz.

Hinweis: Der Event wird nicht beim direkten Ändern in Übersichtsliste/Container ausgelöst.

5.4.5 Datensatz wird zusammengeführt

Der Datensatz wird zusammengeführt.

- Data: Record1 stellt den Primär-, Record2 den Sekundär-Datensatz dar.

Hinweis: Das Speichern eines Datensatzes kann durch Setzen der Eigenschaft Cancel des Event Objekts auf "True" unterbunden werden.

5.4.6 Datensatz wurde zusammengeführt

Der Datensatz wurde zusammengeführt.

- Data: Record1 stellt den Ergebnis-Datensatz dar.

5.4.7 Feldänderung

Ein Feld wurde geändert. Siehe die Eigenschaft ChangedFields im Event Objekt.

- Data: Record1 stellt den betroffenen Datensatz dar.

5.4.8 Zeitüberschreitung

Eine Zeitüberschreitung ist eingetreten.

- Data: Record1 stellt den betroffenen Datensatz dar.

5.4.9 Neuer Datensatz wird bearbeitet

Wird ausgelöst, sobald in der Eingabemaske ein neuer Datensatz anlegt wird. Kann z.B. zum Vorbelegen von Feldern bei neuen Datensätzen per Script verwendet werden.

Beispiel VBScript:

```
Option Explicit

Dim oProject, oActiveView, oCurrentInputForm
```

```
' Aktives cRM Projekt

Set oProject = cRM.CurrentProject

Set oActiveView = oProject.ActiveViews.ActiveView

Set oCurrentInputForm = oActiveView.CurrentInputForm(0)

' Feld "ABC" in der Ansicht 'Firmen' auf "A" setzen

oCurrentInputForm.SetContentsByName CStr("ABC"), CStr("A")
```

Beispiel C# Script:

```
// Aktives Projekt

Project currentProject = cRM.CurrentProject;

combit.cRM.COM.View activeView = currentProject.ActiveViews.ActiveView;

InputForm currentInputForm = activeView.CurrentInputForm(0);


// Feld "ABC" in der Ansicht "Firmen" auf "A" setzen

currentInputForm.SetContentsByName("ABC", "A");
```

5.4.10 Menübefehl wird ausgeführt

Das Ereignis wird ausgeführt, wenn ein Menübefehl ausgeführt wird.

Menü:

- Data: <MenuID> (siehe Kapitel "Ansichten-Menü" für die entsprechenden Menü-IDs)

Relationen-Container:

- Data: <MenuID>|<RelationFieldName#RelationContainerID>|<FieldName>|<FieldContent> (siehe Kapitel "Kontextmenüs" für die entsprechenden Menü-IDs)

Termin-Container:

- Data: <MenuID>|:AppointmentsContainer (siehe Kapitel "Kontextmenüs" für die entsprechenden Menü-IDs)

Aufgaben-Container:

- Data: <MenuID>|:TodosContainer (siehe Kapitel "Kontextmenüs" für die entsprechenden Menü-IDs)

Funktionsdefinitionen:

- Data: <MenuID>|<RelationFieldName#RelationContainerID>

> Hinweis: Hierbei handelt es sich um die Funktionsdefinitionen, welche als Schaltflächen im Eingabemaskendesigner platziert werden können - bspw. "Container.Löschen" oder "Funktionsdefinition Übernahmemaske". Dabei werden diese Funktionsdefinitionen auf entsprechend passende Menübefehle interpretiert - bspw. wird für die Funktionsdefinition "Container.Löschen" der Menüpunkt "Kontextmenü - Container-Datensatz: Löschen" (ID: 32847) verwendet. Siehe hierzu auch Kapitel "Ansichten-Menü" sowie Kapitel "Kontextmenüs" für die entsprechenden Menü-IDs.

Beispiel zum Verhindern des Löschen-Menübefehls (Menü-ID 32774)

VBScript:

```
MsgBox "Löschen ist nicht erlaubt!"

WScript.Event.Cancel = True ' Verhindern des Befehls
```

C# Script:

```
MessageBox.Show("Löschen ist nicht erlaubt!", cRM.AppTitle, MessageBoxButtons.OK);

WScript.Event.Cancel = true; // Verhindern des Befehls
```

5.4.11 Menübefehl verstecken

Das Ereignis versteckt einen Menübefehl.

Menü:

- Data: <MenuID> (siehe Kapitel "Menü-IDs" für die entsprechenden Menü-IDs)

Relationen-Container:

- Data: <MenuID>|<RelationFieldName#RelationContainerID>|<FieldName>|<FieldContent> (siehe Kapitel "Kontextmenüs" für die entsprechenden Menü-IDs)

Termin-Container:

- Data: <MenuID>|:AppointmentsContainer (siehe Kapitel "Kontextmenüs" für die entsprechenden Menü-IDs)

Aufgaben-Container:

- Data: <MenuID>|:TodosContainer (siehe Kapitel "Kontextmenüs" für die entsprechenden Menü-IDs)

> Hinweis: Das Verstecken von Menübefehlen in einem bestimmten Container bezieht sich immer auf alle Container dieser Ansicht.

Beispiel zum Verstecken des Suchen-Menübefehls (Menü-ID 32810)

VBScript:

```
WScript.Event.Data = "32810"
```

C# Script:

```
WScript.Event.Data = "32810";
```

Beispiel zum Verstecken eines Menüpunkts aus allen Container-Kontextmenüs

VBScript:

```
Option Explicit

Dim dataItems

dataItems = Split(WScript.Event.Data, "|", -1, 1)

Dim menuID

menuID = dataItems(0)

Dim relationContainerID
```

```
relationContainerID = dataItems(1)

If (relationContainerID = "ID.Aktivitäten.CompanyID#{45F5CAD2-73A1-4D48-B21E-
F38D27093D12}") Then

    MsgBox "Sie dürfen hier im Aktivitäten-Container keinen neuen Datensatz anlegen.
Daher wird die Aktion nun abgebrochen.", vbOKOnly + vbInformation, "Hinweis/Info"

    WScript.Event.Cancel = True

End If
```

C# Script:

```
string[] dataItems = WScript.Event.Data.Split('|');

if (dataItems.Length > 1)

{

    string menuID = dataItems[0];

    string relationContainerID = dataItems[1];



    if (relationContainerID == "ID.Aktivitäten.CompanyID#{45F5CAD2-73A1-4D48-B21E-
F38D27093D12}")

    {

        MessageBox.Show("Sie dürfen im Aktivitäten-Container keinen neuen Datensatz
anlegen. Daher wird die Aktion nun abgebrochen.", cRM.AppTitle + " - Hinweis/Info" ,
MessageBoxButtons.OK, MessageBoxIcon.Information);

        WScript.Event.Cancel = true;

    }

}
```

5.5 EventFieldChange Objekt

Einzelne Feldänderung.

5.5.1 Eigenschaften

ContentNew, read-only

Beschreibung:

Liefert den neuen Inhalt des geänderten Feldes zurück.

Typ:

Variant

ContentOld, read-only

Beschreibung:

Liefert den alten Inhalt des geänderten Feldes zurück.

Typ:

Variant

Fieldname, read-only

Beschreibung:

Liefert den Feldnamen des geänderten Feldes zurück.

Typ:

String

5.6 ListEventFieldChange Objekt

Liste von Feldänderungen.

5.6.1 Eigenschaften

Count, read-only

Beschreibung:

Liefert die Anzahl Feldänderungen zurück.

Typ:

Long

5.6.2 Methoden

Item

Beschreibung:

Liefert eine einzelne Feldänderung vom Typ EventFieldChange zurück.

Typ:

EventFieldChange

6 eMail Autopilot

Der eMail Autopilot startet bei einer gefundenen eMail ein konfiguriertes Script. Damit mit Hilfe des gestarteten Scripts das cRM-Objekt verwendet werden kann, können Sie entweder das konfigurierte cRM-Objekt vom Autopilot verwenden oder selbst im Script ein eigenes cRM-Objekt erstellen.

Eine ausführliche Beschreibung der Autopilot-Konfiguration finden Sie im Abschnitt "eMail Autopilot" im Kapitel "Workflows und Ereignisse" des Handbuchs.

Wichtig: Der eMail Autopilot steht erst ab der cRM Professional Edition zur Verfügung.

Bitte beachten Sie, dass bei Übergabe von String-Parametern diese immer mit CStr() formatiert werden müssen.

Beispiel: `Call Mail.Forward(CStr(<Neuer Betreff>))`

6.1 Attachment Objekt

6.1.1 Eigenschaften

Name

Beschreibung:

Dateiname des Dateianhangs.

Hinweis: Der Dateiname wird bei Simple MAPI im 8.3-Format zurückgeliefert.

Typ:

String

6.1.2 Methoden

SaveAs

Beschreibung:

Speichert die Datei im angegebenen Verzeichnis.

Parameter:

Parametername	Typ	Beschreibung
sPath	String	Speicherpfad

Rückgabewert:

Bool

6.2 Attachments Objekt

6.2.1 Eigenschaften

Count

Beschreibung:

Gibt die Anzahl an gefundenen Anhängen zurück.

Typ:

Int

6.2.2 Methoden

Item

Beschreibung:

Gibt das entsprechende Attachment-Objekt zurück. Es muss die Index-Nummer des Eintrages übergeben werden. Der Index geht von 0 bis Count-1.

Parameter:

Parametername	Typ	Beschreibung
Index	Long	Index-Nummer.

Typ:

Attachment

6.3 HostApp Objekt

Bietet Zugriff auf die Applikation und das Mail-Objekt.

6.3.1 Eigenschaften

Mail

Beschreibung:

Gibt die aktuelle eMail als Mail-Objekt zurück.

Typ:

Mail

StopFurtherAction

Beschreibung:

Wird diese Eigenschaft auf True gesetzt, werden keine weiteren Betreffzeilen ausgewertet. Das Programm fängt dann mit dem Bearbeiten der nächsten Mail an.

Rückgabewert:

Bool

Version

Beschreibung:

Gibt die Version zurück.

Rückgabewert:

Long

6.3.2 Methoden

AddProtocolEntry

Beschreibung:

Über diese Methode des HostApp-Objektes können eigene Ausgaben in die Protokolldatei des eMail Autopiloten hinzugefügt werden. Jeder hinzugefügte Eintrag in der Protokolldatei erhält den Präfix "Custom: ", um die eigenen Einträge identifizieren zu können.

Hinweis: Hierfür muss die Protokollierung im eMail Autopiloten aktiviert sein.

Parameter:

Parametername	Typ	Beschreibung
sLogEntry	String	Neuer Eintrag für die Protokolldatei.

Beispiel VBScript:

…

```
Call HostApp.AddProtocolEntry(CStr("<Eigener Protokolleintrag>"))
```

…

Beispiel C# Script:

…

```
HostApp.AddProtocolEntry("<Eigener Protokolleintrag>");…
```

SetOutputString

Beschreibung:

Dient zur Ausgabe von Zwischenschritten innerhalb eines Scripts. Der Text wird in der Hauptansicht des Programms angezeigt.

Hinweis: Der bisherige Ausgabetext wird gelöscht!

Parameter:

Parametername	Typ	Beschreibung
sOutputString	String	Statusanzeige im Programm

SetOutputStringAppend

Beschreibung:

Dient zur Ausgabe von Zwischenschritten innerhalb eines Scripts. Der Text wird in der Hauptansicht des Programms angezeigt. Der Statustext wird an die bestehende Ausgabe anhängt.

Parameter:

Parametername	Typ	Beschreibung
sOutputString	String	Statusanzeige im Programm.

6.4 Mail Objekt

Bietet Zugriff auf eine vom Autopilot erkannte eMail.

6.4.1 Eigenschaften

Attachments

Beschreibung:

Liefert einen ggf. vorhandenen Mailanhang als Attachments-Objekt zurück. Um zu überprüfen, ob die eMail Anhänge hat, muss das angelegte Objekt mit `Is Nothing` auf Gültigkeit überprüft werden.

Typ:

Attachments

BodyMessage

Beschreibung:

Liefert den kompletten Nachrichtentext zurück.

Rückgabewert:

String

Recipients

Beschreibung:

Liefert eine semikolon-separierte Liste der eMail-Empfänger zurück.

Rückgabewert:

String

SendDateTime

Beschreibung:

Gibt das Sendedatum inkl. Uhrzeit der eMail zurück. Das Format wird gemäß den Windows Systemeinstellungen zurückgegeben, d.h. bei einer Regionseinstellung "Deutsch" erfolgt die Ausgabe umgerechnet in die lokale Zeit im Format: 16.07.2012 11:09:04

Rückgabewert:

String

Sender

Beschreibung:

Name des Absenders inklusive eMail-Adresse.

Rückgabewert:

String

Beispiel VBScript:

```
Set oMailObject = HostApp.Mail

sMailSubject = oMailObject.Subject

sMailSender = oMailObject.Sender

HostApp.SetOutputString("Betreff der eMail: " & sMailSubject)

HostApp.SetOutputStringAppend("Absender der eMail: " & sMailSender)
```

Beispiel C# Script:

```
Mail mailObject = HostApp.Mail;

string mailSubject = mailObject.Subject;

string mailSender = mailObject.Sender;

HostApp.SetOutputString("Betreff der eMail: " + mailSubject);

HostApp.SetOutputStringAppend("Absender der eMail: " + mailSender);
```

SenderAddressResolved

Beschreibung:

eMail Adresse des Absenders (wird nach RFC822 Standard für Internet-Mails ermittelt).

Rückgabewert:

String

Subject

Beschreibung:

Gibt den Betreff der eMail zurück.

Rückgabewert:

String

6.4.2 Methoden

Delete

Beschreibung:

Löscht die eMail physikalisch aus dem Postfach.

Forward

Beschreibung:

Leitet die gesamte eMail an den konfigurierten eMail-Empfänger weiter. Ausführliche Informationen zur Konfiguration des eMail Betreffs finden Sie unter "eMail Betreff" im Abschnitt "eMail Autopilot" im Kapitel "Workflows und Ereignisse" des Handbuchs.

Hinweis: Wenn keine Parameter angeben sind, so werden die im Programm konfigurierten Werte verwendet.

Parameter:

Parametername	Typ	Beschreibung
sSubject	String	Neuer Betreff der zu weiterleitenden eMail (optional).
sRecipients	String	eMail-Adresse des Empfängers (optional).

Rückgabewert:

Bool

SaveAs

Beschreibung:

Speichert die gesamte eMail inkl. Anhang als Datei.

Parameter:

Parametername	Typ	Beschreibung
sFile	String	Pfad und Dateiname der Maildatei. Als Dateiendung muss *.eml* verwendet werden!

Rückgabewert:

Bool

SetMarkAsRead

Beschreibung:

Über diese Methode des Mail-Objektes können eMails bei Simple MAPI und Extended MAPI als gelesen markiert werden. Im Konfigurationsdialog ist hierfür eine Option ("eMail als gelesen markieren") vorhanden, um das Standard-Verhalten steuern zu können.

Hinweis: Diese Methode hat nur Auswirkungen, wenn Simple MAPI oder Extended MAPI verwendet wird. Für SMTP/POP3 ist diese Methode irrelevant.

Beispiel VBScript:

…

```
Call HostApp.Mail.SetMarkAsRead()
```

…

Beispiel C# Script:

…

```
HostApp.Mail.SetMarkAsRead();
```

…

7 Änderungen und Neuerungen in der Version 8

7.1 Eigenschaften und Methoden

7.1.1 Eigenschaften

Version 8.002

- 'AllDayEvent' im Appointment Objekt

7.1.2 Methoden

Version 8.000

- 'AppendFileExt2' im DocMngr Objekt
- 'BindFctHandler' im DialogForm Objekt [C#]
- 'BindTimerHandler' im DialogForm Objekt [C#]
- 'DialogSelectRecordMultiple' im RecordSet Objekt
- 'ConvertPlainText2HTML' im Application Objekt
- 'ConvertHTML2PlainText' im Application Objekt
- 'SendMail' im Application Objekt
- 'GetContentsValueByName' im Record Objekt
- 'GetContentsValueByFormula' im Record Objekt
- 'SetContentsValueByName' im Record Objekt
- 'SetContentsValueByFormula' im Record Objekt
- 'CurrentRecord' im RecordSet Objekt verhält sich nun analog zu 'CurrentRecordSynchronized'

Version 8.002

- 'GetContentsValueByName' im InputForm Objekt
- 'SetContentsValueByName' im InputForm Objekt
- EditMailTemplate im View Objekt
- EditReportTemplate im View Objekt
- RowIDFldName im ViewConfig Objekt
- Optionaler Parameter 'Files' für Methoden SendMailDialog, SendMailDirect, SendSingleMailDialog im Record und RecordSet Objekt

Version 8.006

- Optionaler Parameter 'Files' für Methode SendSingleMail im Record Objekt

Version 8.007

- GetIniProfileString im Application Objekt
- SetIniProfileString im Application Objekt

8 Änderungen und Neuerungen in der Version 7

8.1 Eigenschaften und Methoden

8.1.1 Eigenschaften

Version 7.000

- 'UserDefined1' im User Objekt
- 'UserDefined2' im User Objekt

8.1.2 Methoden

Version 7.000

- 'ExecuteInstantReportByDescription' im Project Objekt
- 'ExecuteInstantReportByName' im Project Objekt
- 'GetGlobalProperty' im Project Objekt
- 'GetUserProperty' im Project Objekt
- 'SetGlobalProperty' im Project Objekt
- 'SetUserProperty' im Project Objekt
- 'ShowField' im InputForm Objekt

Version 7.006

- 'SendBulkMail' im RecordSet Objekt

8.2 Ereignisse

Version 7.000

- Ansichtenspezifisches Ereignis 'Neuer Datensatz wird bearbeitet'
- Projektspezifisches Ereignis, Termine und Aufgaben Ereignis, Ansichtenspezifisches Ereignis 'Menübefehl wird ausgeführt'
- Projektspezifisches Ereignis, Termine und Aufgaben Ereignis, Ansichtenspezifisches Ereignis 'Menübefehl wird ausgeführt'

9 Änderungen und Neuerungen in der Version 6

9.1 Objekte

Version 6.000

- 'Attendee' Objekt
- 'Attendees' Objekt
- 'Categories' Objekt
- 'Category' Objekt
- 'Link' Objekt
- 'Links' Objekt
- 'DialogForm' Objekt

Version 6.008

- 'ListWebElements' Objekt
- 'WebElement' Objekt

9.2 Eigenschaften und Methoden

9.2.1 Eigenschaften

Version 6.000

- 'DialogForm' im Application Objekt
- 'Edition' im Application Objekt
- 'DBSystemVersion' im Project Objekt
- 'Attendees' im Appointment Objekt
- 'Categories' im Appointment Objekt
- 'Links' im Appointment Objekt
- 'Attendees' im ToDo Objekt
- 'Categories' im ToDo Objekt
- 'Links' im ToDo Objekt
- 'DialogForm' im Application Objekt

Version 6.006

- 'TempDir' im Application Objekt

Version 6.007

- 'ProjectDir' im Project Objekt

Version 6.008

- 'WebElements' im InputForm Objekt
- 'ID' im WebElement Objekt

Version 6.010

- 'AppTitle' im Application Objekt

9.2.2 Methoden

Version 6.000

- Optionaler Parameter 'UseModalPreviewWindow' für Methoden PrintCard, PrintLabel, PrintReport im Record und RecordSet Objekt
- 'SendRecordRef' im Record Objekt
- 'SendMailDialog' im RecordSet Objekt
- 'SyncWithActiveDirectory' im Users Objekt
- 'ExecuteInstantReportByName' im RecordSet Objekt

Version 6.006

- 'DialogSelectDir' im Application Objekt
- 'DialogSelectFile' im Application Objekt
- 'TransferData' im Record Objekt
- 'CreateTempFile' im Application Objekt

Version 6.008

- 'ExecuteScriptByCode' im Project Objekt
- 'ExecuteScriptByFilename' im Project Objekt
- 'ItemByName' im WebElements Objekt
- 'IE' im WebElement Objekt

9.3 Ereignisse

Version 6.000

- Eigenschaft 'Event.ChangedFields' im Event Objekt für Ereignis 'Feldänderung'
- Objekt 'ListEventFieldChange'
- Objekt 'EventFieldChange'

10 Änderungen und Neuerungen in der Version 5

10.1 Objekte

Version 5.000

- 'Container' Objekt
- 'ListContainers' Objekt
- 'SQLShell' Objekt

Version 5.003

- 'ListCodeDefinitions' Objekt

10.2 Eigenschaften und Methoden

10.2.1 Eigenschaften

Version 5.000

- 'Containers' im InputForm Objekt
- 'FileVersion' im Application Objekt
- 'UILanguageID' im Application Objekt
- 'ViewMode' im View Objekt

Version 5.004

- 'MainWindowHandle' im Application Objekt

Version 5.007

- 'Importance' im Appointment Objekt

10.2.2 Methoden

Version 5.000

- 'CurrentRecord' im Container Objekt
- 'ExecuteWorkflowByFilename' im Project Objekt
- 'ExportAsiCal' im Appointment Objekt
- 'ExportAsiCal' im ToDo Objekt

- 'FindRecordByEmail' im RecordSet Objekt
- 'FldAlias' im ViewConfig Objekt
- 'GetcRMByProcessID' im Application Objekt
- 'ItemByName' im ListContainers Objekt
- 'PrintCard', 'PrintLabel', 'PrintReport' im Record Objekt
- 'SetSessionProperty' im Project Objekt
- 'Sleep' im WScript Objekt

Version 5.003

- 'CodeDefinitions' im ViewConfig Objekt
- 'Count' im ListCodeDefinitions Objekt
- 'Item' im ListCodeDefinitions Objekt

Version 5.004

- 'CheckVATID' im Project Objekt

Version 5.005

- 'SetWaitDlgText' im Application/WScript Objekt

Version 5.006

- 'ConvertToLatestVersion' im InputForm Objekt
- 'AddProtocolEntry' im HostApp Objekt des eMail Autopilot
- 'SetMarkAsRead' im Mail Objekt des eMail Autopilot

Version 5.009

- 'AddToFavorites' im Record Objekt
- 'AddToHistory' im Record Objekt

10.3 Ereignisse

Version 5.006

- Ansichtenspezifisches Ereignis 'Datensatz wird gespeichert' enthält jetzt den Datensatz kurz bevor dieser gespeichert wird

Version 5.007

- Eigenschaft 'Event.Cancel' im Event Objekt für Ereignis 'Datensatz wird gespeichert'

11 Webansicht

Sie können für jede Ansicht oder für die Infozentrale eine Internetseite oder eine eigene HTML-Seite hinterlegen. Die Anzeige innerhalb einer Ansicht erfolgt über Ansicht > Web. Die HTML-Seiten können hierbei mit jedem beliebigen HTML-Editor erstellt werden. Die darzustellende Webseite wird in der Ansichtskonfiguration angegeben.

Ausführliche Informationen zur Konfiguration der Webansicht finden Sie unter "Projekteigenschaften" im Kapitel "Konfiguration Projekt" bzw. unter "Allgemein" im Kapitel "Konfiguration Ansichten" des Handbuchs.

Hinweis: C# Scripte stehen in der Webansicht nicht zur Verfügung.

11.1 Dynamische Inhalte in HTML-Seiten

Die Anzeige von dynamischen Informationen (z.B. aus der combit Anwendung) innerhalb einer HTML-Seite geschieht über eine Scriptfunktion. Die Scriptfunktion ermittelt den dynamischen Inhalt und gibt ihn über das Document Object Model (DOM) des Webbrowsers mittels document.write Kommando aus.

Hinweis: Im Document Object Model (DOM) des W3-Konsortiums ist das document-Objekt das Ausgangsobjekt für den Elementbaum. Eine ausführliche Beschreibung finden Sie z.B. unter http://de.selfhtml.org

```
<head>

<script language="VBScript">

<!--

Function Test()

    document.write("<b> Hallo </b>")

End Function

//-->

</script>

</head>

<body>

<p><script language="VBScript">Test</script></p>

</body>

</html>
```

11.1.1 Zugriff auf das Application-Objekt

Der Zugriff auf das Application-Objekt der combit Anwendung erfolgt innerhalb eines Scripts über das window.external Objekt. Sie können nun wie in normalen Scripten mit allen COM-Objekten der combit Anwendung arbeiten.

```
Dim ocRM

Set ocRM = window.external
```

11.1.2 Beispiel: Name des Projekts anzeigen

```
<html>

<head>

<title>Beispiel</title>

<script language="VBScript">

<!--

Function Display_Project

    Dim ocRM, oProject

    Set ocRM = window.external

    Set oProject = ocRM.CurrentProject

    document.write oProject.Name

     Set oProject = Nothing

     Set ocRM = Nothing

End Function

//-->

</script>

</head>

<body>

<h1>Aktuelles Projekt:</h1>

<p><script language="VBScript">Display_Project</script></p>

</body>

</html>
```

Tipp: Um Informationen in der HTML-Ansicht zu "speichern" oder über mehrere HTML-Seiten hinweg zu erhalten, stehen Ihnen im Project-Objekt der combit Anwendung die Methoden SetUserProperty() und GettUserProperty() zur Verfügung. Hierüber können Werte persistent gespeichert und geladen werden. Dies wird z.B. für die Speicherung/Abfrage des Menü-Navigationsstatus in der Datei "script.js" verwendet.

12 Web-Elemente

Es bestehen die folgenden drei Möglichkeiten, ein Web-Element in der Eingabemaske zu nutzen bzw. anzusteuern.

Hinweis: Die Webseiten werden dabei in einem Internet Explorer Control angezeigt. Dieses läuft intern im Kompatibilitätsmodus (siehe auch "<meta http-equiv="X-UA-Compatible" content="IE=edge" />") wodurch es Unterschiede/Einschränkungen insbesondere bei der Darstellung von CSS im Vergleich zur Internet Explorer Standalone-Anwendung geben kann.

Bei Problemen mit der Darstellung einer Webseite, besonders wenn es sich um eine Seite handelt, welche den Inhalt per Script dynamisch erzeugt (siehe Beispiele in den Referenzlösungen), kann es hilfreich sein, die Option "Scriptfehler anzeigen" in den Eigenschaften des entsprechenden Web-Element Objekts zu aktivieren.

Hinweis: C# Scripte stehen in der Webansicht nicht zur Verfügung.

Wichtig: Bitte stellen Sie sicher, dass Web-Elemente bei Verwendung bestimmter Methoden (z.B. PrintLabel) prüfen, ob sich der aktuelle Datensatz im Bearbeitungsmodus befindet und in diesem Fall nichts weiter machen.

12.1 Direkt per URL-Aufruf

Im Eigenschaftsdialog des Web-Elements im Eingabemaskendesigner geben Sie dazu für die Eigenschaft URL die Adresse einer Webseite an und übergeben ggf. weitere Parameter.

Am Beispiel des Web-Elements "PDF" in der mitgelieferten Large Solution sähe dies wie folgt aus:

```
%PRJDIR%\Web-Elemente\showPDF\showPDF.html?pdfFile=%PRJDIR%\Produkt1.pdf
```

12.2 Innerhalb einer Webseite

Der Zugriff auf das Application-Objekt der combit Anwendung erfolgt innerhalb einer Webseite über das window.external.Application Objekt.

```
var oApplication = window.external.Application
```

Der Zugriff auf das "zugehörige" View-Objekt der combit Anwendung erfolgt innerhalb einer Webseite über das window.external.View Objekt.

```
var oView = window.external.View
```

Der Zugriff auf den Namen bzw. die ID des Web-Elements erfolgt innerhalb einer Webseite über window.external.Name.

```
var sWebElementName = window.external.Name
```

Wichtig: Beachten Sie hierbei, dass der Zugriff auf das View-Objekt nur auf dem vorgenannten Weg durchgeführt werden darf, nicht aber über das ActiveView-Objekt des Application-Objekts der combit Anwendung, da ein Web-Element seinen Inhalt immer asynchron lädt und ausführt.

12.3 Aus einem Script

Der Zugriff auf ein Web-Element in einem Script (innerhalb oder außerhalb der combit Anwendung aufgerufen) erfolgt über die WebElements Eigenschaft des InputForm-Objekts.

Siehe die entsprechenden Kapitel ListWebElements Objekt und WebElement Objekt.

13 FAQ

Im nachfolgenden Kapitel finden Sie einige Beispiele für die Verwendung von VBScripts mit der combit Anwendung in Form von Scriptauszügen.

Die nachfolgenden Scriptvorlagen sowie einige weitere Scriptvorlagen finden Sie im Unterverzeichnis "Scripts" des jeweiligen Solution-Verzeichnisses Ihrer Installation.

Bitte beachten Sie auch dass, um Scripte außerhalb der combit Anwendung starten zu können, das Application-Objekt "cRM.Application" benötigt wird. Siehe auch Beispiel "cRM.Application Instanz erzeugen" sowie den entsprechenden Eintrag im Kapitel "Scripte und Programmierreferenz" des Handbuchs. Die folgenden Scripte mit Ausnahme des vorgenannten Beispiels sind nur innerhalb der combit Anwendung lauffähig.

13.1 Script-Beispiele

13.1.1 cRM.Application Instanz erzeugen

VBScript:

```
Option Explicit


' Variablen deklarieren

Dim ocRM, oProject

Dim sUser, sProject, sPassword


' Beachten Sie bitte, dass hier ggf. die Benutzerdaten auf Ihre Installation

' angepasst werden müssen, falls die combit Anwendung neu gestartet werden
muss

sUser = "Administrator"

sPassword = ""

sProject = "C:\Programme\combit\cRM\Solutions\Medium\Medium.crm"


' Kontrollieren ob die combit Anwendung bereits läuft und ggf. starten

On Error Resume Next


    ' Auf bereits aktive combit Anwendung zugreifen

    Set ocRM = GetObject(,"cRM.application")

    If Err <> 0 Then

        Err.Clear


        ' Neue Instanz erzeugen
```

```
        Set ocRM = CreateObject("cRM.application")


        ' Login

        Set oProject = ocRM.Login (sProject, sUser, sPassword)

    Else

        Set oProject = ocRM.CurrentProject

    End If

On Error GoTo 0


' combit Anwendung anzeigen

ocRM.Visible = True


MsgBox "Projektname: " & oProject.Name, vbInformation, "combit Anwendung"


' Objekte freigeben

Set oProject = Nothing

Set ocRM = Nothing
```

13.1.2 Überprüfung, ob Ansichten aktiv sind

VBScript:

```
Option Explicit


' Aktives Projekt

Dim oProject, oListView

Set oProject = cRM.CurrentProject

Set oListView = oProject.ActiveViews


' Überprüfung, ob Ansichten aktiv sind

If oListView.Count > 0 Then

    MsgBox "Es ist/sind: " & oListView.Count & " Ansichten aktiv.", vbInforma-
tion, "combit Anwendung"

Else

    MsgBox "Es sind keine Ansichten aktiv.", vbInformation, "combit Anwendung"

End If
```

```
' Objekte freigeben

Set oListView = Nothing

Set oProject = Nothing
```

C# Script:

```
// Überprüfung, ob Ansichten aktiv sind

Project currentProject = cRM.CurrentProject;
ListViews activeViews = currentProject.ActiveViews;

if (activeViews.Count > 0)
{
    MessageBox.Show("Es ist/sind: " + activeViews.Count.ToString() + " An-
sichten aktiv.", cRM.AppTitle, MessageBoxButtons.OK);
}
else
{
    MessageBox.Show("Es sind keine Ansichten aktiv.", cRM.AppTitle, Message-
BoxButtons.OK);

}
```

13.1.3 Auf die aktiven Ansichten eines Projektes zugreifen

VBScript:

```
Option Explicit



' Aktives Projekt

Dim oProject, oListViews, oView

Set oProject = cRM.CurrentProject

Set oListViews = oProject.ActiveViews

Set oView = oListViews.ActiveView



MsgBox "Aktive Ansicht: " & oView.Name, vbInformation, "combit Anwendung"



' .... <eigener Scriptcode> ....



' Objekte freigeben:

Set oView = Nothing

Set oListViews = Nothing

Set oProject = Nothing
```

C# Script:

```
// Aktives Projekt

Project currentProject = cRM.CurrentProject;

ListViews activeViews = currentProject.ActiveViews;

combit.cRM.COM.View activeView = activeViews.ActiveView;


MessageBox.Show("Aktive Ansicht: " + activeView.Name, cRM.AppTitle,
MessageBoxButtons.OK);


// .... <eigener Scriptcode> ....
```

13.1.4 Auf eine konfigurierte Ansicht eines Projektes zugreifen

VBScript:

```
Option Explicit


' Name der konfigurierten Ansicht

Const sView = "Kontakte"


' Aktives Projekt

Dim oProject, oListViewConfigs

Set oProject = cRM.CurrentProject

Set oListViewConfigs = oProject.ViewConfigs


Dim oViewConfig

Set oViewConfig = oListViewConfigs.ItemByName(CStr(sView))


MsgBox "Datenbank-Tabellenname der Ansicht " & sView & ": " & oViewCon-
fig.DBTableName, vbInformation, "combit Anwendung"


' .... <eigener Scriptcode> ....


' Objekte freigeben

Set oViewConfig = Nothing

Set oListViewConfigs = Nothing
```

```
Set oProject = Nothing
```

C# Script:

```
// Name der konfigurierten Ansicht

string viewName = "Kontakte";


// Aktives Projekt

Project currentProject = cRM.CurrentProject;

ListViewConfigs viewConfigs = currentProject.ViewConfigs;

ViewConfig viewConfig = viewConfigs.ItemByName(viewName);


MessageBox.Show("Datenbank-Tabellenname der Ansicht " + viewName + ": " +
viewConfig.DBTableName, cRM.AppTitle, MessageBoxButtons.OK);


// .... <eigener Scriptcode> ....
```

13.1.5 Einen Datensatz in einer konfigurierten Ansicht anlegen

VBScript:

```
Option Explicit


' Name der konfigurierten Ansicht

Const sView = "Firmen"


' Aktives Projekt

Dim oProject, oListViewConfigs

Set oProject = cRM.CurrentProject

Set oListViewConfigs = oProject.ViewConfigs


Dim oViewConfig

Set oViewConfig = oListViewConfigs.ItemByName(CStr(sView))


Dim oRecordSet

Set oRecordSet = oViewConfig.CreateRecordSet
```

```
' Neuen Datensatz anlegen

Dim oRecord

Set oRecord = oRecordSet.NewRecord


' Datensatz sperren

oRecord.Lock


' Zeichenfeld schreiben

oRecord.SetContentsByName "Company", "Muster GmbH"


' Aktuelles Datum in internes COM Datumsformat schreiben

' YYYYMMDD - Datum / YYYYMMDDHHMMSS - Datum mit Uhrzeit

oRecord.SetContentsByName "CreditQstDate", CStr(ConvertDateTocRMDate(Date))


' Numerisches Feld schreiben

oRecord.SetContentsByName "TurnoverTarget", CStr(2000)


' Logisches Feld setzen

oRecord.SetContentsByName "CreditOK", CStr("1")


' Datensatz speichern und Schreibvorgang prüfen

If oRecord.Save = True Then

    MsgBox "Datensatz wurde geschrieben!", vbInformation, "combit Anwendung"

Else

    MsgBox "Datensatz konnte nicht geschrieben werden!", vbExclamation, "com-
bit Anwendung"

End If


' Datensatz entsperren

oRecord.Unlock


' Objekte freigeben

Set oRecord = Nothing

Set oRecordSet = Nothing
```

```
' Aktive Ansicht aktualisieren
cRM.CurrentProject.ActiveViews.ActiveView.Update

Set oViewConfig = Nothing
Set oListViewConfigs = Nothing
Set oProject = Nothing

'-----------------------------------------------------------------------------
' Funktion: Konvertierung eines 8-stelligen Datums in das interne COM Datums-
format der combit Anwendung
'-----------------------------------------------------------------------------
Function ConvertDateTocRMDate (myDate)
    ConvertDateTocRMDate = Mid(myDate,7,4) & Mid(myDate,4,2) & Left(myDate,2)
End Function
```

C# Script:

```
// Name der konfigurierten Ansicht
string viewName = "Firmen";

// Aktives Projekt
Project currentProject = cRM.CurrentProject;
ListViewConfigs viewConfigs = currentProject.ViewConfigs;
ViewConfig view = viewConfigs.ItemByName(viewName);

RecordSet recordSet = view.CreateRecordSet();

// Neuen Datensatz anlegen
Record record = recordSet.NewRecord();

// Datensatz sperren
record.Lock();

// Zeichenfeld schreiben
```

```
record.SetContentsByName("Company", "Muster GmbH");


// Aktuelles Datum in internes COM-Datumsformat schreiben

record.SetContentsByName("CreditQstDate", DateTime.Now.ToString("YYYYMMDD"));


// Numerisches Feld schreiben

record.SetContentsByName("TurnoverTarget", "2000");


// Logisches Feld setzen

record.SetContentsByName("CreditOK", "1");


// Datensatz speichern und Schreibvorgang prüfen

if (record.Save())

{

    MessageBox.Show("Datensatz wurde geschrieben!", cRM.AppTitle,
MessageBoxButtons.OK);

}

else

{

    MessageBox.Show("Datensatz konnte nicht geschrieben werden!",
cRM.AppTitle, MessageBoxButtons.OK);

}


// Datensatz entsperren

record.Unlock();


// Aktive Ansicht aktualisieren

cRM.CurrentProject.ActiveViews.ActiveView.Update();
```

13.1.6 Erstellen eines Filters

VBScript:

```
Option Explicit


' Aktives Projekt
```

```
Dim oProject, oActiveView, oCurrentRecordSet, bFilter

Set oProject = cRM.CurrentProject

Set oActiveView = oProject.ActiveViews.ActiveView

Set oCurrentRecordSet = oActiveView.CurrentRecordSet


' Filter erstellen für alle Datensätze im Postleitzahlenbereich 70000 - 80000

bFilter = oCurrentRecordSet.SetFilter("upper(""PLZZ"") >= upper('70000') AND
upper(""PLZZ"") <= upper('80000')")


If bFilter = True Then

    ' Überhaupt Treffer vorhanden?

    If oCurrentRecordSet.MoveFirst Then

        MsgBox "Anzahl Datensätze im Filter: " & oActiveView.FilterRecCount,
vbInformation, "combit Anwendung"

    Else

        MsgBox "Es konnten keine Datensätze gefunden werden!", vbInformation,
"combit Anwendung"

    End If

Else

    MsgBox "Es konnte kein Filter erstellt werden!", vbInformation, "combit
Anwendung"

End If


' Objekte freigeben

Set oCurrentRecordSet = Nothing

Set oActiveView = Nothing

Set oProject = Nothing
```

C# Script:

```
// Aktives Projekt

Project currentProject = cRM.CurrentProject;

combit.cRM.COM.View activeView = currentProject.ActiveViews.ActiveView;

RecordSet recordSet = activeView.CurrentRecordSet;


// Filter erstellen für alle Datensätze im Postleitzahlenbereich 70000 -
80000
```

```
bool returnFilter = recordSet.SetFilter("upper(\"PLZ\") >= upper(\"70000\"
AND upper (\"PLZZ\") <= upper(\"80000\")", false);


if (returnFilter)

{

    // Überhaupt Treffer vorhanden?

    if (recordSet.MoveFirst())

    {

        MessageBox.Show("Anzahl Datensätze im Filter: " +
activeView.FilterRecCount.ToString(), cRM.AppTitle, MessageBoxButtons.OK);

    }

    else

    {

        MessageBox.Show("Es konnten keine Datensätze gefunden werden!",
cRM.AppTitle, MessageBoxButtons.OK);

    }

}

else

{

    MessageBox.Show("Es konnte kein Filter erstellt werden!", cRM.AppTitle,
MessageBoxButtons.OK);

}
```

13.1.7 Auswahl eines Datensatzes durch den Benutzer

VBScript:

```
Option Explicit


' Aktives Projekt

Dim oProject, oActiveView, oViewConfigs, oViewConfig, oCurrentRecordSet,
oCurrentRecord

Set oProject = cRM.CurrentProject

Set oActiveView = oProject.ActiveViews.ActiveView

Set oViewConfigs = oProject.ViewConfigs

Set oViewConfig = oViewConfigs.ItemByName(oActiveView.Name)

Set oCurrentRecordSet = oViewConfig.CreateRecordSet
```

```
' Filter erstellen

Dim bFilter

bFilter = oCurrentRecordSet.SetFilter("upper(""PLZZ"") >= upper('70000') AND
upper(""PLZZ"") <= upper('80000')")


If bFilter = True Then


' Überprüfung, ob Treffer

    If oCurrentRecordSet.MoveFirst Then

        ' Treffer vorhanden!

        ' Auswahl des Datensatzes durch den Benutzer

        Set oCurrentRecord = oCurrentRecordSet.DialogSelectRecord("Bitte wäh-
len Sie den gewünschten Datensatz aus.")


        ' Überprüfung, ob ein Datensatz ausgewählt wurde

        If Not oCurrentRecord Is Nothing Then

            MsgBox "Es wurde folgender Datensatz ausgewählt:" & vbCrLf & _

            "Name: " & oCurrentRecord.GetContentsByName("Name") & ", " & _

            oCurrentRecord.GetContentsByName("Firstname") & vbCrLf & _

            "PLZ: " & oCurrentRecord.GetContentsByName("ZIP"), vbInformation,
"combit Anwendung"

        Else

            Msgbox "Es wurde kein Datensatz ausgewählt.", vbInformation, "com-
bit Anwendung"

        End If

    Else

        Msgbox "Keine Datensätze für diesen Filter vorhanden!", vbInformation,
"combit Anwendung"

    End If

Else

    Msgbox "Es konnte kein Filter erstellt werden!", vbInformation, "combit
Anwendung"

End If


' Objekte freigeben

Set oCurrentRecord = Nothing
```

```
Set oCurrentRecordSet = Nothing

Set oViewConfig = Nothing

Set oViewConfigs = Nothing

Set oActiveView     = Nothing

Set oProject = Nothing
```

C# Script:

```
// Aktives Projekt

Project currentProject = cRM.CurrentProject;

combit.cRM.COM.View activeView = currentProject.ActiveViews.ActiveView;

ListViewConfigs viewConfigs = currentProject.ViewConfigs;

ViewConfig viewConfig = viewConfigs.ItemByName(activeView.Name);

RecordSet recordSet = viewConfig.CreateRecordSet();


// Filter erstellen

bool returnFilter = recordSet.SetFilter("upper(\"PLZ\") >= upper(\"70000\"
AND upper (\"PLZZ\") <= upper(\"80000\")", false);


if (returnFilter)

{

    // Überprüfung, ob Treffer

    if (recordSet.MoveFirst())

    {

        // Treffer vorhanden & Auswahl des Datensatzes durch den Benutzer

        Record record = recordSet.DialogSelectRecord("Bitte wählen Sie den
gewünschten Datensatz aus.");


        // Überprüfung, ob ein Datensatz ausgewählt wurde

        if (record != null)

        {

            MessageBox.Show("Es wurde folgender Datensatz ausgewählt:" + "\n"
+

                "Name: " + record.GetContentsByName("Name") + ", " +
record.GetContentsByName("Firstname") + "\n" +

                "PLZ: " + record.GetContentsByName("ZIP"), cRM.AppTitle,
MessageBoxButtons.OK);
```

```
        }
        else
        {
            MessageBox.Show("Es wurde kein Datensatz ausgewählt.",
cRM.AppTitle, MessageBoxButtons.OK);
        }
    }
    else
    {
        MessageBox.Show("Keine Datensätze für diesen Filter vorhanden!",
cRM.AppTitle, MessageBoxButtons.OK);
    }
}
else
{
    MessageBox.Show("Es konnte kein Filter erstellt werden!", cRM.AppTitle,
MessageBoxButtons.OK);
}
```

13.1.8 Verwendung des Events "Eingehender Anruf wurde gesucht"

VBScript:

```
' Das Ereignis "Eingehender Anruf wurde gesucht",
' wird ausgelöst, wenn der phone manager einen entsprechenden
' Datensatz gefunden hat. Der nachfolgende Scriptcode für das Ereignis
' kann durch eine Anrufsimulation (Extras > Anrufsimulation) getestet werden.

If WScript.Event.View Is Nothing Then
    MsgBox WScript.Event.Data & " wurde nicht gefunden!"
Else
    If(MsgBox(WScript.Event.Data & " gefunden!" & _
        WScript.Event.View.FilterRecCount & " Treffer." & _
        vbCrLF & "Möchten Sie diese anzeigen lassen?", vbYesNo) = vbYes)
Then
        Msgbox "Gefundener Datensatz: " &_
WScript.Event.View.CurrentRecordSet.CurrentRecord.GetContentsByName("Company")
```

```
        ' <<< eigener Scriptcode >>>

    End If

End If
```

C# Script:

```
// Das Ereignis "Eingehender Anruf wurde gesucht",

// wird ausgelöst, wenn der phone manager einen entsprechenden

// Datensatz gefunden hat. Der nachfolgende Scriptcode für das Ereignis

// kann durch eine Anrufsimulation (Extras > Anrufsimulation) getestet werden.


if (WScript.Event.View == null)

{

    MessageBox.Show(WScript.Event.Data + " wurde nicht gefunden!",
cRM.AppTitle, MessageBoxButtons.OK);

}

else

{

    DialogResult result = MessageBox.Show(WScript.Event.Data + " gefunden!" +
WScript.Event.View.FilterRecCount + " Treffer." + "\n" +

        "Möchten Sie diese anzeigen?", cRM.AppTitle, MessageBoxButtons.YesNo);

    if (result == DialogResult.Yes)

    {

        MessageBox.Show("Gefundener Datensatz: " +
WScript.Event.View.CurrentRecordSet.CurrentRecord.GetContentsByName("Company")
, cRM.AppTitle, MessageBoxButtons.OK);

    }

}
```

13.1.9 Direkter Versand einer eMail in einer aktiven Ansicht

VBScript:

```
Option Explicit

Dim oProject, oActiveView, oCurrentRecordSet, oCurrentRecord


' Aktives Projekt

Set oProject = cRM.CurrentProject

Set oActiveView = oProject.ActiveViews.ActiveView
```

```
Set oCurrentRecordSet = oActiveView.CurrentRecordSet


' Aktueller Datensatz

Dim oRecord

Set oCurrentRecord = oCurrentRecordSet.CurrentRecord()


' Direkter Versand der eMail:

' 1. Parameter = eMaildadresse des Empfänger

' 2. Parameter = Subjekt (Betreff) der eMail

' 3. Parameter = Mailtext

' 4. Parameter = ein Dateianhang

(optional, kann leer gelassen werden CStr(""))

Dim bReturn

bReturn = oCurrentRecord.SendMailDirect(

   CStr(oCurrentRecord.GetContentsByName("Email")), CStr("Ihre
   Anfrage"), CStr("Mailtext"), CStr("c:\test.pdf"))


' Objekte freigeben

Set oCurrentRecord = Nothing

Set oCurrentRecordSet = Nothing

Set oActiveView = Nothing

Set oProject = Nothing
```

C# Script:

```
// Aktives Projekt

Project currentProject = cRM.CurrentProject;

combit.cRM.COM.View activeView = currentProject.ActiveViews.ActiveView;

RecordSet recordSet = activeView.CurrentRecordSet;


// Aktueller Datensatz

Record record = recordSet.CurrentRecordSynchronized;


// Direkter Versand der eMail

bool returnSendMailDirect =
record.SendMailDirect(record.GetContentsByName("Email"), "Ihre Anfrage",
```

```
"Mailtext", @"C:\Export.pdf");


if (returnSendMailDirect)

{

    MessageBox.Show("Der eMail-Versand wurde erfolgreich durchgeführt!",
cRM.AppTitle, MessageBoxButtons.OK);

}

else

{

    MessageBox.Show("Der eMail-Versand konnte nicht erfolgreich durchgeführt
werden!", cRM.AppTitle, MessageBoxButtons.OK);

}
```

13.1.10 Ein logisches Feld direkt in der aktiven Eingabemaske setzen (via InputForm Objekt)

VBScript:

```
Option Explicit

Dim oProject,  oActiveView, oCurrentInputForm


' Aktives Projekt

Set oProject = cRM.CurrentProject

Set oActiveView = oProject.ActiveViews.ActiveView


Set oCurrentInputForm = oActiveView.CurrentInputForm(0)


' Feld 'BonitätOK' in der Ansicht 'Firmen' - Registerkarte 'Vertrieb' aktivie-
ren:

oCurrentInputForm.SetContentsByName CStr("CreditOK"), CStr("T")


MsgBox "Die Checkbox wurde aktiviert: " & oCurrentInput-
Form.GetContentsByName(CStr("CreditOK"))


' Feld 'BonitätOK' in der Ansicht 'Firmen' - Registerkarte 'Vertrieb' wieder
deaktivieren

oCurrentInputForm.SetContentsByName CStr("CreditOK"), CStr("F")
```

```
' Objekte freigeben

Set oCurrentInputForm = Nothing

Set oActiveView = Nothing

Set oProject = Nothing
```

C# Script:

```
// Aktives Projekt

Project currentProject = cRM.CurrentProject;

combit.cRM.COM.View activeView = currentProject.ActiveViews.ActiveView;



InputForm currentInputForm = activeView.CurrentInputForm(0);



// Feld "BonitätOK" in der Ansicht "Firmen" - Registerkarte "Vertrieb" akti-
vieren:

currentInputForm.SetContentsByName("CreditOK", "T");

MessageBox.Show("Die Checkbox wurde aktiviert: " +
currentInputForm.GetContentsByName("CreditOK"), cRM.AppTitle,
MessageBoxButtons.OK);



// Feld "BonitätOK" in der Ansicht "Firmen" - Registerkarte "Vertrieb" wieder
deaktivieren:

currentInputForm.SetContentsByName("CreditOK", "F");
```

13.1.11 Filter-Ergebnis überprüfen

VBScript:

```
Option Explicit

Dim oProject, oActiveView, oCurrentRecordSet



' Aktives Projekt

Set oProject = cRM.CurrentProject

Set oActiveView = oProject.ActiveViews.ActiveView

Set oCurrentRecordSet = oActiveView.CurrentRecordSet



' Beliebigen Filter erstellen

if oCurrentRecordSet.SetFilter(...) then

' Filter syntaktisch OK
```

```
    if oCurrentRecordSet.MoveFirst then
        ' Es gibt Treffer
        if oCurrentRecordSet.RecCount > 1 then
            '... Achtung, kein eindeutiger Treffer, sondern mehrere!
        else
            ' eindeutiger Treffer
            ' .... <eigener Scriptcode> ....
        end if
    else
        ' keine Treffer
    end if
else
' Filter hat Syntax Fehler o.ä.
end if

' Objekte freigeben
Set oCurrentRecordSet = Nothing
Set oActiveView = Nothing
Set oProject = Nothing
```

C# Script:

```
// Aktives Projekt
Project currentProject = cRM.CurrentProject;
combit.cRM.COM.View activeView = currentProject.ActiveViews.ActiveView;
RecordSet recordSet = activeView.CurrentRecordSet;

// Beliebigen Filter erstellen
if (recordSet.SetFilter(...))
{
    // Filter syntaktisch OK
    if (recordSet.MoveFirst())
    {
        // Es gibt Treffer
        if (recordSet.RecCount > 1)
```

```
        {
            // ... Achtung, kein eindeutiger Treffer, sondern mehrere!
        }
        else
        {
            // eindeutiger Treffer

            // ..... <eigener Scriptcode> .....
        }
    }
    else
    {
        // keine Treffer
    }
}
else
{
    // Filter hat Syntaxfehler o.ä.
}
```

13.1.12 Erstellen eines Filters anhand einer Telefonnummer

VBScript:

```
Option Explicit
Dim oProject, oActiveView, oCurrentRecordSet, bFound, sPhoneNumber

' Aktives Projekt
Set oProject = cRM.CurrentProject
Set oActiveView = oProject.ActiveViews.ActiveView
Set oCurrentRecordSet = oActiveView.CurrentRecordSet

sPhoneNumber = InputBox("Zu suchende Telefonnummer", "Test Script")

bFound = oCurrentRecordSet.FindRecordByPhoneNumber(sPhoneNumber, True, false)
```

```
If bFound Then

   Msgbox "Anzahl Datensätze im Filter: " & oActiveView.FilterRecCount

Else

   Msgbox "Es konnten keine Datensätze gefunden werden!"

End If


' Objekte freigeben

Set oCurrentRecordSet = Nothing

Set oActiveView = Nothing

Set oProject = Nothing
```

C# Script:

```
<!—#include ref="Microsoft.VisualBasic.dll"-->

// Aktives Projekt

Project currentProject = cRM.CurrentProject;

combit.cRM.COM.View activeView = currentProject.ActiveViews.ActiveView;

RecordSet recordSet = activeView.CurrentRecordSet;


string phoneNumber = Microsoft.VisualBasic.Interaction.InputBox("Zu suchende
Telefonnummer", cRM.AppTitle, "07531 90 60 10");


bool phoneNumberFound = recordSet.FindRecordByPhoneNumber(phoneNumber, true,
false);


if (phoneNumberFound)

{

    MessageBox.Show("Anzahl Datensätze im Filter: " +
activeView.FilterRecCount, cRM.AppTitle, MessageBoxButtons.OK);

}

else

{

    MessageBox.Show("Es konnten keine Datensätze gefunden werden.",
cRM.AppTitle, MessageBoxButtons.OK);

}
```

13.1.13 Feldinhalt in einem aktiven Datensatz in der 'Firmen'-Ansicht ändern

VBScript:

```
Option Explicit

' Aktives Projekt
Dim oProject, oActiveViews, oActiveView
Set oProject = cRM.CurrentProject
Set oActiveViews = oProject.ActiveViews
Set oActiveView = oActiveViews.ActiveView

' Aktiven Filter in der aktiven Ansicht verwenden
Dim oRecordSet
Set oRecordSet = oActiveView.CurrentRecordSet

' Aktiven Datensatz verwenden
Dim oRecord
Set oRecord = oRecordSet.CurrentRecord

' Datensatz sperren
oRecord.Lock

' Feldinhalt auslesen
Dim sValue
sValue = oRecord.GetContentsByName("Company")

' Zeichenfeld schreiben
oRecord.SetContentsByName "Company2", CStr(sValue)

' Datensatz speichern und Schreibvorgang prüfen
If oRecord.Save = True Then
    MsgBox "Datensatz wurde geschrieben!", vbInformation, "combit Anwendung"
Else
    MsgBox "Datensatz konnte nicht geschrieben werden!", vbExclamation, "com-
```

```
bit Anwendung"
End If

' Datensatz entsperren
oRecord.Unlock

' Objekte freigeben
Set oRecord = Nothing
Set oRecordSet = Nothing

' Aktive Ansicht aktualisieren
oActiveView.Update

Set oActiveView = Nothing
Set oActiveViews = Nothing
Set oProject = Nothing
```

C# Script:

```
// Aktives Projekt
Project currentProject = cRM.CurrentProject;
ListViews activeViews = currentProject.ActiveViews;
combit.cRM.COM.View activeView = activeViews.ActiveView;

// Aktiven Filter in der aktiven Ansicht verwenden
RecordSet recordSet = activeView.CurrentRecordSet;

// Aktiven Datensatz verwenden
Record record = recordSet.CurrentRecordSynchronized;

// Datensatz sperren
record.Lock();

// Feldinhalt auslesen
string fieldContent = record.GetContentsByName("Company");
```

```
// Zeichenfeld schreiben

record.SetContentsByName("Company", fieldContent);


// Datensatz speichern und Schreibvorgang prüfen

if (record.Save())

{

    MessageBox.Show("Datensatz wurde geschrieben!", cRM.AppTitle,
MessageBoxButtons.OK);

}

else

{

    MessageBox.Show("Datensatz konnte nicht geschrieben werden!",
cRM.AppTitle, MessageBoxButtons.OK);

}


// Datensatz entsperren

record.Unlock();


// Aktive Ansicht aktualisieren

activeView.Update();
```

13.1.14 Einfügen einer vom eMail Autopilot erkannten eMail in den Dokumentcontainer der combit Anwendung

VBScript:

```
Option Explicit


' Name der konfigurierten Ansicht

Const sView = "Kontakte"

Const sDocView = "Aktivitäten"


' eMail Autopilot Objekte

Dim oMail

Set oMail = HostApp.Mail
```

```
Dim sMailSender, sMailBody, sMailSubject
sMailSender = oMail.SenderAddressResolved ' Absender
sMailBody = oMail.BodyMessage ' Nachrichtentext
sMailSubject = oMail.Subject ' Betreff

' Scripting File Objekt
Dim fso, stmpFileName
Set fso = CreateObject("Scripting.FileSystemObject")
stmpFileName = fso.getspecialfolder(2) & "\x.eml" ' Ablage in den Temp Ordner

Dim oProject, oListViewConfigs
Set oProject = cRM.CurrentProject
Set oListViewConfigs = oProject.ViewConfigs

Dim oViewConfig
Set oViewConfig = oListViewConfigs.ItemByName(CStr(sView))

' Nach der eMail Adresse suchen
Dim bFilter, sFilter
sFilter = "upper(""Kontakte"".""Email"") = upper('"& sMailSender &"')"

Dim oRecordSet, oRecord
Set oRecordSet = oViewConfig.CreateRecordSet

' Nach der eMail Adresse filtern
oRecordSet.SetFilter("")
bFilter = oRecordSet.SetFilter(sFilter)

If bFilter = True Then

    ' Überhaupt Treffer vorhanden?
    If oRecordSet.MoveFirst Then
```

```
        ' Ersten gefundenen Treffer verwenden
        Set oRecord = oRecordSet.CurrentRecord

        ' Relationsobjekt erzeugen
        Dim oListRelations
        Set oListRelations = oViewConfig.Relations

        Dim i, sRelationAliasName, oRelation
        For i = 1 To oListRelations.Count

            Set oRelation = oListRelations.Item(i)
            sRelationAliasName = oRelation.Alias

            ' Alias für die "Dokumenten" Relation ermitteln
            If sRelationAliasName = CStr(sDocView) Then

                ' eMail speichern
                oMail.SaveAs(stmpFileName)

                ' DocMngr Objekt anlegen
                Dim oDocMngr
                Set oDocMngr = oProject.DocMngr
                Dim bAppend

                ' Datei hinzufügen
                bAppend = oDocMngr.AppendFile(oRecord, oRelation, stmpFileNa-
me, "*.eml")

                If bAppend = False Then
                    HostApp.SetOutputStringAppend "Die eMail konnte nicht an-
gehängt werden!"
                Else
                    HostApp.SetOutputStringAppend "Die eMail wurde angehängt."
                End If
```

```
                ' Temporäre Maildatei wieder löschen
                fso.DeleteFile stmpFileName
                Exit For
            End If

            Set oRelation = Nothing

        Next

        Set oRecord = Nothing
        Set oListRelations = Nothing

    Else
        HostApp.SetOutputStringAppend "Es konnten keine Datensätze gefunden
werden!"
    End If
Else
    HostApp.SetOutputStringAppend "Es konnte kein Filter erstellt werden!"
End If

' Objekte freigeben
Set oViewConfig = Nothing
Set oListViewConfigs = Nothing
Set oProject = Nothing
Set oMail = Nothing
```

13.1.15 Aufgabe zu einem Datensatz anlegen

VBScript:

```
Option Explicit

' Globale Konstante
Const cDateBegin = "17.01.2012 08:00:00"
Const cDateEnd = "17.01.2012 08:15:00"
```

```
' Globale Variablen

Dim oProject, otm, oTask


' Erstellen der benötigten Objekte

Set oProject = cRM.CurrentProject

Set oTM = oProject.timemanager

Set oTask = oTM.ToDos.Add()


' Verknüpfung mit dem Datensatz in der aufrufenden Ansicht

Dim oView, oViewConfig, sViewFamilyName, sPKFieldName, oRecord

Set oView = cRM.CurrentProject.ActiveViews.ActiveView

Set oViewConfig = oView.Config

sViewFamilyName = oViewConfig.FamilyName

sPKFieldName = oViewConfig.PrimaryKeyFldName

Set oRecord =
cRM.CurrentProject.ActiveViews.ActiveView.CurrentRecordSet.CurrentRecord


' Beispiel eines Datensatzverweises:

' <ProjectID>|<ViewName>|<ViewFamilyName>|<PKField>|<PKValue>|<DisplayText>

Dim sRecordReference

sRecordReference = CStr(oProject.ID) & "|" & _

                CStr(oView.Name) & "|" & _

                CStr(sViewFamilyName) & "|" & _

                CStr(sPKFieldName) & "|" & _

                CStr(oRecord.GetContentsByName(CStr(sPKFieldName))) & "|" & _

                CStr(oRecord.GetContentsByName(CStr("Firstname")) & " " &
oRecord.GetContentsByName(CStr("Name")))


Dim sContact

sContact = CStr(oRecord.GetContentsByName("Firstname")) & " " & _

        CStr(oRecord.GetContentsByName("Name"))


' Setzen der Eigenschaften der Aufgabe
```

```
With oTask

    .Subject "Besprechung"

    .Body "Besprechung mit Max Mustermann" & Chr(13) & "Thema: Aufgabe per
VBScript anlegen"

    .Contact sContact

    .Start cDateBegin

    .End cDateEnd

    .HostDatabase = sRecordReference

End With


' Erinnerung 24h vorher

oTask.Reminder = True

oTask.ReminderMinutesBeforeStart = 1440  ' Angabe in Minuten


' Speichern der angelegten Aufgabe

oTask.Save

' alternativ kann man auch oTask.Display verwenden

' Hier würde dann der Dialog zur gesetzten Aufgabe nochmal angezeigt werden


' Ansicht aktualisieren

oTM.UpdateViews


Set oRecord = Nothing

Set oViewConfig = Nothing

Set oView = Nothing

Set oTask = Nothing

Set oTM = Nothing

Set oProject = Nothing
```

C# Script:

```
// Globale Konstanten

DateTime dateBegin = new DateTime(2012, 1, 17, 8, 0, 0);

DateTime dateEnd = new DateTime(2012, 1, 17, 8, 15, 0);
```

```
// Erstellen der benötigten Objekte

Project currentProject = cRM.CurrentProject;

TimeManager timeManager = currentProject.TimeManager;

ToDo toDo = timeManager.ToDos.Add();


// Verknüpfung mit dem Datensatz in der aufrufenden Ansicht

combit.cRM.COM.View activeView = currentProject.ActiveViews.ActiveView;

ViewConfig viewConfig = activeView.Config;

string viewFamilyName = viewConfig.FamilyName;

string primaryKeyFieldName = viewConfig.PrimaryKeyFldName;


Record record =
cRM.CurrentProject.ActiveViews.ActiveView.CurrentRecordSet.CurrentRecordSynchr
onized;


// Beispiel eines Datensatzverweises:

// <ProjectID>|<ViewName>|<ViewFamilyName>|<PKField>|<PKValue>|<DisplayText>


string contactInformation = record.GetContentsByName("Firstname") + " " +
record.GetContentsByName("Name");

string recordReference = currentProject.ID + "|" + viewConfig.Name + "|" +
viewFamilyName + "|" + primaryKeyFieldName + "|" +
record.GetContentsByName(primaryKeyFieldName) + "|" + contactInformation;


// Setzen der Eigenschaften der Aufgabe

toDo.Subject = "Besprechung";

toDo.Body = "Besprechung mit Max Mustermann \n" +

    "Thema: Aufgabe per VBScript anlegen";

toDo.Contact = contactInformation;

toDo.Start = dateBegin;

toDo.End = dateEnd;

toDo.HostDatabase = recordReference;


// Erinnerung 24h vorher

toDo.Reminder = true;

toDo.ReminderMinutesBeforeStart = 1440; // Angabe in Minuten
```

```
// Speichern der angelegten Aufgabe
toDo.Save();
// Alternativ kann man auch toDo.Display() verwenden
// Hier würde dann der Dialog zur gesetzten Aufgabe angezeigt werden

// Ansicht aktualisieren
timeManager.UpdateViews();
```

13.1.16 Alle Datensätze eines RecordSets durchlaufen

VBScript:

```
Option Explicit

' Objekte deklarieren
Dim oViewConfigs, oViewConfig, oRecordSet, oRecord
Dim iCountRecord
Dim sInformation

' Benutzereingabe: Ansicht, in der die Datensätze durchlaufen werden sollen
' & Feld, in dem der Inhalt ausgelesen werden soll
Const sView = "Firmen"
Const sField = "Company"

' Objekte initialisieren
Set oViewConfigs = cRM.CurrentProject.ViewConfigs
Set oViewConfig = oViewConfigs.ItemByName(sView)
Set oRecordSet = oViewConfig.CreateRecordSet

' Auf den ersten Datensatz des RecordSets springen
oRecordSet.MoveFirst

' Abfragen, wie viele Datensätze im RecordSet sind
iRecordCount = oRecordSet.RecCount
```

```
' Durch jeden Datensatz durchlaufen
For i = 1 To iRecordCount

' Record-Objekt zuweisen
    Set oRecord = oRecordSet.CurrentRecordSynchronized

' Information aus dem definierten Feld auslesen
    sInformation = sInformation & oRecord.GetContentsByName(sField) & "|"
    Set oRecord = Nothing
' Zum nächsten Datensatz springen
    oRecordSet.MoveNext
Next

Set oRecordSet = Nothing
Set oViewConfig = Nothing
Set oViewConfigs = Nothing
```

C# Script:

```
// Benutzereingabe: Ansicht, in der die Datensätze durchlaufen werden sollen
// & Feld, in dem der Inhalt ausgelesen werden soll
string viewToCheck = "Firmen";
string fieldToCheck = "Company";

// Objekte initialisieren
ListViewConfigs viewConfigs = cRM.CurrentProject.ViewConfigs;
ViewConfig viewConfig = viewConfigs.ItemByName(viewToCheck);
RecordSet recordSet = viewConfig.CreateRecordSet();

// Auf den ersten Datensatz des RecordSets springen
recordSet.MoveFirst();

// Abfragen, wie viele Datensätze im RecordSet sind
long recordCount = recordSet.RecCount;
```

```
string information = string.Empty;


// Durch jeden Datensatz durchlaufen

for (int i = 1; i <= recordCount; i++)

{

    // Record-Objekt zuweisen

    Record record = recordSet.CurrentRecordSynchronized;


    // Informationen aus dem definierten Feld auslesen

    information = information + record.GetContentsByName(fieldToCheck) + "|";


    // Zum nächsten Datensatz springen

    recordSet.MoveNext();

}
```

13.1.17 Ereignis 'Feldänderung' per Script verarbeiten

Siehe Script 'EventFieldChange.vbs' im Script-Verzeichnis der Solutions Events, Medium, Medium (English), Large.

13.1.18 Ereignis 'Zeitüberschreitung' per Script verarbeiten

Siehe Script 'EventTimeout.vbs' im Script-Verzeichnis der Solutions Events, Medium, Medium (English), Large.

14 COM-Schnittstelle für .NET

Die COM Schnittstelle der combit Anwendung kann auch über einen sog. Wrapper in .NET genutzt werden. Weitere Informationen hierzu finden Sie in der combit Knowledgebase unter folgendem Link: http://www.combit.net/knowledgebase/?action=display&articleid=KBAD001208

Beispiel:

```
// Neues Application Objekt erstellen

cRMApplication cRM = new
cRMApplication(EApplicationStartType.CreateAlwaysNewInstance);

Project proj = cRM.Login(@"C:\Programme\combit\cRM\Solutions\Large\Large.crm",
"Administrator", "");

// Wenn Login-Vorgang erfolgreich war

if (proj != null)

{

    MessageBox.Show(this, "Projektname: " + proj.Name + "\n" + "Projektpfad: " +
proj.FilePath, cRM.AppTitle, MessageBoxButtons.OK);

}

// COM-Objekte freigeben

proj.Dispose();
```

Hinweis: Bitte beachten Sie, dass combit hierfür keinen Support leistet!

15 combit Knowledgebase

Auf unserer Internetseite finden Sie eine Knowledgebase mit hunderten Artikeln mit Hintergrund-Informationen, Tipps und Tricks.

> Hinweis: Artikel zur COM-Schnittstelle finden Sie meistens unter dem Suchbegriff *COM*, *OLE* oder *Script*.

Schauen Sie mal rein: http://www.combit.net/crm-software/knowledgebase/