

# Waimea Report

Dokumentation für Entwickler

Gültig ab Version 8.0.0

27.04.2012

---

## Inhaltsverzeichnis

<b>1. Einleitung.....</b>	<b>6</b>
<b>2. Start über Kommandozeile .....</b>	<b>7</b>
<b>3. Interface .....</b>	<b>9</b>
3.1. Aufrufkonventionen .....	9
3.2. Syntax der Dokumentation .....	10
<b>4. Reportfunktionen.....</b>	<b>11</b>
Rep_ActivateReportWindow .....	11
Rep_AlignDialog .....	11
Rep_AppendGroup .....	12
Rep_ArrangeDialog .....	12
Rep_ArrangeSelectedObjects.....	13
Rep_ChangePrimaryObjectType .....	14
Rep_CheckTranslateAccel .....	15
Rep_Close.....	15
Rep_CopySelectedObjects .....	15
Rep_CountObjects.....	16
Rep_CreateHtmlOutput .....	16
Rep_CreateNewObject.....	17
Rep_CreateNewObjectExtended.....	17
Rep_CreateNewZone .....	18
Rep_CreateNewZoneExtended .....	19
Rep_DeclareAvailableFields .....	19
Rep_DeclarePrinterOutput.....	20
Rep_DeleteAllZones .....	20
Rep_DeleteSelectedObjects.....	20
Rep_DeleteZone.....	21
Rep_DisplayUsedDataFieldNames .....	21
Rep_EditColors .....	21
Rep_EditCopyList.....	22
Rep_EditDataSource.....	22
Rep_EditDisplayOptions .....	23
Rep_EditDataSource.....	23
Rep_EditFonts .....	24
Rep_EditFrameColors.....	24
Rep_EditGrid .....	25
Rep_EditGroups .....	25
Rep_EditHtmlOutput.....	26
Rep_EditHyperFrame .....	26
Rep_EditLanguage.....	26
Rep_EditPageBackground .....	27
Rep_EditPageFormat.....	27

---

Rep_EditPrimaryObjectPosAndSize .....	27
Rep_EditPrimarySelectedObject .....	28
Rep_EditPrinterProperties .....	28
Rep_EditProperties .....	28
Rep_EditRaster .....	29
Rep_EditRecordControl .....	29
Rep_EditReportMacros .....	30
Rep_EditUnitSystem .....	30
Rep_EditZones .....	30
Rep_FindNextObject .....	31
Rep_GetChangeFlag .....	32
Rep_GetCurrentZoneHandle .....	33
Rep_GetCurrentZoneTypeName .....	33
Rep_GetCustomPointer .....	33
Rep_GetCustomString .....	34
Rep_GetDataFieldNamesInZone .....	35
Rep_GetDataSource .....	35
Rep_GetEditOrderMode .....	36
Rep_GetHelpFileName .....	37
Rep_GetInstance .....	37
Rep_GetLanguage .....	37
Rep_GetPageBgrColor .....	38
Rep_GetPageBgrMode .....	38
Rep_GetPrimarySelectedObject .....	39
Rep_GetPrinterNameAndPort .....	39
Rep_GetProperty .....	40
Rep_GetReportMacro .....	40
Rep_GetReportMacroSize .....	41
Rep_GetTitleFileName .....	42
Rep_GetUsedDataFieldNames .....	42
Rep_GetVersion .....	43
Rep_GetZoneHandle .....	43
Rep_GetZoneNameList .....	44
Rep_IsReportClipboardAvailable .....	44
Rep_IsReportWindow .....	44
Rep_MakeUntitled .....	45
Rep_MuToString .....	45
Rep_Open .....	46
Rep_Open_Expanded .....	46
Rep_PasteObjectsToZone .....	47
Rep_Preview_File .....	48
Rep_Preview_Report .....	49
Rep_Preview_Report_NonModal .....	49
Rep_Print_File .....	51
Rep_Print_Report .....	52
Rep_ReArrangeDisplay .....	53
Rep_Save .....	53
Rep_SaveAs .....	54
Rep_Select .....	54
Rep_SelectedObjectsToBackground .....	55

Rep_SelectedObjectsToForeground.....	56
Rep_SetChangeFlag.....	56
Rep_SetCopyList.....	57
Rep_SetCustomPointer.....	57
Rep_SetCustomString.....	57
Rep_SetDataSource.....	58
Rep_SetEditOrderMode.....	59
Rep_SetErrorHandlerCallback.....	59
Rep_SetFieldnamesCallback.....	60
Rep_SetLanguage.....	61
Rep_SetObjectsAlignment.....	61
Rep_SetObjectsFormat.....	62
Rep_SetPageBgrBitmap.....	63
Rep_SetPageBgrColor.....	63
Rep_SetPageBgrMode.....	64
Rep_SetPrinterNameAndPort.....	64
Rep_SetProperty.....	65
Rep_SetReportMacro.....	66
Rep_SpacingDialog.....	67
Rep_SetUserCallback.....	68
Rep_Wizzard.....	69
Rep_Zoom.....	70
<b>5. Zonenfunktionen.....</b>	<b>71</b>
RepZone_DeleteObject.....	71
RepZone_GetIndicatorField.....	71
RepZone_GetName.....	72
RepZone_ResetOccurenceBits.....	72
RepZone_SetIndicatorField.....	72
RepZone_SetName.....	73
RepZone_SetOccurenceBits.....	74
<b>6. Objektfunktionen.....</b>	<b>75</b>
RepObj_GetFontDisplayString.....	75
RepObj_GetFontProperty.....	76
RepObj_GetLongProperty.....	76
RepObj_GetOutline.....	76
RepObj_GetPositionDisplayString.....	77
RepObj_GetSizeDisplayString.....	78
RepObj_GetStringProperty.....	78
RepObj_GetTypeDisplayString.....	79
RepObj_SetFontProperty.....	80
RepObj_SetLongProperty.....	80
RepObj_SetOutline.....	81
RepObj_SetStringProperty.....	81
<b>7. Formatdeskriptoren.....</b>	<b>82</b>
7.1. Datumsformat.....	82
7.2. Uhrzeitformat.....	84

---

7.3.	Ganzzahlformat .....	85
7.4.	Gleitkommazahlformat .....	86
<b>8.</b>	<b>Datenquellen für den Report .....</b>	<b>88</b>
8.1.	ANSI-Dateien .....	88
8.2.	ASCII-Dateien .....	88
8.3.	Tex-Ass-Selektionsdateien .....	88
8.4.	ODBC-Datenbanken .....	88
8.5.	DLL-Callback-Funktion (datensatzweise) .....	88
8.6.	DLL-Callback-Funktion (feldweise) .....	89
8.7.	Simulation von Daten .....	91
8.8.	Auswählen der Datenquelle .....	91
<b>9.</b>	<b>Betrieb des Reports als Druck-Server .....</b>	<b>92</b>
<b>10.</b>	<b>Compilieren und Linken .....</b>	<b>93</b>
10.1.	Header-Dateien .....	93
10.2.	Linker .....	93
<b>Anhang REPPROP</b>	<b>Report Properties .....</b>	<b>94</b>
<b>Anhang LOBJPROP</b>	<b>Long Properties von Objekten .....</b>	<b>102</b>
<b>Anhang SOBJPROP</b>	<b>String Properties von Objekten .....</b>	<b>109</b>
<b>Anhang FOBJPROP</b>	<b>Font Properties von Objekten .....</b>	<b>111</b>
<b>Anhang OBJ</b>	<b>Objekttypen .....</b>	<b>112</b>
<b>Anhang STAT</b>	<b>Statusbits von Objekten .....</b>	<b>113</b>
<b>Anhang NOTIF</b>	<b>Notification Messages .....</b>	<b>115</b>
<b>Anhang OBSOLETE</b>	<b>Veraltete Funktionen .....</b>	<b>119</b>
<b>Anhang EXAMPLE</b>	<b>Quellcode-Beispiel einer Einbindung .....</b>	<b>121</b>

## 1. Einleitung

Die folgende Dokumentation ist für Programmierer gedacht, die den Listengenerator **WaimeaReport** in eine bestehende Applikationsumgebung integrieren wollen.

Die Software besteht grundsätzlich aus zwei Komponenten - einem Editor zum Erstellen und Bearbeiten von Report-Layouts - und einem Print-Modul zur Ausgabe einer Datenliste in Form eines vorgegebenen Report-Layouts. Das Print-Modul bietet darüber hinaus eine Druckseitenvorschau.

Der Listengenerator **WaimeaReport** steht einerseits als *ausführbare Applikation*, andererseits als *Development-Kit* zur Einbindung in dritte Applikationen zur Verfügung.

Alle wesentlichen Leistungen und Funktionen der Software sind in der zentralen dynamischen Bibliothek **repdll.dll** konzentriert. Alle für die Einbindung notwendigen Funktionen sind in dieser Dokumentation beschrieben und können direkt aufgerufen werden.

In der Regel wird nur der Print-Server und die Druckseitenvorschau in die Zielumgebung integriert. Der Layout-Editor wird meist als eigenständige Applikation an den Kunden weitergegeben. Nichtsdestoweniger kann natürlich auch der Editor voll in die Rahmenapplikation eingebunden werden, wobei auch hier alle Bedienfunktionen von der Rahmenapplikation durch Aufruf gezielter DLL-Funktionen eingespeist werden können.

Die Rahmenapplikation stellt dem Report ein beliebiges Fenster zur Verfügung - das sogenannte Host-Fenster. Das Report-Modul öffnet in der *Client Area* des Host-Fensters ein rahmenloses, unsichtbares Verwaltungsfenster, in dem das gesamte Geschehen - sei es der Editor oder der Print-Server - abgewickelt wird. Dieses Fenster wird beim Initialisieren (**Rep\_Open\_Expanded**) angelegt. Sein Handle wird an die Rahmenapplikation zurückgeliefert und bei allen weiteren Aufrufen, die diesem Report gelten, als Parameter mitgegeben.

Mittels dieser Dokumentation wird es einem erfahrenen Entwickler innerhalb kurzer Zeit gelingen, den **WaimeaReport** voll in eine Rahmenapplikation zu integrieren.

Um den Einstieg zu erleichtern, haben wir den C++-Quellcode der Rahmenapplikation, welche als ausführbare Applikation ausgeliefert wird, ausführlich dokumentiert. Er liegt der Entwicklungsversion bei.

**Das Waimea-Team wünscht viel Erfolg bei der Integration.**

## 2. Start über Kommandozeile

**WaimeaReport** bietet zur Verwendung als Anwendung im Stapelbetrieb (Batch) die Möglichkeit, als Kommandozeile gestartet werden. Für diesen Fall können das zu verwendende Reportformular sowie eine Reihe von Schaltern als Kommandozeilenparameter an den Report übergeben werden.

Der Programmstart via Kommandozeile beginnt wie üblich mit der Angabe des Programmnamens. Dieser ist *report32.exe*. Der Programmname darf auch mit Pfad angegeben werden.

Nach dem Programmnamen können beliebig viele Kommandozeilenparameter folgen. Diese werden durch ein Leerzeichen voneinander getrennt. Der erste Parameter, der nicht mit einem Minuszeichen oder einem Schrägstrich / beginnen, wird als Dateiname einer Reportvorlage interpretiert und automatisch geöffnet.

Darüber hinaus können folgende Schalter gesetzt werden, die mit einem Minuszeichen oder einem / beginnen müssen:

`/minimize` Der Report wird automatisch minimiert gestartet. Das Programmsymbol erscheint in der Statuszeile, es wird aber kein Fenster eröffnet.

`/exit` Der Report wird sofort beendet.

`/print:name` Der aktive geöffnete Report wird auf dem Drucker mit dem angegebenen Namen gedruckt. Der Name des Druckers folgt dem Schalter `/print` durch einen Doppelpunkt getrennt. Wenn der Name des Druckers weggelassen werden, erfolgt die Ausgabe auf dem Standarddrucker. In diesem Fall muß auch der Doppelpunkt fehlen.

`/print` Der aktive geöffnete Report wird auf dem Standarddrucker ausgegeben.

`/outputfile=datei` Die Druckausgabe erfolgt in die angegebene Datei anstelle auf den Druckkanal

`/docname=name` Der Druckauftrag geht unter dem angegebenen Namen an den Windows-Spooler. Wenn dieser Schalter nicht angegeben wird, wird ein Default-Name verwendet.

Alle angegebenen Schalter werden von links nach rechts abgearbeitet. Wenn ein Schalter oder ein Dateiname Leerzeichen enthält, so kann dieser in Anführungszeichen "" eingeschlossen werden. Der Delimiter, also das Minuszeichen bzw. der Schrägstrich, muß im Falle eines Schalters mit eingeschlossen werden.

Der folgende Schalter druckt beispielsweise den aktiven Report auf einem explizit angegebenen Drucker: `"/print:HP Laser Jet 6P"`.

## Beispiel

Das folgende Beispiel startet **WaimeaReport** unsichtbar (minimiert), öffnet das Reportformular *test.rep* aus dem Verzeichnis *c:\temp* und druckt den Report auf dem Drucker *HP Laser Jet 6P*. Nach dem Druck wird der Report automatisch wieder beendet:

```
Report32 c:\temp\test.rep /minimize "/print:HP Laser Jet 6P" /exit
```

Das Beispiel läßt sich erweitern. Die folgende Kommandozeile druckt den Report zusätzlich auf dem Drucker mit dem Namen *Test*:

```
Report32 c:\temp\test.rep /minimize "/print:HP Laser Jet 6P" /print:Test /exit
```

Der Start von **WaimeaReport** auf Kommandozeilenebene ist insbesondere für die Einbindung in Systemumgebungen gedacht, in denen der direkte Zugriff auf DLL-Schnittstellen nicht möglich ist.

Für weitergehende Kontrolle über den Report sollte auf eine Einbindung über Interface-Routinen zurückgegriffen werden. Diese wird in den folgenden Kapiteln erläutert.

### 3. Interface

Das Programmier-Interface von **WaimeaReport** besteht aus einer Vielzahl von Funktionen, die in einer DLL zur Verfügung gestellt werden. Grundsätzlich stehen alle Funktionen als 16-Bit-Variante und als 32-Bit-Variante zur Verfügung.

Damit der Report in möglichst viele Programm-Umgebungen problemlos eingebunden werden kann, werden alle Schnittstellen von **WaimeaReport** für unterschiedliche Aufrufkonventionen bereitgestellt. Je nach dem, welche Aufrufkonventionen das Entwicklungssystem, in das der Report eingebunden werden soll, unterstützt, kann auf die jeweilige Aufrufkonvention zurückgegriffen werden.

#### 3.1. Aufrufkonventionen

Zur Zeit werden – je nach Entwicklungsumgebung – zwei unterschiedliche Aufrufkonvention für DLL-Funktionen verwendet. Es sind dies die Konventionen **stdcall** und **cdecl**. Funktionen, die mit der Aufrufkonvention **stdcall** kompiliert sind, entfernen selbst ihre Aufrufparameter vom Stack, während dies bei Funktionen, die mit **cdecl** kompiliert sind, der Programmcode des aufrufenden Programmzweigs erledigt. Dadurch sind beide Konventionen zueinander inkompatibel.

Leider befinden sich auf dem Markt sowohl Entwicklungssysteme, die die Aufrufkonvention **cdecl** verwenden (z.B. *Smalltalk*), als auch solche, die **stdcall** benötigen (z.B. *Visual Basic*). Um möglichst viele Programmierumgebungen unterstützen zu können, werden alle Schnittstellen des Reportgenerators in beiden Konventionen bereitgestellt.

Die Namen aller Funktionen, die mit der Aufrufkonvention **cdecl** kompiliert sind, beginnen mit dem Prefix **Rep\_**xxxxx oder **RepObj\_**xxxxx.

Die Namen aller Funktionen, die mit der Aufrufkonvention **stdcall** kompiliert sind, beginnen mit dem Prefix **SRep\_**xxxxx oder **SRepObj\_**xxxxx. Es wurde dem Namen also ein großes **S** vorangestellt.

Der Übersichtlichkeit halber werden in dieser Dokumentation nur die **cdecl**-Schnittstellen beschrieben. Die Beschreibung kann jedoch eins zu eins auch für die **stdcall**-Schnittstellen verwendet werden. Es muß lediglich dem Namen der Funktion ein großes **S** vorangestellt werden.

#### Visual Basic

Wenn **WaimeaReport** in eine *Visual Basic* Umgebung eingebunden wird, muß in jedem Fall die Konvention **stdcall** verwendet werden. Es müssen also diejenigen Routinen verwendet werden, deren Namen mit einem großen **S** beginnt.

## C oder C++

Wenn **WaimeaReport** in eine C oder C++ Umgebung eingebettet wird, können beide Aufrufkonventionen verwendet werden. Der Compiler unterscheidet die jeweilige Konvention durch die *Prototypes*, welche in der Header-Datei definiert sind.

## Andere Umgebungen

Wenn **WaimeaReport** in eine andere Umgebung eingebettet wird, entnehmen Sie bitte der Dokumentation der jeweiligen Entwicklungsumgebung, welche Aufrufkonvention erwartet wird.

## 3.2. Syntax der Dokumentation

Es stehen zwei Gruppen von Funktionen zur Verfügung – Reportfunktionen und Objektfunktionen. Reportfunktionen beziehen sich immer auf einen bestimmten Report – Objektfunktionen beziehen sich immer auf ein bestimmtes Objekt eines Reports.

Die folgenden beiden Kapitel beschreiben alle Report- und Objektfunktionen. Die einzelnen Funktionen sind in alphabetischer Reihenfolge angeordnet. Die Beschreibung jeder Funktion besteht aus einem Kopf, welcher den Namen und die Parameter der Funktion sowie die Typen der Parameter aufzählt und einem Rumpf, welcher die Funktion beschreibt.

Wenn die Funktion einen Funktionswert zurückliefert, wird dieser nach dem Rumpf beschrieben. Danach folgen Verweise auf verwandte Funktionen, falls solche existieren.

## 4. Reportfunktionen

### Rep\_ActivateReportWindow

HWND      *hwndReport*

Diese Funktion aktiviert ein Reportfenster. Dabei erhält das Reportfenster den Fokus. Die Funktion wird vornehmlich aufgerufen, wenn im Rahmen einer MDI-Applikation ein MDI-Fenster, in dem ein Report abgewickelt wird, aktiviert wird.

*Die Funktion liefert keinen Funktionswert zurück.*

### Rep\_AlignDialog

HWND      *hwndReport*  
int \*      *Modus*

Diese Funktion öffnet einen Dialog zur Auswahl einer von mehreren möglichen Ausrichtungen. Die augenblickliche Ausrichtung wird im Dialog voreingestellt. Die bei Bestätigung des Dialogs ausgewählte Ausrichtung wird zurückgeliefert.

Auf Parameter *Modus* wird einer der folgenden Werte übergeben:

ARR_ALIGNLEFT	am linken Rand ausrichten
ARR_ALIGNRIGHT	am rechten Rand ausrichten
ARR_ALIGNTOP	am oberen Rand ausrichten
ARR_ALIGNBOTTOM	am unteren Rand ausrichten
ARR_ALIGNVERTCENTER	mittig zur Senkrechten ausrichten

Wenn der Dialog bestätigt wurde, wird auf dem Parameter *Modus* der vom Benutzer ausgewählte Status zurückgeliefert. **Modus ist eine Zeigervariable!**

Diese Funktion selbst führt keine Änderung durch. Erst durch Aufruf der Funktion *Rep\_ArrangeSelectedObjects* und Übergabe der von *Rep\_AlignDialog* erhaltenen Werte werden die selektierten Objekte wie angegeben arrangiert.

**Funktionswert vom Typ int**

- 0 Der Dialog wurde bestätigt. **Modus** liefert den ausgewählten Wert.
- 1 Der Dialog wurde abgebrochen.

**Verwandte Funktionen**

Rep\_ArrangeDialog  
 Rep\_SpacingDialog  
 Rep\_ArrangeSelectedObjects

**Rep\_AppendGroup**

HWND	<i>hwndRep</i>
char *	<i>FieldName</i>
int	<i>Column</i>
int	<i>Feed</i>

Diese Funktion fügt einem geöffneten Report eine Gruppe hinzu. **FieldName** gibt den Namen des Datenfeldes an, auf welchen sich der Gruppenwechsel bezieht. **Column** gibt die Anzahl Zeichen an, die zum Auslösen eines Gruppenwechsels verglichen werden soll. **Feed** gibt an, ob bei einem Gruppenwechsel ein Spalten- oder Seitenvorschub durchgeführt werden soll.

**Feed** kann folgende Werte annehmen:

GROUPFEED_NONE	es wird kein Vorschub ausgelöst
GROUPFEED_COLUMN	es wird ein Spaltenwechsel ausgelöst
GROUPFEED_PAGE	es wird ein Seitenvorschub ausgelöst

**Funktionswert vom Typ int**

- IDOK die neue Gruppe wurde fehlerfrei hinzugefügt.
- IDCANCEL es ist ein Fehler aufgetreten

**Verwandte Funktionen**

Rep\_EditGroups

**Rep\_ArrangeDialog**

HWND	<i>hwndReport</i>
int *	<i>Modus</i>
long *	<i>Abstand</i>

Diese Funktion öffnet einen Dialog zur Konfiguration der Anordnung, Ausrichtung und Größe von Objekten.

Auf Parameter **Modus** wird einer der folgenden Werte übergeben:

ARR_ALIGNLEFT	am linken Rand ausrichten
ARR_ALIGNRIGHT	am rechten Rand ausrichten
ARR_ALIGNTOP	am oberen Rand ausrichten
ARR_ALIGNBOTTOM	am unteren Rand ausrichten
ARR_ALIGNVERTCENTER	mittig zur Senkrechten ausrichten
ARR_ALIGHORICENTER	mittig zur Waagerechten ausrichten
ARR_ADJUSTWIDTH	Breite angleichen
ARR_ADJUSTHEIGHT	Höhe angleichen
ARR_ADJUSTHORIDIST	nebeneinander anordnen
ARR_ADJUSTVERTDIST	untereinander anordnen

Wenn der Dialog bestätigt wurde, wird auf dem Parameter **Modus** der vom Benutzer ausgewählte Status und auf **Abstand** die Distanz zur Anordnung zurückgeliefert. **Modus und Abstand sind Zeigervariablen!** Diese Funktion selbst führt keine Änderung durch. Erst durch Aufruf der Funktion **Rep\_ArrangeSelectedObjects** und Übergabe der von **Rep\_ArrangeDialog** erhaltenen Werte werden die selektierten Objekte wie angegeben arrangiert.

#### **Funktionswert vom Typ int**

0	Der Dialog wurde bestätigt.
-1	Der Dialog wurde abgebrochen.

#### **Verwandte Funktionen**

Rep\_AlignDialog  
Rep\_SpacingDialog  
Rep\_ArrangeSelectedObjects

### **Rep\_ArrangeSelectedObjects**

<b>HWND</b>	<b>hwndReport</b>
<b>int</b>	<b>Modus</b>
<b>long</b>	<b>Abstand</b>

Diese Funktion verändert die Anordnung, Ausrichtung und Größe von selektierten Objekten. Der Parameter **Modus** bestimmt in Form einer Liste von Angaben, welche der genannten Operationen ausgeführt werden sollen.

Mittels der zugehörigen Dialog-Funktionen **Rep\_ArrangeDialog**, **Rep\_AlignDialog** und **Rep\_SpacingDialog** kann der Parameter **Modus** zuvor durch den entsprechenden Dialog automatisch ermittelt werden.

**Modus** ist eine durch logisches *Oder* kombinierte Liste eines oderer mehrerer der folgenden Werte:

ARR_ALIGNLEFT	am linken Rand ausrichten
ARR_ALIGNRIGHT	am rechten Rand ausrichten
ARR_ALIGNTOP	am oberen Rand ausrichten
ARR_ALIGNBOTTOM	am unteren Rand ausrichten
ARR_ALIGNVERTCENTER	mittig zur Senkrechten ausrichten
ARR_ALIGNHORICENTER	mittig zur Waagerechten ausrichten
ARR_ADJUSTWIDTH	Breite angleichen
ARR_ADJUSTHEIGHT	Höhe angleichen
ARR_ADJUSTHORIDIST	nebeneinander anordnen
ARR_ADJUSTVERTDIST	untereinander anordnen

Sich widersprechende Werte dürfen nicht kombiniert werden, wie z.B. die gleichzeitige Ausrichtung am linken und rechten Rand.

Die Funktion liefert keinen Funktionswert zurück.

#### **Verwandte Funktionen**

Rep\_AlignDialog  
Rep\_SpacingDialog  
Rep\_ArrangeDialog

## **Rep\_ChangePrimaryObjectType**

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>long</b>	<b><i>ObjektTyp</i></b>

Diese Funktion wandelt den Typ des primär markierten Objekts des Reports um. Das bestehende Objekt wird entfernt und durch ein neues Objekt eines anderen Typs ersetzt. Dabei werden nur der Umriß und die Randeigenschaften des Objekts übernommen. Alle anderen Eigenschaften werden bei dem neuen Objekt der Voreinstellung angepaßt.

Der Typ des Objekts wird als Parameter ***ObjektTyp*** übergeben. Wenn hier der Wert 0 angegeben wird, öffnet sich ein Dialog, in dem der Benutzer den Typ des neuen Objekts auswählen kann. Die Liste aller möglichen Objekttypen ist im Anhang OBJ beschrieben.

Die Funktion liefert keinen Funktionswert zurück.

#### **Verwandte Funktionen**

Rep\_CreateNewObject

## Rep\_CheckTranslateAccel

**MSG \***      **Message**

Diese Funktion kann in der Message-Loop der Rahmen-Applikation aufgerufen werden. Sie prüft, ob die angegebene Windows-Message in der Anwendung mittels der Funktion **TranslateAccelerator** modifiziert werden sollte oder nicht.

Insbesondere die Messages **WM\_KEYUP**, **WM\_KEYDOWN** and **WM\_CHAR** für Edit-Control-Fenster sollten nicht modifiziert werden, damit diejenigen Tasten, die innerhalb von Edit-Controls eine spezifische Bedeutung haben, einwandfrei funktionieren können.

**Funktionswert vom Typ int**

- 1 Die angegebene Message kann ohne weiteres mittels der Funktion **TranslateAccelerator** modifiziert werden.
- 0 Die angegebene Message sollte nicht mittels der Funktion **TranslateAccelerator** modifiziert werden.

## Rep\_Close

**HWND**      **hwndReport**

Diese Funktion schließt ein zuvor mit *Rep\_Open* oder *Rep\_Open\_Expanded* geöffnetes Reportfenster. Alle von diesem Report belegten Systemresource werden wieder freigegeben. Das Parent-Window kann für einen anderen Report weiterverwendet werden. Wird das Parent-Window geschlossen, so ist es nicht nötig *Rep\_Close* aufzurufen, da impliziet ein *Rep\_Close* durchgeführt wird.

Die Funktion liefert keinen Funktionswert zurück.

## Rep\_CopySelectedObjects

**HWND**      **hwndReport**  
**int**          **Modus**

Diese Funktion kopiert alle markierten Objekte in die Zwischenablage. Der Parameter **Modus** spezifiziert, welche Objekte in die Zwischenablage kopiert werden:

**RM\_CURRENTZONE**      es werden alle markierten Objekte der aktuellen Zone in die Zwischenablage kopiert

Die Funktion liefert keinen Funktionswert zurück.

**Verwandte Funktionen**

Rep\_PasteObjectsToZone  
Rep\_DeleteSelectedObjects

**Rep\_CountObjects**

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>int</b>	<b><i>Modus</i></b>
<b>long</b>	<b><i>ObjektTypen</i></b>

Diese Funktion zählt alle Objekte eines Reports oder einer Zone und liefert deren Anzahl als Funktionswert zurück. Der Parameter ***Modus*** spezifiziert, welche Objekte gezählt werden sollen. Es können eine oder mehrere der folgenden Angaben durch ein logisches *oder* miteinander kombiniert werden:

RM_TOTALREPORT	es werden alle Objekte gezählt
RM_CURRENTZONE	es werden alle Objekte der aktuellen Zone gezählt
RM_SELECTED	es werden nur markierte Objekte gezählt

Es dürfen keine Werte kombiniert werden, die sich widersprechen.

Der Parameter ***ObjektTypen*** spezifiziert, welche Art von Objekten gezählt werden soll. Grundsätzlich können alle im Anhang OBJ aufgelisteten Objekttypen verwendet werden. Es können ein oder mehrere Objekttypen durch ein logisches *oder* miteinander kombiniert werden.

Darüber hinaus können folgende Objektgruppen angesprochen werden:

OBJTYPE_ALL	alle Objekte
OBJTYPE_VARIABLE	alle Objekte mit Datenfeldbezug
OBJTYPE_ALL_BUT_FIELD	all Objekte außer Datenfeldern

**Funktionswert vom Typ int**

Die Funktion liefert die Anzahl der Objekte zurück, die aufgrund der angegebenen Parameter ermittelt wurde.

**Rep\_CreateHtmlOutput**

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>char *</b>	<b><i>FileName</i></b>
<b>long</b>	<b><i>Flags</i></b>

Diese Funktion erzeugt einen kompletten Report auf Grundlage des durch ***hwndReport*** angegebenen Reportformulars. Das Ergebnis ist eine oder mehrere HTML-Dateien. ***FileName*** gibt den Namen bzw den Namensprefix für die HTML-Datei(en) an. Wird NULL oder ein Leerstring übergeben, so wird ein Dateiname angefragt. ***Flags*** ist für zukünftige Zwecke gedacht und sollte mit 0 vorbesetzt werden. Die Ausgabe erfolgt gemäß den im Report festgelegten HTML-Merkmalen (siehe ***Rep\_EditHtmlOutput***).

**Funktionswert vom Typ int**

- 0 Der HTML-Output wurde erfolgreich ausgeführt.
- 1 Es ist eine Fehler aufgetreten oder der dateinamensdialog wurde abgebrochen.

**Verwandte Funktionen**

Rep\_EditHtmlOutput

**Rep\_CreateNewObject**

HWND	<i>hwndReport</i>
long	<i>ObjektTyp</i>

Diese Funktion legt ein neues Objekt in der aktuellen Zone des Reports an. Der Typ des Objekts wird als Parameter **ObjektTyp** übergeben. Wenn hier der Wert 0 angegeben wird, öffnet sich ein Dialog, in dem der Benutzer den Typ des neuen Objekts auswählen kann. Die Liste aller verfügbaren Objekttypen ist im Anhang OBJ beschrieben.

Die Funktion liefert keinen Funktionswert zurück.

**Verwandte Funktionen**

Rep\_CreateNewObjectExtended

**Rep\_CreateNewObjectExtended**

HWND	<i>hwndReport</i>
long	<i>ObjektTyp</i>
long	<i>Flags</i>

Diese Funktion legt ein neues Objekt in der aktuellen Zone des Reports an. Der Typ des Objekts wird als Parameter **ObjektTyp** übergeben. Wenn hier der Wert 0 angegeben wird, öffnet sich ein Dialog, in dem der Benutzer den Typ des neuen Objekts auswählen kann. Die Liste aller verfügbaren Objekttypen ist im Anhang OBJ beschrieben.

Wenn keine Flags angegeben werden, eröffnet die Funktion bei bestimmten Objekttypen nach dem Anlegen des Objekts automatisch einen Dialog, der die Modifikation des neu angelegten Objekts ermöglicht.

Wenn dieser Dialog nicht erwünscht ist, kann folgendes Flag angegeben werden:

REP\_CO\_NODIALOG

Die Funktion liefert keinen Funktionswert zurück.

#### **Verwandte Funktionen**

Rep\_CreateNewObject

### **Rep\_CreateNewZone**

<b>HWND</b>	<b>hwndReport</b>
<b>Int</b>	<b>Type</b>
<b>const char *</b>	<b>Name</b>

Diese Funktion legt eine neue Zone des angegebenen Typs an und hängt diese an den Report mit dem Handle **hwndReport** an.

Der Zonentyp kann folgende Werte annehmen:

ZONE\_REPHEAD  
ZONE\_PAGEHEAD  
ZONE\_PAGEFOOT  
ZONE\_LASTPAGEFOOT  
ZONE\_DATA  
ZONE\_FINALELEM  
ZONE\_GROUPHEAD  
ZONE\_GROUPDATA  
ZONE\_GROUPFOOT

Wenn der Zonentyp keine Gruppenzone ist und eine entsprechende Zone bereits existiert, ist diese Funktion ohne Wirkung und es wird der Wert 0 zurückgeliefert.

#### **Die Funktion liefert einen Funktionswert vom Typ HREPZONE**

n Handle der neu angelegten Zone  
0 Fehler aufgetreten.

#### **Verwandte Funktionen**

Rep\_CreateNewZoneExtended

## Rep\_CreateNewZoneExtended

HWND	<i>hwndReport</i>
int	<i>Type</i>
char *	<i>Name</i>
HREPZONE	<i>InsertBefore</i>
long	<i>Flags</i>

Diese Funktion legt analog zur Routine *Rep\_CreateNewZone* eine neue Zone des angegebenen Typs an. Anders als bei *Rep\_CreateNewZone* kann hier ein Handle einer anderen, existierenden Zone (*InsertBefore*) mitgegeben werden. Ist der Handle nicht NULL, so wird die neu erzeugte Zone direkt vor der angegebenen Zone in die Zonenkette eingefügt. Der Zonenhandle kann z.B. durch einen früheren Aufruf von *Rep\_CreateZone* oder *Rep\_GetZoneHandle* ermittelt worden sein. *Flags* ist für zukünftige Zwecke gedacht und muß mit Null vorbelegt werden.

*Die Funktion liefert einen Funktionswert vom Typ HREPZONE*

n        Handle der neu angelegten Zone  
0        Es ist ein Fehler aufgetreten.

### *Verwandte Funktionen*

Rep\_CreateNewZone  
Rep\_GetZoneHandle

## Rep\_DeclareAvailableFields

HWND	<i>hwndReport</i>
const char *	<i>FieldList</i>

Diese Funktion deklariert eine Liste von Feldnamen, die vom Report an allen Stellen zur Auswahl angezeigt wird, an denen diese Leistung aufgerufen wird. Die einzelnen Feldnamen in der Liste müssen durch Tabulatoren voneinander getrennt sein.

Durch Angabe des Wertes NULL als *FieldList* wird eine bereits angemeldete Liste neutralisiert.

Die Funktion liefert keinen Funktionswert zurück.

### *Verwandte Funktionen*

Rep\_SetFieldnamesCallback

## Rep\_DeclarePrinterOutput

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>const char *</b>	<b><i>FileName</i></b>
<b>long</b>	<b><i>Flags</i></b>

Diese Funktion deklariert einen Dateinamen, in denen künftig der alle Druckaufträge dieses Reports umgeleitet werden.

Der Parameter ***Flags*** sollte stets den Wert 0 haben.

Die Funktion liefert keinen Funktionswert zurück.

## Rep\_DeleteAllZones

<b>HWND</b>	<b><i>hwndReport</i></b>
-------------	--------------------------

Diese Funktion löscht alle Zonen eines Reports und alle darin vorhandenen Objekte.

Die Funktion liefert keinen Funktionswert zurück.

### ***Verwandte Funktionen***

Rep\_CreateNewZone  
Rep\_CreateNewZoneExtended

## Rep\_DeleteSelectedObjects

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>int</b>	<b><i>Modus</i></b>

Diese Funktion löscht alle markierten Objekte einer Zone. Der Parameter ***Modus*** spezifiziert, welche Objekte gelöscht werden:

RM\_CURRENTZONE                    es werden alle markierten Objekte der aktuellen Zone gelöscht.

Die Funktion liefert keinen Funktionswert zurück.

### ***Verwandte Funktionen***

Rep\_PasteObjectsToZone  
Rep\_CopySelectedObjects

## Rep\_DeleteZone

<b>HWND</b>	<b>hwndReport</b>
<b>HREPZONE</b>	<b>Zone</b>
<b>long</b>	<b>Flags</b>

Diese Routine löscht die angegebenen Zone aus dem Reportformular. **Zone** muß der Handle einer gültigen Zone des durch **hwndReport** bestimmten Reports sein. Er kann durch Aufruf von Funktionen wie z.B. **Rep\_CreateNewZone** oder **Rep\_GetZoneHandle** ermittelt worden sein. **Flags** ist für zukünftige Zwecke gedacht und muß mit Null vorbelegt werden.

*Die Funktion liefert keinen Funktionswert zurück.*

### *Verwandte Funktionen*

Rep\_CreateNewZone  
Rep\_CreateNewZoneExtended  
Rep\_GetZoneHandle

## Rep\_DisplayUsedDataFieldNames

<b>HWND</b>	<b>hwndReport</b>
-------------	-------------------

Diese Funktion zeigt eine Liste aller in dem angegebenen Report verwendeten Datenfeldnamen an. Dabei ist es gleichgültig, in welcher Zone des Reports die Datenfeldnamen verwendet werden.

Es werden auch die Namen von Datenfeldern angezeigt, die in Gruppen oder in Zonendefinitionen verwendet werden. Das gilt auch dann, wenn diese im Report nicht gedruckt werden.

Die Funktion liefert keinen Funktionswert zurück.

### *Verwandte Funktionen*

Rep\_GetUsedDataFieldNames

## Rep\_EditColors

<b>HWND</b>	<b>hwndReport</b>
<b>int</b>	<b>Modus</b>

Diese Funktion öffnet einen Farben-Dialog zum Einstellen der Hintergrundfarbe aller markierten Objekte. Objekte, die keine Hintergrundfarbe haben (z.B. Linien), bleiben unverändert, auch wenn sie markiert sind.

Der Parameter **Modus** spezifiziert, welche Objekte beeinflusst werden sollen:

RM\_CURRENTZONE            es werden nur die markierten Objekte der aktuellen Zone beeinflusst

**Funktionswert vom Typ int**

- 1    Der Dialog wurde bestätigt.
- 0    Der Dialog wurde abgebrochen.

**Verwandte Funktionen**

Rep\_EditFrameColors  
Rep\_EditFonts  
Rep\_EditProperties

## Rep\_EditCopyList

**HWND**        *hwndRep*

Diese Funktion öffnet den Dialog zum Anlegen oder Verändern der Ausfertigungsliste eines Reportformulars.

**Funktionswert vom Typ int**

- IDOK Der Dialog wurde mit **OK** bestätigt.
- IDCANCEL        Der Dialog wurde abgebrochen.

**Verwandte Funktionen**

Rep\_SetCopyList

## Rep\_EditDataSource

**HWND**                    *hwndRep*

Diese Funktion eröffnet einen Dialog und zum Einstellen und Verändern der Datenquelle des angegebenen Reports.

Der Parameter **hwndRep** definiert den Report, dessen Datenquelle ausgelesen werden soll.

**Funktionswert**

IDOK            Der Datenquellen-Dialog wurde mit OK beendet.  
IDCANCEL       Der Datenquellen-Dialog wurde mit ABBRUCH abgebrochen

**Verwandte Funktionen**

Rep\_SetDataSource  
Rep\_GetDataSource

**Rep\_EditDisplayOptions**

**HWND**        *hwndReport*

Diese Funktion öffnet einen Dialog, der es dem Benutzer ermöglicht, die Darstellung des Report-Editors auf dem Bildschirm zu konfigurieren. Es kann z.B. die Farbe des Fensterhintergrundes und anderes mehr eingestellt werden. Der Dialog erscheint insbesondere, wenn im Menü *Anzeige* der Eintrag *Bildschirmdarstellung...* aufgerufen wird.

**Funktionswert vom Typ int**

1    Der Dialog wurde bestätigt.  
0    Der Dialog wurde abgebrochen.

**Rep\_EditDataSource**

**HWND**        *hwndReport*

Diese Funktion öffnet einen Dialog, der es dem Benutzer ermöglicht, eine Datenquelle für den angegebenen Report auszuwählen.

Als Datenquelle werden verschiedene Datei- und Datenbankzugriffe angeboten. Auch eine Simulation von Daten kann ausgewählt werden.

Die ausgewählte Datenquelle wird fest im Report eingetragen. Sie wird benutzt, wenn der Report gedruckt wird oder wenn eine Druckseitenvorschau auf dem Bildschirm erstellt wird. In der Druckseitenvorschau kann die Datenquelle nochmals verändert werden, bevor der eigentliche Druck erzeugt wird.

**Funktionswert vom Typ int**

1    Der Dialog wurde bestätigt.  
0    Der Dialog wurde abgebrochen.

**Verwandte Funktionen**

Rep\_GetDataSourceType  
Rep\_SetDataSourceType  
Rep\_GetDataSouceltem  
Rep\_SetDataSouceltem

**Rep\_EditFonts**

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>int</b>	<b><i>Modus</i></b>

Diese Funktion öffnet den Schriften-Dialog. Hier können die Schriften aller markierten Objekte neu definiert werden. Objekte, die nicht über eine Schrift verfügen (z.B. Rechtecke), bleiben unverändert, auch wenn sie markiert sind.

Der Parameter ***Modus*** spezifiziert, welche Objekte beeinflusst werden sollen:

RM\_CURRENTZONE            es werden nur die markierten Objekte der aktuellen Zone beeinflusst

**Funktionswert vom Typ int**

- 1    Der Dialog wurde mit **OK** bestätigt.
- 0    Der Dialog wurde abgebrochen.

**Verwandte Funktionen**

Rep\_EditFrameColors  
Rep\_EditColors  
Rep\_EditProperties

**Rep\_EditFrameColors**

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>int</b>	<b><i>Modus</i></b>

Diese Funktion öffnet einen Farben-Dialog zum Einstellen der Rahmenfarben aller markierten Objekte. Die Rahmenfarben sind nicht mit den Hintergrundfarben zu verwechseln.

Es werden auch die Rahmenfarben von Objekten eingestellt, für die momentan die Eigenschaft *kein Rand* eingestellt ist. Das Ergebnis ist also nicht sichtbar, tritt jedoch zutage, wenn der Rand auf *sichtbar* umgestellt wird.

Der Parameter **Modus** spezifiziert, welche Objekte beeinflusst werden sollen:

RM\_CURRENTZONE            es werden nur die markierten Objekte der aktuellen Zone beeinflusst

**Funktionswert vom Typ int**

- 1    Der Dialog wurde mit **OK** bestätigt.
- 0    Der Dialog wurde abgebrochen.

**Verwandte Funktionen**

Rep\_EditColors  
Rep\_EditFonts  
Rep\_EditProperties

## Rep\_EditGrid

**HWND      *hwndReport***

Diese Funktion öffnet einen Dialog zur Konfiguration der Anzeige eines Gitternetzes im Reportformular. Dies ist ein hilfreiches Werkzeug zur exakten und gleichmäßigen Positionierung von Objekten über Zonen-  
grenzen hinweg.  
Die Einstellungen bezüglich des Gitternetzes werden im Reportformular selbst gespeichert.

**Funktionswert vom Typ int**

- TRUE      Der Dialog wurde bestätigt.
- FALSE     Der Dialog wurde abgebrochen.

**Verwandte Funktionen**

Rep\_EditRaster

## Rep\_EditGroups

**HWND      *hwndReport***

Diese Funktion öffnet den Dialog zum Anlegen, Ändern und Löschen von Gruppendefinitionen. Eine gleich-  
namige Funktion ist im Menü des Reporteditors vorhanden.

**Funktionswert vom Typ int**

- TRUE      Der Dialog wurde bestätigt.
- FALSE     Der Dialog wurde abgebrochen.

## Rep\_EditHtmlOutput

**HWND**      *hwndReport*

Diese Funktion öffnet den Dialog zur Konfiguration des HTML-Exports. Titel, Autor und Suchbegriffe sowie das Ausgabeverzeichnis und Bildformate können hier angegeben werden.

### **Funktionswert vom Typ int**

TRUE        Der Dialog wurde bestätigt.  
FALSE       Der Dialog wurde abgebrochen.

### **Verwandte Funktionen**

Rep\_CreateHtmlOutput

## Rep\_EditHyperFrame

**HWND**      *hwndReport*

Diese Funktion öffnet den Dialog zur Konfiguration des zonenübergreifenden Hyperrahmens. Hier kann ein Zonenbereich bestimmt werden, um den ein gemeinsamer, einheitlicher Rahmen gezogen werden soll.

### **Funktionswert vom Typ int**

TRUE        Der Dialog wurde bestätigt.  
FALSE       Der Dialog wurde abgebrochen.

## Rep\_EditLanguage

**HWND**      *hwndReport*

Diese Funktion öffnet den Dialog zum Einstellen der aktuell zu verwendenden Landessprache. Formatangaben, z.B bei Gleitkommazahlen, oder dem Datum werden je eingestellte Landessprache separat gespeichert. So kann das vorliegende Reportformular online für eine internationale Verwendung umgeschaltet werden.

### **Funktionswert vom Typ int**

TRUE        Der Dialog wurde bestätigt.  
FALSE       Der Dialog wurde abgebrochen.

## Rep\_EditPageBackground

**HWND**      *hwndReport*

Diese Funktion öffnet einen Dialog, der es ermöglicht, die Gestaltung des Seitenhintergrundes zu konfigurieren. Es kann insbesondere die Farbe der Seiten und eine Hintergrundbitmap vorgegeben werden.

### **Funktionswert vom Typ int**

IDOK              Der Dialog wurde bestätigt.  
IDCANCEL         Der Dialog wurde abgebrochen.

### **Verwandte Funktionen**

Rep\_SetPageBgrBitmap  
Rep\_GetPageBgrMode  
Rep\_SetPageBgrMode  
Rep\_GetPageBgrColor  
Rep\_SetPageBgrColor

## Rep\_EditPageFormat

**HWND**      *hwndReport*

Diese Funktion öffnet den Dialog zur Konfiguration des Blattformats. Die Blattgröße, die Randabstände, die Anzahl Spalten sowie die Druckrichtung (*Portrait, Landscape*) werden hier eingestellt.

### **Funktionswert vom Typ int**

TRUE              Der Dialog wurde bestätigt.  
FALSE             Der Dialog wurde abgebrochen.

## Rep\_EditPrimaryObjectPosAndSize

**HWND**      *hwndReport*

Diese Funktion öffnet einen Dialog, der es ermöglicht, die Größe und Position des vorrangig markierten Objekts zu ändern. Die Position und Größe kann metrisch eingegeben werden.

Insbesondere können der linke und obere Rand eines Objekts nicht nur absolut, sondern auch relativ zur Position anderer Objekte angegeben werden.

### **Funktionswert vom Typ int**

TRUE              Der Dialog wurde bestätigt.  
FALSE             Der Dialog wurde abgebrochen.

## Rep\_EditPrimarySelectedObject

**HWND**      *hwndReport*

Diese Funktion öffnet einen Dialog, welcher die Änderung der Eigenschaften des augenblicklich primär markierten Objekts ermöglicht. Je nach Typ des Objekts, welches primär markiert ist, enthält der Dialog Eingabemöglichkeiten, die auf die spezifischen Eigenarten des betreffenden Objekts zugeschnitten sind.

Wenn kein Objekt primär markiert ist, ist der Aufruf dieser Funktion ohne Wirkung.

**Funktionswert vom Typ** int

TRUE      Der Dialog wurde mit **OK** bestätigt.  
FALSE     Der Dialog wurde abgebrochen.

## Rep\_EditPrinterProperties

**HWND**      *hwndReport*

Diese Funktion öffnet einen Dialog, der es dem Benutzer ermöglicht, die Druckereinstellungen vorzunehmen. Insbesondere kann der Benutzer wählen, welche Seiten auf welchem Druckerschacht ausgegeben werden.

**Funktionswert vom Typ** int

TRUE      Der Dialog wurde bestätigt.  
FALSE     Der Dialog wurde abgebrochen.

## Rep\_EditProperties

**HWND**      *hwndReport*  
**int**         *Modus*

Diese Funktion öffnet einen Dialog, der es ermöglicht, gemeinsame Eigenschaften aller markierten Objekte gleichzeitig einzustellen – z.B. Schriften, Randfarben, Farben, usw..

Der Parameter **Modus** spezifiziert, welche Objekte beeinflusst werden sollen:

RM\_CURRENTZONE      es werden nur die markierten Objekte der aktuellen Zone beeinflusst

**Funktionswert vom Typ** int

1      Der Dialog wurde mit bestätigt.  
0      Der Dialog wurde abgebrochen.

**Verwandte Funktionen**

Rep\_EditColors  
Rep\_EditFrameColors  
Rep\_EditFonts

**Rep\_EditRaster**

**HWND**      *hwndReport*

Diese Funktion öffnet einen Dialog zur Konfiguration eines Rasters. In diesem Raster werden alle mit der Maus durchgeführten geometrischen Modifikationen ausgeführt. Dies ist z.B. hilfreich zur schnellen, gleichmäßigen Anordnung von Objekten über Zonengrenzen hinweg und hilft dem Reportformular ein geordnetes Aussehen zu verleihen.

Die Rastereinstellungen werden im Reportformular selbst gespeichert.

**Funktionswert vom Typ int**

TRUE          Der Dialog wurde bestätigt.  
FALSE         Der Dialog wurde abgebrochen.

**Verwandte Funktionen**

Rep\_EditGrid

**Rep\_EditRecordControl**

**HWND**      *hwndReport*

Diese Funktion öffnet den Dialog zur Konfiguration der Datensatzvervielfachung. Hier kann ein Faktor eingestellt werden, welcher bestimmt, wie oft jeder Datensatz bei der Druckaufbereitung repliziert werden soll. Unter Zuhilfenahme von Event-Makros (siehe **Rep\_EditReportMacros**) kann die Datensatzvervielfachung gesteuert werden, Replikationen unterdrückt werden, oder die Ausgabe pro Replikation modifiziert werden.

**Funktionswert vom Typ int**

IDOK            Der Dialog wurde bestätigt.  
IDCANCEL       Der Dialog wurde abgebrochen.

## Rep\_EditReportMacros

**HWND**      *hwndReport*

Diese Funktion öffnet den Dialog zum Eingeben und Bearbeiten der globalen Makros eines Reports. Diese Makros werden während des Druckens automatisch aufgerufen, wenn eines der Ereignisse auftritt, das für eine Makroausführung vorbereitet ist, wie z.B. vor dem Drucken jedes Datensatzes, etc..

### **Funktionswert vom Typ int**

TRUE        Der Dialog wurde bestätigt.  
FALSE       Der Dialog wurde abgebrochen.

## Rep\_EditUnitSystem

**HWND**      *hwndReport*

Diese Funktion öffnet einen Dialog, der es dem Benutzer ermöglicht, das Maßeinheitensystem (metrisch oder zöllig) sowie den Dezimaltrenner (Komma oder Punkt) landesspezifisch einzustellen. Die Angaben werden zusammen mit dem Report gespeichert. Dadurch ist es möglich, sowohl Reports für den englischsprachigen Raum als auch für den metrischen Sprachraum gleichzeitig zu verwenden.

### **Funktionswert vom Typ int**

TRUE        Der Dialog wurde bestätigt.  
FALSE       Der Dialog wurde abgebrochen.

## Rep\_EditZones

**HWND**      *hwndReport*

Diese Funktion öffnet einen Dialog, der das Anlegen, Ändern und Löschen von Report-Zonen ermöglicht. Eine gleichnamige Funktion ist im Menü des Report-Editors verfügbar.

### **Funktionswert vom Typ int**

TRUE        Der Dialog wurde bestätigt.  
FALSE       Der Dialog wurde abgebrochen.

## Rep\_FindNextObject

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>HROB</b>	<b><i>ReferenzObject</i></b>
<b>const char *</b>	<b><i>ZonenName</i></b>
<b>const char *</b>	<b><i>ObjektName</i></b>
<b>long</b>	<b><i>ObjektTypen</i></b>

Diese Funktion ermittelt den Handle eines Objektes. Das Objekt wird wahlweise durch seinen Namen und/oder durch seinen Vorgänger bestimmt.

Es kann wahlweise im ganzen Report oder auf ausgewählten Zonen nach dem Objekt gesucht werden.

Es können alle Objekte oder ausgewählte Typen gesucht werden.

Der Parameter ***ReferenzObject*** bezeichnet einen Objekt-Handle. Wenn dieser den Wert **0** hat (oder kein gültiger Objekt-Handle ist), wird das erste Objekt ermittelt, das den Anforderungen der übrigen Parameter entspricht. Wenn ***ReferenzObject*** ein gültiger Objekt-Handle ist, wird das nächstfolgende Objekt ermittelt, das den Anforderungen der übrigen Parameter entspricht.

Der Parameter ***ZonenName*** ist eine Zeichenfolge, welche den Namen der Zonen beschreibt, auf denen Objekte gesucht werden sollen. Der Name der Zone darf auch *wilde* Zeichen beinhalten. Ein Fragezeichen (?) steht für ein beliebiges Zeichen im Namen. Ein Stern (\*) am Ende des Namens bedeutet, daß ab hier beliebig viele beliebige Zeichen folgen dürfen. Insbesondere spricht der Zonenname ***""***, alle Zonen des Reports an. Wenn–der Parameter NULL ist, werden ebenfalls alle Zonen des Reports angesprochen.

Der Parameter ***ObjektName*** ist eine Zeichenfolge, die den Namen der Objekte beschreibt, nach denen gesucht werden soll. Der Name des Objekts darf auch *wilde* Zeichen beinhalten. Ein Fragezeichen (?) steht für ein beliebiges Zeichen im Namen. Ein Stern (\*) am Ende des Namens bedeutet, daß ab hier beliebig viele beliebige Zeichen folgen dürfen. Insbesondere spricht der Namen ***""*** alle Objekte an. Die Angabe NULL spricht ebenfalls alle Objekte an.

Der Parameter ***ObjektTypen*** spezifiziert, nach welcher Art von Objekten gesucht werden soll. Es können ein oder mehrere Objekttypen durch ein logisches *oder* miteinander kombiniert werden. Die Liste aller zur Verfügung stehender Objekttypen ist im Anhang OBJ zusammengefaßt. Außerdem können folgende Objektgruppen angesprochen werden:

### ***Funktionswert vom Typ HROB***

Wert != 0	Der Handle des nächsten gefundenen Objektes.
Wert == 0	Es wurde kein (weiteres) Objekt gefunden.

**Beispiele**

Die folgende Schleife besetzt den Inhalt aller Datenfelder aller Zonen, deren Name mit **T** beginnt, mit dem Text **Hallo** vor:

```
HROB Objekt = (HROB) 0;

while ((Object = Rep_FindNextObject (hRep, Object, "T*", "*", OBJTYPE_FIELD)) != (HROB) 0)
    RepObj_SetText (Object, "Hallo");
```

Die folgende Sequenz ermittelt den aktuellen Inhalt des Datenfeldes **Klaus**:

```
HROB Objekt = (HROB) 0;
char Puffer [100];

Object = Rep_FindNextObject (hRep, Object, "*", "Klaus", OBJTYPE_FIELD);
RepObj_GetText (Object, Puffer, sizeof (Puffer));
```

**Verwandte Funktionen**

RepObj\_GetText  
RepObj\_SetText

**Rep\_GetChangeFlag**

**HWND**      **hwndReport**

Diese Funktion liefert zurück, ob ein Report verändert worden ist. Hierzu gehört das Anlegen oder Löschen von Objekten, das Vergrößern, Verschieben, etc.

Unabhängig davon, ob ein Report wirklich verändert wurde, kann die interne Merkvariable mittels der Funktion **Rep\_SetChangeFlag** jederzeit auf *verändert* oder auf *nicht verändert* gesetzt werden.

**Funktionswert vom Typ int**

- 1    Der Report wurde verändert.
- 0    Der Report wurde nicht verändert.

**Verwandte Funktionen**

Rep\_SetChangeFlag

## Rep\_GetCurrentZoneHandle

**HWND**      *hwndReport*

Diese Funktion liefert den Handle der augenblicklich aktiven Zone des Reports mit dem Handle *hwndReport* als Datentyp HREPZONE zurück.

Die Funktion liefert einen Funktionswert vom Typ HREPZONE

n          Handle der augenblicklich aktiven Zone  
0          Keine aktive Zone gefunden

### *Verwandte Funktionen*

Rep\_GetCurrentZoneName  
Rep\_GetZoneHandle

## Rep\_GetCurrentZoneTypeName

**HWND**      *hwndReport*  
**char \***     *Ergebnis*  
**int**         *MaxLngErgebnis*

Diese Funktion liefert den Namen des Typs der augenblicklich aktiven Zone auf dem Puffer *Ergebnis* zurück. Es werden nicht mehr als *MaxLngErgebnis* Zeichen übertragen. Wenn der Name länger ist, wird er abgeschnitten. Das Ergebnis ist in jedem Fall mit einem Nullbyte abgeschlossen. Dies gilt auch dann, wenn der Name abgeschnitten wurde.

Der Name des Zonentyps ZONE\_PAGEFOOT liefert beispielsweise *Seitenfuß* zurück.

Die Funktion liefert keinen Funktionswert zurück.

### *Verwandte Funktionen*

Rep\_GetCurrentZoneHandle  
Rep\_GetZoneHandle

## Rep\_GetCustomPointer

**HWND**      *hwndReport*

Diese Funktion liefert den *CustomPointer* eines Reports. Der *CustomPointer* bietet die Möglichkeit, eine beliebige Information, die ursächlich gar nicht zum Report gehört, an einen geöffneten Report zu binden. Der *CustomPointer* wird nicht im Report-Dokument gespeichert und wird beim Schließen des Reports verworfen. Zur dauerhaften Bindung von Information an einen Report siehe die Funktion *Rep\_SetCustomString*.

Der *CustomPointer* ist aus Sicht des Reports eine *Black Box*. Die Information wird ohne Ansehen von Form und Inhalt vom Report verwaltet.

Er kann jederzeit mittels der korrespondierenden Funktion **Rep\_SetCustomPointer** in den Report gesetzt werden, um später mittels **Rep\_GetCustomPointer** wieder ausgelesen zu werden.

Die Funktion liefert keinen Funktionswert zurück.

#### **Verwandte Funktionen**

Rep\_SetCustomPointer  
Rep\_SetCutomString  
Rep\_GetCustomString

### **Rep\_GetCustomString**

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>char *</b>	<b><i>Ergebnis</i></b>
<b>int</b>	<b><i>MaxLngErgebnis</i></b>

Diese Funktion liefert den *CustomString* eines Reports. Der *CustomString* ist ein String, der es ermöglicht, eine beliebige Information, die ursächlich gar nicht zum Report gehört, in einem Report zu speichern.

Diese Möglichkeit erlaubt es, bei der Einbindung des Reports in fremde Applikationen, Informationen im Report zu speichern, die die umgebende Applikation über den Report benötigt.

Der *CustomString* ist aus Sicht des Reports eine *Black Box*. Der String wird ohne Ansehen von Form und Inhalt vom Report verwaltet.

Er kann jederzeit mittels der korrespondierenden Funktion **Rep\_SetCustomString** in den Report gesetzt werden, um später mittels **Rep\_GetCustomString** wieder ausgelesen zu werden.

Der String wird auch korrekt *weitervererbt*, wenn ein Report geladen und mittels *Speichern als...* in eine andere Datei gespeichert wird.

Der *CustomString* wird in den durch den Parameter **Ergebnis** bezeichneten Puffer kopiert. Es werden nicht mehr als **MaxLngErgebnis** Zeichen übertragen. Wenn der *CustomString* länger ist, wird er abgeschnitten. Das Ergebnis ist in jedem Fall mit einem Nullbyte abgeschlossen. Dies gilt auch dann, wenn der *CustomString* abgeschnitten wurde.

Die Funktion liefert keinen Funktionswert zurück.

#### **Verwandte Funktionen**

Rep\_SetCustomString  
Rep\_SetCustomPointer  
Rep\_GetCustomPointer

## Rep\_GetDataFieldNamesInZone

<b>HWND</b>	<b><i>hwndRep</i></b>
<b>char *</b>	<b><i>ZoneName</i></b>
<b>char *</b>	<b><i>Result</i></b>
<b>int</b>	<b><i>MaxLngResult</i></b>
<b>char</b>	<b><i>Separator</i></b>

Diese Funktion erzeugt eine Liste aller in der durch ***ZoneName*** spezifizierten Zone verwendeten Datenfeldnamen. ***ZoneName*** darf auch eine *Wildcard* sein.

Die Liste wird in Form eines nullterminierten Strings erzeugt. Die einzelnen Datenfeldnamen sind untereinander durch den angegebenen ***Separator*** voneinander getrennt. Der ***Separator*** kann auch ein Tabulatorzeichen (hex 09) sein.

Der fertige Ergebnisstring wird in den durch Parameter ***Ergebnis*** angegebenen Puffer kopiert. Es werden maximal ***MaxLngErgebnis*** Zeichen übertragen. Wenn der Puffer zu kurz ist, wird das Ende des Strings abgeschnitten. In jedem Fall ist das Ergebnis mit einem Nullbyte abgeschlossen – auch dann, wenn es abgeschnitten wurde.

### ***Verwandte Funktionen***

Rep\_GetUsedDataFieldNames  
Rep\_DisplayUsedDataFieldNames

## Rep\_GetDataSource

<b>HWND</b>	<b><i>hwndRep</i></b>
<b>char *</b>	<b><i>DataSourceDescriptor</i></b>
<b>int</b>	<b><i>Size</i></b>

Diese Funktion liefert die augenblicklich für einen Report definierte Datenquelle in Form eines Strings zurück. Der String beschreibt die Datenquelle mit allen ihren Parametern. Die Syntax der Datenquellenbeschreibung ist in einer gesonderten Dokumentation beschrieben.

Der Parameter ***hwndRep*** definiert den Report, dessen Datenquelle ausgelesen werden soll.

Der Parameter ***DataSourceDescriptor*** ist ein Puffer, in dem die Datenquelle des Reports als nullterminierter String abgeleitet werden soll.

Der Parameter ***Size*** definiert die Länge des Puffers. Wenn die Datenquellenbeschreibung länger ist, wird sie abgeschnitten.

**Funktionswert**

Die Funktion liefert keinen Funktionswert zurück.

**Verwandte Funktionen**

Rep\_SetDataSource  
Rep\_EditDataSource

**Rep\_GetEditMode**

HWND      *hwndReport*

Diese Funktion liefert zurück, ob der Modus *Festlegen der Reihenfolge* ein- oder ausgeschaltet ist. Für den Fall, daß er eingeschaltet ist, liefert die Funktion darüberhinaus die Information, welches Objekt als nächstes zu bestimmen ist.

**Funktionswert vom Typ int**

- n > 0      Der Modus *Reihenfolge festlegen* ist aktiv. Das nächste Objekt, dessen Index festgelegt wird, ist das Objekt mit der Nummer **n**. Die Numerierung zählt von 1 an.
- 0          Der Modus *Reihenfolge festlegen* ist nicht aktiv.
- 1         Es ist ein Fehler aufgetreten. Eine entsprechende Fehlermeldung wurde bereits angezeigt.

**Verwandte Funktionen**

Rep\_SetEditMode

## Rep\_GetHelpFileName

<b>char *</b>	<b>Name</b>
<b>int</b>	<b>NameSize</b>

Diese Funktion liefert den Namen der Hilfedatei der Report-DLL zurück.

Der Parameter **Name** zeigt auf einen Puffer, auf dem der Name der Hilfedatei zurückgeliefert wird. Es werden nicht mehr als **NameSize** Zeichen übertragen. Das Ergebnis wird mit einem Nullbyte abgeschlossen – auch wenn es abgeschnitten wird.

**Diese Funktion liefert keinen Wert zurück.**

## Rep\_GetInstance

**Diese Funktion hat keine Parameter**

Diese Funktion liefert den *Instance-Handle* der Report-DLL zurück. Dieser kann bei späteren Windows API-Aufrufen, z.B. Auslesen von Ressourcen des Reports, mitgegeben werden.

**Funktionswert vom Typ HINSTANCE**

## Rep\_GetLanguage

<b>HWND</b>	<b>hwndReport</b>
-------------	-------------------

Diese Funktion liefert die augenblicklich für den Report **hwndReport** eingestellte Landessprachenkennung zurück. Die Landessprachenkennung bestimmt, welche Sprache ausgegeben wird, falls mehrere Sprachvarianten im Reportformular eingetragen sind.

**Funktionswert vom Typ int**

Als Funktionswert wird die internationale Sprachkennung zurückgeliefert, also z.B.

49	Deutsch
44	Englisch
31	Holländisch
etc.	

### **Verwandte Funktionen**

Rep\_SetLanguage

## Rep\_GetPageBgrColor

**HWND**      *hwndReport*

Diese Funktion liefert die augenblicklich als Seitenhintergrund eingestellte Farbe zurück. Die Farbe ist nur aktiv, wenn das Bit REPORT\_PAGEBGR\_SOLID gesetzt ist (siehe Funktion *Rep\_SetPageBgrMode*).

**Der Funktionswert vom Typ COLORREF liefert die Farbe zurück.**

### **Verwandte Funktionen**

Rep\_EditPageBackground  
Rep\_SetPageBgrBitmap  
Rep\_SetPageBgrMode  
Rep\_GetPageBgrMode  
Rep\_SetPageBgrColor

## Rep\_GetPageBgrMode

**HWND**      *hwndReport*

Diese Funktion liefert eine Liste von Status-Bits zurück, die den Zustand des Seitenhintergrundes bestimmen.

**Der Funktionswert vom Typ long ist eine Bitliste, die folgende Bits enthalten kann**

REPORT\_PAGEBGR\_SOLID  
Es ist ein farbiger Seitenhintergrund aktiv.

REPORT\_PAGEBGR\_BITMAP  
Es ist eine Bitmap aktiv

REPORT\_PAGEBGR\_FIXRATIO  
Wenn eine Bitmap aktiv ist, wird diese immer im Original-Seitenverhältnis dargestellt.

REPORT\_PAGEBGR\_CENTER  
Wenn eine Bitmap aktiv ist, wird diese immer mittig auf der Seite dargestellt.

### **Verwandte Funktionen**

Rep\_EditPageBackground  
Rep\_SetPageBgrBitmap  
Rep\_SetPageBgrMode  
Rep\_GetPageBgrColor  
Rep\_SetPageBgrColor

## Rep\_GetPrimarySelectedObject

**HWND**      *hwndReport*

Diese Funktion liefert den Handle des augenblicklich primär markierten Objekts zurück. Der Handle hat den Datentyp **HROB**. Es stehen eine Reihe von Funktionen zur Verfügung, mit denen die Eigenschaften von Objekten abgefragt oder verändert werden können. Diese Funktionen erwarten den Handle des betroffenen Objekts als Eingabeparameter.

### **Funktionswert vom Typ HROB**

hRob      Handle des augenblicklich primär markierten Objekts.  
0          Es ist momentan kein Eingabeobjekt aktiv.

### **Verwandte Funktionen**

RepObj\_GetType  
RepObj\_GetName  
RepObj\_GetFontFaceName  
RepObj\_GetFontHeight  
RepObj\_GetOutline  
RepObj\_SetOutline  
RepObj\_SetAlign  
RepObj\_SetFormat

## Rep\_GetPrinterNameAndPort

**HWND**      *hwndReport*  
**char \***     *Name*  
**int**         *NameSize*  
**char \***     *Port*  
**int**         *PortSize*

Diese Funktion liefert den Namen und den Port des augenblicklich im Report eingestellten Druckers ein. Wenn kein expliziter Drucker eingestellt ist, sondern der Standarddrucker verwendet wird, werden leere Strings zurückgeliefert.

Der Parameter **Name** zeigt auf einen Puffer, auf dem der Druckername zurückgeliefert wird. Es werden nicht mehr als **NameSize** Zeichen übertragen. Das Ergebnis wird mit einem Nullbyte abgeschlossen – auch wenn es abgeschnitten wird.

Der Parameter **Port** zeigt auf einen Puffer, auf dem der Drucker-Port zurückgeliefert wird. Es werden nicht mehr als **PortSize** Zeichen übertragen. Das Ergebnis wird mit einem Nullbyte abgeschlossen – auch wenn es abgeschnitten wird.

Die Funktion liefert keinen Funktionswert zurück.

## Rep\_GetProperty

HWND	<i>hwndReport</i>
int	<i>Property</i>
char *	<i>Buffer</i>
int	<i>Size</i>

Diese Funktion liest eine von zahlreichen Eigenschaften eines Reports aus und liefert diese an das aufrufende Programm zurück. Eine Eigenschaft ist entweder ein numerischer Wert oder eine Zeichenfolge.

Der Parameter **Property** bestimmt, welche Eigenschaft ausgelesen werden soll. Im Anhang REPPROP sind alle Properties des Reports genau beschrieben.

String-Properties werden auf dem Parameter **Buffer** zurückgeliefert. Wenn die Eigenschaft länger ist als **Size** Zeichen, wird sie abgeschnitten. In jedem Fall ist das Ergebnis nullterminiert.

### **Funktionswert vom Typ long**

Wenn die gelesene Eigenschaft ein numerischer Wert ist, wird dieser als Funktionswert zurückgeliefert. Wenn die Eigenschaft eine Zeichenfolge ist, wird immer der Wert 0 zurückgeliefert.

### **Verwandte Funktionen**

Rep\_SetProperty

## Rep\_GetReportMacro

HWND	<i>hwndReport</i>
int	<i>MacroType</i>
char *	<i>Buffer</i>
int	<i>Size</i>

Diese Funktion liefert den Code eines der im Report abgelegten Makros auf dem Parameter **Buffer** zurück. Es werden nicht mehr als **Size** Zeichen übertragen. Wenn der Makrocode länger ist, wird er abgeschnitten.

Die Größe des benötigten Puffers kann zuvor mittels der Funktion *Rep\_GetReportMacroSize* abgefragt werden.

Der Parameter **MacroType** bestimmt, welches Makro gelesen wird:

<b>WERT</b>	<b>Bedeutung</b>
REPMAC_AFTERREADRECORD	Nach dem Einlesen des Datensatzes
REPMAC_ONSTARTOFRECORD	Vor der Ausgabe des Datensatzes
REPMAC_ONSTARTOFREPHEAD	Vor der Ausgabe des Reportkopfes
REPMAC_ONSTARTOFFINALELEM	Vor der Ausgabe des Schlußelementes
REPMAC_ONSTARTOFPAGEHEAD	Vor der Ausgabe des Seitenkopfes
REPMAC_ONSTARTOFPAGEFOOT	Vor der Ausgabe des Seitenfußes
REPMAC_ONSTARTOFGROUPHEAD	Vor der Ausgabe des Gruppenkopfes
REPMAC_ONSTARTOFGROUPFOOT	Vor der Ausgabe des Gruppenfußes

#### **Funktionswert**

Die Funktion liefert keinen Funktionswert zurück.

#### **Verwandte Funktionen**

Rep\_GetReportMacroSize  
Rep\_SetReportMacro

### **Rep\_GetReportMacroSize**

<b>HWND</b>	<b>hwndReport</b>
<b>int</b>	<b>MacroType</b>

Diese Funktion liefert die Länge des Codes eines der im Report abgelegten Makros zurück. Diese Funktion kann verwendet werden, um die Länge des Puffers für die Funktion *Rep\_GetReportMacro* zu ermitteln.

Der Parameter **MacroType** bestimmt, von welchem Makro die Länge berechnet werden soll:

<b>WERT</b>	<b>Bedeutung</b>
REPMAC_AFTERREADRECORD	Nach dem Einlesen des Datensatzes
REPMAC_ONSTARTOFRECORD	Vor der Ausgabe des Datensatzes
REPMAC_ONSTARTOFREPHEAD	Vor der Ausgabe des Reportkopfes
REPMAC_ONSTARTOFFINALELEM	Vor der Ausgabe des Schlußelementes
REPMAC_ONSTARTOFPAGEHEAD	Vor der Ausgabe des Seitenkopfes
REPMAC_ONSTARTOFPAGEFOOT	Vor der Ausgabe des Seitenfußes
REPMAC_ONSTARTOFGROUPHEAD	Vor der Ausgabe des Gruppenkopfes
REPMAC_ONSTARTOFGROUPFOOT	Vor der Ausgabe des Gruppenfußes

#### **Funktionswert vom Typ int**

Anzahl Zeichen, die der Makro-Code benötigt

**Verwandte Funktionen**

Rep\_GetReportMacro  
Rep\_SetReportMacro

**Rep\_GetTitleFileName**

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>char *</b>	<b><i>DateiName</i></b>
<b>long</b>	<b><i>MaxLngName</i></b>

Diese Funktion liefert den Namen der Datei, aus der ein Report eröffnet wurde, auf dem Parameter ***DateiName*** zurück. Dies ist der Dateiname, der im Fenstertitel angezeigt wird. Es werden maximal ***MaxLng*** Zeichen übertragen. Das Ergebnis wird in jedem Fall mit einem Nullbyte abgeschlossen. Dies gilt auch dann, wenn es abgeschnitten wurde, weil der Puffer zu kurz ist.

Wenn es sich um einen neu angelegten Report handelt, wird ein leerer String zurückgeliefert.

Der Name wird in den durch den Parameter ***DateiName*** bezeichneten Puffer kopiert. Es werden nicht mehr als ***MaxLngName*** Zeichen übertragen. Wenn der Name länger ist, wird er abgeschnitten. Das Ergebnis ist in jedem Fall mit einem Nullbyte abgeschlossen. Dies gilt auch dann, wenn der Name abgeschnitten wurde.

Die Funktion liefert keinen Funktionswert zurück.

**Verwandte Funktionen**

Rep\_MakeUntitled

**Rep\_GetUsedDataFieldNames**

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>char *</b>	<b><i>Ergebnis</i></b>
<b>int</b>	<b><i>MaxLngErgebnis</i></b>
<b>char</b>	<b><i>Separator</i></b>

Diese Funktion erzeugt eine Liste aller in dem angegebenen Report verwendeten Datenfeldnamen. Dabei ist es gleichgültig, in welcher Zone des Reports die Datenfeldnamen verwendet werden.

Es werden auch die Namen von Datenfeldern in die Liste aufgenommen, die in Gruppen oder in Zonendefinitionen verwendet werden. Das gilt auch dann, wenn diese im Report nicht gedruckt werden.

Die Liste wird in Form eines nullterminierten Strings erzeugt. Die einzelnen Datenfeldnamen sind untereinander durch den angegebenen ***Separator*** voneinander getrennt. Der ***Separator*** kann auch ein Tabulatorzeichen (hex 09) sein.

Der fertige Ergebnisstring wird in den durch Parameter **Ergebnis** angegebenen Puffer kopiert. Es werden maximal **MaxLngErgebnis** Zeichen übertragen. Wenn der Puffer zu kurz ist, wird das Ende des Strings abgeschnitten. In jedem Fall ist das Ergebnis mit einem Nullbyte abgeschlossen – auch dann, wenn es abgeschnitten wurde.

#### **Verwandte Funktionen**

Rep\_DisplayUsedDataFieldNames

Rep\_GetDataFieldNamesInZone

### **Rep\_GetVersion**

<b>char *</b>	<b>Version</b>
<b>int</b>	<b>MaxLng</b>

Diese Funktion liefert die aktuelle Version der Report-DLL zurück. Die Version wird als Zeichenfolge auf der Variablen **Version** abgelegt. Es werden nicht mehr als **MaxLng** Zeichen übertragen. Das Ergebnis wird in jedem Fall mit einem Nullbyte abgeschlossen. Dies gilt auch dann, wenn es abgeschnitten wurde, weil der Puffer zu kurz ist.

Die Funktion liefert keinen Funktionswert zurück.

### **Rep\_GetZoneHandle**

<b>HWNDhwndReport</b>
<b>const char * Name</b>

Diese Funktion liefert den Handle der ersten Zone des Reports mit dem Handle **hwndReport**, die mit dem Namen **Name** übereinstimmt. Der Name kann auch eine *Wildcard* sein.

**Die Funktion liefert einen Funktionswert vom Typ HREPZONE**

n	Handle der ersten gefundenen Zone, die den angegebenen Namen trägt
0	Keine Zone des angegebenen Namens gefunden

#### **Verwandte Funktionen**

Rep\_GetCurrentZoneName

Rep\_GetCurrentZoneHandle

## Rep\_GetZoneNameList

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>char *</b>	<b><i>Result</i></b>
<b>int</b>	<b><i>MaxLngResult</i></b>
<b>char</b>	<b><i>Separator</i></b>

Diese Funktion erzeugt eine Liste aller in dem angegebenen Report verwendeten Zonennamen. Es werden auch die Namen der Zonen in die Liste aufgenommen, welche nicht gedruckt werden.

Die Liste wird in Form eines nullterminierten Strings erzeugt. Die einzelnen Zonennamen sind untereinander durch den angegebenen ***Separator*** voneinander getrennt. Der ***Separator*** kann auch ein Tabulatorzeichen (hex 09) sein.

Der fertige Ergebnisstring wird in den durch Parameter ***Result*** angegebenen Puffer kopiert. Es werden maximal ***MaxLngResult*** Zeichen übertragen. Wenn der Puffer zu kurz ist, wird das Ende des Strings abgeschnitten. In jedem Fall ist das Ergebnis mit einem Nullbyte abgeschlossen – auch dann, wenn es abgeschnitten wurde.

***Die Funktion liefert keinen Wert zurück***

## Rep\_IsReportClipboardAvailable

***Die Funktion hat keine Parameter***

Diese Funktion prüft, ob sich reportspezifische Informationen in der Zwischenablage befinden. Diese sind z.B. Objekte, die ausgeschnitten oder in die Zwischenablage kopiert wurden.

***Funktionswert vom Typ int***

- 1 Es befindet sich reportspezifische Information in der Zwischenablage.
- 0 Es befindet sich keine reportspezifische Information in der Zwischenablage.

## Rep\_IsReportWindow

<b>HWND</b>	<b><i>hwndReport</i></b>
-------------	--------------------------

Diese Funktion prüft, ob ***hwndReport*** ein gültiger Reportfenster-Handle ist. Beim Öffnen eines Reports mittels der Funktion ***Rep\_Open*** wird der Handle des geöffneten Reportfensters als Funktionswert zurückgeliefert.

***Funktionswert vom Typ int***

- 1 ***hwndReport*** ist ein gültiger Reportfenster-Handle.
- 0 ***hwndReport*** ist kein Reportfenster-Handle.

## Rep\_MakeUntitled

**HWND**      *hwndReport*

Diese Funktion setzt einen Report, welcher aus einer Datei geöffnet wurde oder auf andere Weise mit einer Titel-Datei assoziiert ist, in den Zustand **Ohne Namen**.

### **Funktionswert vom Typ int**

- 0 Die Funktion wurde fehlerfrei ausgeführt.
- 1 Es trat ein Fehler auf. Eine entsprechende Meldung wurde bereits angezeigt.

### **Verwandte Funktionen**

Rep\_GetTitleFileName

## Rep\_MuToString

**HWND**      *hwndReport*  
**long**      *Wert*  
**int**        *EinheitKategorie*  
**char \***     *Ergebnis*  
**int**        *MaxLngErgebnis*

Diese Funktion konvertiert eine Maßzahl, welche in Mikrometern angegeben ist, in eine Dezimalzahl-String. Dabei wird das für den Report eingestellte Maßeinheitensystem (metrisch oder englisch) berücksichtigt. Ein eventuell notwendiges Dezimalkomma wird ebenfalls den landesspezifischen Einstellungen des Reports entnommen.

Die zu konvertierende Maßzahl wird in Mikrometern als ganze Zahl auf dem Parameter **Wert** übergeben.

Der Parameter **EinheitKategorie** legt fest, ob die Ausgabe in einer kleinen Einheit, z.B. in Millimetern oder in einer größeren Einheit, z.B. Zentimetern erfolgen soll. Die Einheit hängt von diesem Parameter und vom im Report eingestellten Maßeinheitensystem ab:

UNITCAT\_SMALL            Die Konvertierung erfolgt in Millimeter, falls das metrische Einheitensystem eingestellt ist oder in Zoll, falls das englische Einheitensystem eingestellt ist.

UNITCAT\_MED              Die Konvertierung erfolgt in Zentimeter, falls das metrische Einheitensystem eingestellt ist oder in Zoll, falls das englische Einheitensystem eingestellt ist.

Die fertige Dezimalzahl wird als String auf dem Parameter **Ergebnis** abgelegt. Es werden nicht mehr als **MaxLngErgebnis** Zeichen übertragen. Das Ergebnis wird mit einem Nullbyte abgeschlossen. Dies gilt auch dann, wenn das Ergebnis zu lang ist und abgeschnitten wurde.

Die Funktion liefert keinen Funktionswert zurück.

## Rep\_Open

<b>HWND</b>	<i><b>hwndHost</b></i>
<b>char *</b>	<i><b>ReportDatei</b></i>
<b>char *</b>	<i><b>Reserved</b></i>
<b>int</b>	<i><b>NotificationMessage</b></i>

Diese Funktion eröffnet ein Reportfenster in einem vorgegebenen Host-Fenster. Das Host-Fenster ist ein Fenster beliebigen Typs, welches die Rahmenapplikation für die Abwicklung des Reports zur Verfügung stellen muß. Das Host-Fenster kann ein MDI-Fenster, ein SDI-Fenster oder ein modaler Dialog sein.

Der Parameter ***hwndHost*** bezeichnet den Fenster-Handle des Host-Fensters. Dieser darf nicht 0 sein und muß ein gültiges Windows-Fenster repräsentieren.

Der Parameter ***ReportDatei*** bezeichnet die Report-Datei, die eröffnet werden soll. Soll ein neuer Report angelegt werden, wie dies z.B. im Reporteditor der Fall ist, wenn der Benutzer die Funktion Neu aufruft, muß hier der Wert NULL angegeben werden.

Der Parameter ***Reserved*** ist für spätere Zwecke reserviert und sollte den Wert NULL enthalten.

Der Parameter ***NotificationMessage*** definiert eine Windows-Message, welche vom Report bei verschiedenen Ereignissen an das Host-Fenster gesendet wird. Wann und mit welchen Parametern eine Message gesendet wird, ist in einem gesonderten Kapitel genauer beschrieben.

### **Funktionswert vom Typ HWND**

- hwnd** Es wurde ein Reportfenster als Unterfenster des angegebenen Host-Fensters eröffnet. Der zurückgelieferte Handle wird als Eingabeparameter in allen Funktionen verwendet, die sich auf diesen Report beziehen sollen.
- 0** Es trat ein Fehler auf. Eine entsprechende Meldung wurde bereits angezeigt. Das Reportfenster ist **nicht** eröffnet worden.

## Rep\_Open\_Expanded

<b>HWND</b>	<i><b>hwndHost</b></i>
<b>char *</b>	<i><b>ReportDatei</b></i>
<b>char *</b>	<i><b>Reserved</b></i>
<b>int</b>	<i><b>NotificationMessage</b></i>
<b>long</b>	<i><b>Flags</b></i>

Diese Funktion öffnet ein Reportfenster. Sie erbringt dieselbe Leistung, wie die Funktion ***Rep\_Open***, ist allerdings um den Parameter ***Flags*** erweitert. Dieser ist eine Bitliste, in der folgende Bits gesetzt werden können:

**ROFLAG\_DENYNONE**

Wenn diese Option eingeschaltet wird, wird der Report in jedem Fall mit dem Zugriff **DENY NONE** eröffnet. Jede andere Applikation kann den Report gleichzeitig zum Lesen und Schreiben eröffnen.

**ROFLAG\_READONLY**

Wenn diese Option eingeschaltet wird, dürfen alle schreibenden Funktionen, wie z.B. **Rep\_Save** oder **Rep\_SaveAs** nicht mehr aufgerufen werden bzw. führen zu einem Fehler.

**ROFLAG\_NO\_PRINT\_IN\_PREVIEW**

Wenn diese Option eingeschaltet wird, wird der *Drucken*-Knopf in der Druckseitenvorschau ausgeblendet.

**ROFLAG\_CREATE\_AUTO\_HOST**

Wenn diese Option eingeschaltet wird, verliert der Parameter **hwndHost** seine Bedeutung. Vielmehr legt der Report intern ein unsichtbares Host-Fenster an. Dieses wird beim Schließen des Reports mittels der Funktion **Rep\_Close** automatisch wieder geschlossen. **hwndHost** darf in diesem Fall auch 0 sein.

**Funktionswert vom Typ HWND**

- hwnd** Es wurde ein Reportfenster als Unterfenster des angegebenen Host-Fensters eröffnet. Der zurückgelieferte Handle wird als Eingabeparameter in allen Funktionen verwendet, die sich auf diesen Report beziehen sollen.
- 0** Es trat ein Fehler auf. Eine entsprechende Meldung wurde bereits angezeigt. Das Reportfenster ist **nicht** eröffnet worden.

**Rep\_PasteObjectsToZone**

**HWND**      **hwndReport**

Diese Funktion fügt alle Objekte, die sich augenblicklich in der Zwischenablage befinden, in die aktuelle Zone des Reports ein.

Die Funktion liefert keinen Funktionswert zurück.

**Verwandte Funktionen**

Rep\_CopySelectedObjetscs

## Rep\_Preview\_File

<b>HWND</b>	<b><i>hwndParent</i></b>
<b>char *</b>	<b><i>ReportDatei</i></b>
<b>char *</b>	<b><i>DataSourceDescriptor</i></b>
<b>char *</b>	<b><i>ProfileDatei</i></b>
<b>long</b>	<b><i>Flags</i></b>

Diese Funktion eröffnet ein Druckseitenvorschau-Dialogfenster und stellt den gesamten Report in dieser Druckseitenvorschau dar.

Parameter ***hwndParent*** muß ein gültiges Windows-Fenster sein.

Das Layout des Reports wird durch das auf Parameter ***ReportDatei*** beschriebene Report-Formular bestimmt. Das Report-Formular muß zuvor mit dem Report-Editor erstellt werden.

Der Parameter ***DataSourceDescriptor*** beschreibt die Datenquelle und deren Parameter. Der Parameter ist ein String. Die Syntax der Datenquellenbeschreibung ist in einer gesonderten Dokumentation beschrieben. Wenn die *im Report eingestellte Datenquelle* verwendet werden soll, muß der Wert NULL angegeben werden.

Der Parameter ***ProfileDatei*** nennt den Namen einer Profile-Datei, in dem sich die Druckseitenvorschau interne Parameter *merken*. Es darf auch der Wert **NULL** angegeben werden. In diesem Fall speichert die Druckseitenvorschau keinerlei Einstellungen.

Mittels des Parameter ***Flags*** können eine oder mehrere Optionen spezifiziert werden. Wenn mehrere Optionen angegeben werden, sind mittels eines bitweisen **oder** zu verknüpfen. Wenn keine Flags gesetzt werden sollen, ist der Wert 0 anzugeben. Folgende Flags können gesetzt werden:

### **ROFLAG\_DENYNONE**

Wenn diese Option eingeschaltet wird, wird der Report in jedem Fall mit dem Zugriff **DENY NONE** eröffnet. Jede andere Applikation kann den Report gleichzeitig zum Lesen und Schreiben eröffnen.

### **ROFLAG\_READONLY**

Wenn diese Option eingeschaltet wird, dürfen alle schreibenden Funktionen, wie z.B. **Rep\_Save** oder **Rep\_SaveAs** nicht mehr aufgerufen werden bzw. führen zu einem Fehler.

### **Funktionswert vom Typ int**

- 0 Die Druckseitenvorschau wurde korrekt dargestellt. Der Dialog wurde durch den Benutzer beendet.
- <0 Es trat ein Fehler auf. Die Druckseitenvorschau konnte nicht dargestellt werden.
- 1 Die Funktion wurde vor dem Öffnen des Dialogs der Druckseitenvorschau durch den Benutzer abgebrochen

**Verwandte Funktionen**

Rep\_Preview\_Report  
Rep\_Preview\_Report\_NonModal  
Rep\_Print\_File  
Rep\_Print\_Report

**Rep\_Preview\_Report**

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>char *</b>	<b><i>DataSourceDescriptor</i></b>

Diese Funktion eröffnet ein Druckseitenvorschau-Dialogfenster und stellt den gesamten Report in dieser Druckseitenvorschau dar. Im Gegensatz zu der Funktion **Rep\_Preview\_File** erhält diese Funktion den Fenster-Handle eines bereits geöffneten Reports (siehe Funktion **Rep\_Open**) als Basis für die Druckseitenvorschau.

Der Parameter **DataSourceDescriptor** beschreibt die Datenquelle und deren Parameter. Der Parameter ist ein String. Die Syntax der Datenquellenbeschreibung ist in einer gesonderten Dokumentation beschrieben. Wenn die *im Report eingestellte Datenquelle* verwendet werden soll, muß der Wert NULL angegeben werden.

**Funktionswert vom Typ int**

- |    |  |
|----|--|
| 0  | Die Druckseitenvorschau wurde korrekt dargestellt. Der Dialog wurde durch den Benutzer beendet.      |
| <0 | Es trat ein Fehler auf. Die Druckseitenvorschau konnte nicht dargestellt werden.                     |
| 1  | Die Funktion wurde vor dem Öffnen des Dialogs der Druckseitenvorschau durch den Benutzer abgebrochen |

**Verwandte Funktionen**

Rep\_Preview\_File  
Rep\_Preview\_Report\_NonModal  
Rep\_Print\_File  
Rep\_Print\_Report

**Rep\_Preview\_Report\_NonModal**

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>char *</b>	<b><i>DataSourceDescriptor</i></b>

Diese Funktion eröffnet ein Druckseitenvorschau-Fenster und stellt den gesamten Report in dieser Druckseitenvorschau dar. Im Gegensatz zur Funktion **Rep\_Preview\_Report** wird die Vorschau in einem nicht mo-

dalen Fenster dargestellt. In einem möglicherweise parallel liegendem Editierfenster können Änderungen am Reportformular oder an der Datenquelle vorgenommen werden, und das Ergebnis kann direkt in der Druckseitenvorschau geprüft werden. Das nicht modale Fenster ist fest mit dem Report (*hwndReport*) assoziiert, sodaß ein erneuter Aufruf von *Rep\_Preview\_Report\_NonModal* kein weiteres Fenster eröffnet, sondern die Vorschau in dem vorhandenen Fenster aktualisiert wird.

Für weitere Informationen bezüglich der Druckseitenvorschau lesen Sie bitte unter *Rep\_Preview\_Report* nach.

**Wichtig:** Üblicherweise läuft die Druckseitenvorschau im selben Prozess und im selben *Thread* wie die aufrufende Applikation. Es kann jedoch über den Registry-Schlüssel

HKEY\_CURRENT\_USER\Software\Waimea Software\Report\Settings

***MultiThreadedPreview=1***

das Multithreading eingeschaltet werden, sodaß für den Preview eigene Prozessorzeit zur Verfügung gestellt wird.

#### ***Funktionswert vom Typ int***

- 0 Die Druckseitenvorschau wurde korrekt dargestellt. Der Dialog wurde durch den Benutzer beendet.
- <0 Es trat ein Fehler auf. Die Druckseitenvorschau konnte nicht dargestellt werden.
- 1 Die Funktion wurde vor dem Öffnen des Dialogs der Druckseitenvorschau durch den Benutzer abgebrochen

#### ***Verwandte Funktionen***

Rep\_Preview\_File  
Rep\_Preview\_Report\_NonModal  
Rep\_Print\_File  
Rep\_Print\_Report

## Rep\_Print\_File

HWND	<i>hwndParent</i>
char *	<i>ReportDatei</i>
char *	<i>DataSourceDescriptor</i>
HDC	<i>hdcDrucker</i>
int	<i>VonSeite</i>
int	<i>BisSeite</i>
int	<i>AnzahlKopien</i>
char *	<i>ProfileDatei</i>
long	<i>Flags</i>

Diese Funktion druckt einen kompletten Report auf dem eingestellten Drucker aus.

Parameter *hwndParent* muß ein gültiges Windows-Fenster sein.

Das Layout des Reports wird durch das auf Parameter *ReportDatei* beschriebene Report-Formular bestimmt. Das Report-Formular muß zuvor mit dem Report-Editor erstellt werden.

Der Parameter *DataSourceDescriptor* beschreibt die Datenquelle und deren Parameter. Der Parameter ist ein String. Die Syntax der Datenquellenbeschreibung ist in einer gesonderten Dokumentation beschrieben. Wenn die *im Report eingestellte Datenquelle* verwendet werden soll, muß der Wert NULL angegeben werden.

Wenn der Parameter *hdcDrucker* den Wert NULL hat, wird der Standard-Drucken-Dialog von Windows eröffnet, um weitere Druckparameter, wie z.B. den zu druckenden Seitenbereich anzufragen. Wenn hier ein gültiger *DeviceContextHandle* eines bereits eröffneten Druckers angegeben wird, entfällt der Dialog. Der angegebene *DeviceContext* wird zum Drucken verwendet.

Die Bedeutung der Parameter *VonSeite* und *BisSeite* hängt vom Wert des Parameters *hdcDrucker* ab. Wenn *hdcDrucker* den Wert NULL hat, werden die Werte *VonSeite* und *BisSeite* als Voreinstellung in den Drucken-Dialog eingeblendet. Wenn beide Parameter **0** sind, wird die Voreinstellung *alle Seiten Drucken* aktiviert. Wenn jedoch *hdcDrucken* einen bereits eröffneten *DeviceContext* beschreibt, grenzen die Parameter *VonSeite* und *BisSeite* den Seitenbereich ein, der gedruckt wird. Wenn alle Seiten gedruckt werden sollen, muß für beide Parameter der Wert **0** eingegeben werden.

Die Bedeutung des Parameters *AnzahlKopien* hängt vom Wert des Parameters *hdcDrucker* ab. Wenn *hdcDrucker* den Wert NULL hat, wird der Wert *AnzahlKopien* als Voreinstellung in den Drucken-Dialog eingeblendet. Wenn jedoch *hdcDrucken* einen bereits eröffneten *DeviceContext* beschreibt, bestimmt *AnzahlKopien* direkt, wieviele Exemplare des Reports gedruckt werden.

Der Parameter *ProfileDatei* nennt den Namen einer Profile-Datei, in dem sich der Report-Druck interne Parameter *merkt*. Es darf auch der Wert **NULL** angegeben werden. In diesem Fall speichert der Druck keinerlei Einstellungen.

Mittels des Parameter *Flags* können eine oder mehrere Optionen spezifiziert werden. Wenn mehrere Optionen angegeben werden, sind mittels eines bitweisen *oder* zu verknüpfen. Wenn keine Flags gesetzt werden sollen, ist der Wert **0** anzugeben. Folgende Flags können gesetzt werden:

**ROFLAG\_DENYNONE**

Wenn diese Option eingeschaltet wird, wird der Report in jedem Fall mit dem Zugriff **DENY NONE** eröffnet. Jede andere Applikation kann den Report gleichzeitig zum Lesen und Schreiben eröffnen.

**ROFLAG\_READONLY**

Wenn diese Option eingeschaltet wird, dürfen alle schreibenden Funktionen, wie z.B. **Rep\_Save** oder **Rep\_SaveAs** nicht mehr aufgerufen werden bzw. führen zu einem Fehler.

**Funktionswert vom Typ int**

- 0 Der Report wurde fehlerfrei gedruckt.
- <0 Es trat ein Fehler auf. Der Report konnte nicht gedruckt werden.
- 1 Die Funktion wurde vor dem Beginn des Drucks durch den Benutzer abgebrochen

**Verwandte Funktionen**

Rep\_Print\_Report  
Rep\_Preview\_File  
Rep\_Preview\_Report  
Rep\_Preview\_Report\_NonModal

**Rep\_Print\_Report**

**HWND**      *hwndReport*  
**char \***     *DataSourceDescriptor*

Diese Funktion druckt einen kompletten Report auf dem eingestellten Drucker aus. Im Gegensatz zu der Funktion **Rep\_Print\_File** erhält diese Funktion den Fenster-Handle eines bereits geöffneten Reports (siehe Funktion **Rep\_Open**) als Basis für den Druck.

Der Parameter **hwndRep** nennt den Handle des eröffneten Reports.

Der Parameter **DataSourceDescriptor** beschreibt die Datenquelle und deren Parameter. Der Parameter ist ein String. Die Syntax der Datenquellenbeschreibung ist in einer gesonderten Dokumentation beschrieben. Wenn die *im Report eingestellte Datenquelle* verwendet werden soll, muß der Wert NULL angegeben werden.

**Funktionswert vom Typ int**

- 0 Der Report wurde fehlerfrei gedruckt.
- <0 Es trat ein Fehler auf. Der Report konnte nicht gedruckt werden.
- 1 Die Funktion wurde vor dem Beginn des Drucks durch den Benutzer abgebrochen

**Verwandte Funktionen**

Rep\_Print\_File  
Rep\_Preview\_File  
Rep\_Preview\_Report  
Rep\_Preview\_Report\_NonModal

**Rep\_ReArrangeDisplay**

HWND      *hwndReport*

Diese Funktion zeichnet den gesamten Inhalt eines Reportfensters neu. Diese Funktion wird z.B. nach Editieren der Anzeigeparameter aufgerufen.

*Diese Funktion liefert keinen Wert zurück*

**Rep\_Save**

HWND      *hwndReport*

Diese Funktion schreibt einen Report in die Datei zurück, aus der er eröffnet wurde. Diese Datei wird im Titel des Reportfensters angezeigt und heißt *Titeldatei*. Der Name der Titeldatei kann mit Hilfe der Funktion **Rep\_GetTitleFileName** abgefragt werden.

Diese Funktion darf nur dann aufgerufen werden, wenn der Report *benannt* ist – wenn er also über eine Titeldatei verfügt. Wenn der Report noch *unbenannt* ist, z.B. weil er neu angelegt wurde, muß die Funktion **Rep\_SaveAs** aufgerufen werden.

Die Funktion **Rep\_GetTitleFileName** kann verwendet werden, um herauszufinden, ob der Report augenblicklich bereits benannt ist oder nicht.

Nach dem Speichern wird die interne Merkvariable *ChangeFlag*, die anzeigt, ob der Report verändert wurde, zurückgesetzt. Diese Variable kann mittels **Rep\_SetChangeFlag** auch explizit gesetzt oder rückgesetzt werden.

**Funktionswert vom Typ int**

- 0      Der Report wurde fehlerfrei gespeichert.
- 1     Es trat ein Fehler auf. Eine Meldung wurde bereits angezeigt. Der Report wurde **nicht** oder **nicht vollständig** gespeichert.

**Verwandte Funktionen**

Rep\_SaveAs  
Rep\_GetTitleFileName  
Rep\_SetChangeFlag

**Rep\_SaveAs**

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>char *</b>	<b><i>DateiName</i></b>

Diese Routine speichert einen *benannten* oder einen *unbenannten* Report in die Datei, deren Name auf Parameter ***DateiName*** angegeben wird. Nach dem Speichern wird die angegebene Datei zur *Titeldatei*. Der Name der Titeldatei kann mit der Funktion ***Rep\_GetTitleFileName*** jederzeit abgefragt werden.

Nach dem Speichern wird die interne Merkvariable *ChangeFlag*, die anzeigt, ob ein Report verändert wurde, zurückgesetzt. Diese Variable kann mittels ***Rep\_SetChangeFlag*** auch explizit gesetzt oder rückgesetzt werden.

0 Der Report wurde fehlerfrei gespeichert.

-1 Es trat ein Fehler auf. Eine Meldung wurde bereits angezeigt. Der Report wurde **nicht** oder **nicht vollständig** gespeichert.

**Verwandte Funktionen**

Rep\_Save  
Rep\_GetTitleFileName  
Rep\_SetChangeFlag

**Rep\_Select**

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>int</b>	<b><i>Modus</i></b>
<b>long</b>	<b><i>ObjektTypen</i></b>

Mit dieser Funktion können alle Objekte eines oder mehrerer Typen markiert werden. Der Parameter ***Modus*** entscheidet, in welcher Zone Objekte markiert werden:

RM\_CURRENTZONE Es werden Objekte der aktuellen Zone markiert

Der Parameter ***ObjektTypen*** nennt die Objekttypen, die markiert werden sollen. Es können mehrere Objekttypen durch logisches *oder* miteinander kombiniert werden. Es können alle Objekttypen angegeben werden, die im Anhang OBJ aufgelistet sind.

Insbesondere kann auch eine der folgenden vordefinierten Objektgruppen angegeben werden:

OBJTYPE\_ALL

Alle Objekte

Die Angabe des Wertes 0 als Objekttyp bewirkt, daß ein Dialog eröffnet wird, der es dem Benutzer ermöglicht, die zu markierenden Objekte vorzugeben.

Die Funktion liefert keinen Funktionswert zurück.

### Rep\_SelectedObjectsToBackground

HWND	<i>hwndReport</i>
int	<i>Modus</i>

Diese Funktion setzt alle markierten Objekte einer Zone in den Hintergrund. Objekte, die im Hintergrund liegen, werden von Objekten im Vordergrund überdeckt. Die Überdeckung ist nur dann sichtbar, wenn die Objekte im Vordergrund nicht *transparent* sind.

Der Parameter *Modus* muß den Wert RM\_CURRENTZONE haben.

Die Funktion liefert keinen Funktionswert zurück.

#### **Verwandte Funktionen**

Rep\_SelectedObjectsToForeground

## Rep\_SelectedObjectsToForeground

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>int</b>	<b><i>Modus</i></b>

Diese Funktion setzt alle markierten Objekte einer Zone in den Vordergrund. Objekte, die im Vordergrund liegen, überdecken Objekte im Hintergrund. Die Überdeckung ist nur dann sichtbar, wenn die Objekte im Vordergrund nicht *transparent* sind.

Der Parameter ***Modus*** muß den Wert RM\_CURRENTZONE haben.

Die Funktion liefert keinen Funktionswert zurück.

### ***Verwandte Funktionen***

Rep\_SelectedObjectsToBackground

## Rep\_SetChangeFlag

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>int</b>	<b><i>Flag</i></b>

Diese Funktion setzt die interne Merkvariable, ob ein Report seit dem Öffnen oder Anlegen oder nach dem Speichern verändert worden ist. Wenn die Variable auf *verändert* steht, wird vor dem Schließen des Reports gefragt, ob die Änderungen gespeichert werden sollen oder nicht.

Der Parameter ***Flag*** definiert, wie die Variable besetzt werden soll:

- 0 Der Report wurde nicht verändert (frisch eröffnet oder gespeichert).
- 1 Der Report wurde verändert.

Die Funktion liefert keinen Funktionswert zurück.

### ***Verwandte Funktionen***

Rep\_GetChangeFlag

## Rep\_SetCopyList

**HWND**            *hwndRep*  
**const char \***    *CopyList*

Diese Funktion setzt eine neue Ausfertigungsliste für den Report. Die Syntax von Ausfertigungslisten ist gesondert beschrieben.

Die Funktion hat keinen Funktionswert.

### **Verwandte Funktionen**

Rep\_EditCopyList

## Rep\_SetCustomPointer

**HWND**            *hwndReport*  
**void \***            *Pointer*

Diese Funktion setzt den *CustomPointer* eines Reports. Der *CustomPointer* bietet die Möglichkeit, eine beliebige Information, die ursächlich gar nicht zum Report gehört, an einen geöffneten Report zu binden. Der *CustomPointer* wird nicht im Report-Dokument gespeichert und wird beim Schließen des Reports verworfen. Zur dauerhaften Bindung von Information an einen Report siehe die Funktion **Rep\_SetCustomString**.

Der *CustomPointer* ist aus Sicht des Reports eine *Black Box*. Die Information wird ohne Ansehen von Form und Inhalt vom Report verwaltet.

Er kann jederzeit mittels der korrespondierenden Funktion **Rep\_GetCustomPointer** wieder ausgelesen werden.

Die Funktion liefert keinen Funktionswert zurück.

### **Verwandte Funktionen**

Rep\_GetCustomPointer  
Rep\_GetCustomString  
Rep\_SetCustomString

## Rep\_SetCustomString

**HWND**            *hwndReport*  
**char \***            *String*

Diese Funktion setzt den *CustomString* eines Reports. Der *CustomString* ist ein String, der es ermöglicht, eine beliebige Information, die ursächlich gar nicht zum Report gehört, in einem Report zu speichern.

Diese Möglichkeit erlaubt es, bei der Einbindung des Reports in fremde Applikationen, Informationen im Report zu speichern, die die umgebende Applikation über den Report benötigt.

Der *CustomString* ist aus Sicht des Reports eine *Black Box*. Der String wird ohne Ansehen von Form und Inhalt vom Report verwaltet.

Er kann jederzeit mittels der korrespondierenden Funktion **Rep\_GetCustomString** wieder ausgelesen werden.

Die Funktion liefert keinen Funktionswert zurück.

#### **Verwandte Funktionen**

Rep\_GetCustomString  
Rep\_GetCustomPointer  
Rep\_SetCustomPointer

## **Rep\_SetDataSource**

<b>HWND</b>	<b><i>hwndRep</i></b>
<b>const char *</b>	<b><i>DataSourceDescriptor</i></b>

Diese Funktion definiert eine neue Datenquelle und deren Parameter für einen Report. Die bisherige Datenquelle wird überschrieben. Die Syntax der Datenquellenbeschreibung ist in einer gesonderten Dokumentation beschrieben.

Der Parameter ***hwndRep*** definiert den Report, der verändert werden soll.  
Der Parameter ***DataSourceDescriptor*** ist ein String, der die neue Datenquelle definiert.

#### **Funktionswert**

Die Funktion liefert keinen Funktionswert zurück.

#### **Verwandte Funktionen**

Rep\_GetDataSource  
Rep\_EditDataSource

## Rep\_SetEditOrderMode

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>int</b>	<b><i>Modus</i></b>

Diese Funktion schaltet den Arbeitsmodus ein oder aus, in dem die Reihenfolge der Objekte definiert werden kann. Der Parameter ***Modus*** hat folgende Bedeutung:

- n Der Modus *Reihenfolge festlegen* wird **eingeschaltet**. Das erste Objekt, dessen Reihenfolge bestimmt werden soll, ist das Objekt mit der Nummer **n**. **n** zählt ab eins.
- 0 Der Modus *Reihenfolge festlegen* wird **ausgeschaltet**. Alle Objekte, die angeordnet wurden, bleiben in der durch den Benutzer bestimmten Reihenfolge.
- 1 Der Modus *Reihenfolge festlegen* wird **eingeschaltet**. Die Nummer des ersten Objekts, dessen Reihenfolge bestimmt werden soll, wird durch einen Dialog beim Benutzer angefragt.

***Die Funktion liefert keinen Funktionswert zurück.***

## Rep\_SetErrorHandlerCallback

<b>long</b>	<b><i>CallbackAddress</i></b>
-------------	-------------------------------

Diese Funktion meldet die Adresse einer externen Routine für die Fehlerbehandlung an. Nach dem Aufruf dieser Routine wird kein interaktiver Fehlerdialog mehr durchgeführt. Beim Auftreten eines Fehlers wird anstelle des Fehlerdialogs die Funktion aufgerufen, deren Adresse hier als Parameter ***CallbackAddress*** angegeben wurde.

**Diese Funktion ist für den Betrieb auf Servern sehr wichtig.**

Die Callback-Funktion wird mit einem Parameter aufgerufen, der auf die Adresse des Fehlertextes zeigt. Der Fehlertext ist bereits mittels des Sprachübersetzungssystems übersetzt und wird in der aktuellen Landessprache übergeben.

Zum Deaktivieren der Callback muß die Funktion mit dem Wert NULL aufgerufen werden.

***Die Funktion liefert keinen Funktionswert zurück.***

## Rep\_SetFieldnamesCallback

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>char *</b>	<b><i>Dll</i></b>
<b>char *</b>	<b><i>Routine</i></b>

Diese Routine definiert eine externe Funktion in einer externen DLL als Callback-Funktion, die immer dann aufgerufen wird, wenn der Benutzer innerhalb eines der zahlreichen Dialoge des Reports auf den Knopf **Datenfelder...** drückt.

Wenn diese Funktion nicht definiert wird, blendet der Report eine eigene Funktion zur Auswahl des Namens eines Datenfeldes ein. Diese hängt vom Typ und Namen der Datenquelle ab, die augenblicklich eingestellt ist.

Wenn diese Leistung durch eine externe Funktion abgewickelt werden soll, so kann diese hierdurch angemeldet werden.

*Dll* benennt die DLL, in welcher sich die externe Funktion befindet. *Routine* benennt die Funktion, welche in dieser DLL aufgerufen werden soll. Die Funktion wird nach den Konventionen der Sprache C aufgerufen und muß folgende Parameter haben:

**Funktion (HWND *hwndReport*, HROB *Objekt*, char \**Puffer*, int *MaxLngPuffer*)**

Die Funktion liefert als Ergebnis einen Datenfeldnamen im Parameter ***Puffer*** ab. Der Puffer ist maximal ***MaxLngPuffer*** Zeichen lang. Mehr Zeichen dürfen nicht übertragen werden. Auf dem ersten Parameter wird der Handle des Report-Fensters übergeben, falls die externe Funktion irgendwelche Funktionen des Report-Interfaces aufrufen will. Der zweite Parameter bezeichnet den Handle des Objektes, welches augenblicklich bearbeitet wird, während diese Funktion aufgerufen wird. Wenn kein Bezug auf ein Objekt vorliegt, ist dieser Parameter 0.

Die externe Funktion liefert einen Funktionswert vom Typ ***int*** zurück:

0	Es wurde ein Datenfeldname in <b><i>Puffer</i></b> bereitgestellt
-1	Es wurde kein Datenfeldname bereitgestellt

Den Wert -1 liefert die Funktion insbesondere dann, wenn der Benutzer einen Dialog abgebrochen hat oder wenn ein Fehler aufgetreten ist.

Eine bereits angemeldete externe Funktion kann wieder abgemeldet werden, indem die Funktion ***Rep\_SetFieldnamesCallback*** erneut aufgerufen wird und für die Parameter ***Dll*** und ***Routine*** der Wert **NULL** angegeben wird.

Die Funktion liefert keinen Funktionswert zurück.

### **Verwandte Funktionen**

Rep\_DeclareAvailableFields

## Rep\_SetLanguage

HWND	<i>hwndReport</i>
Int	<i>Language</i>

Diese Funktion setzt eine neue aktive Landessprache für den Report *hwndReport*. Die aktive Landessprache ist diejenige, in der der Report ausgegeben wird, falls mehrere Sprachen im reportformular erfaßt wurden. Außerdem entscheidet die aktive Sprache darüber, welche Sprachvariante beeinflusst wird, wenn Texte verändert werden.

Der Parameter *Language* bestimmt, welche Sprache eingestellt werden soll. Der Wert entspricht dem üblichen internationalen Code, also z.B.

49	Deutsch
44	Englisch
31	Holländisch

etc.

### Verwandte Funktionen

Rep\_GetLanguage

## Rep\_SetObjectsAlignment

HWND	<i>hwndReport</i>
char *	<i>ZonenId</i>
char *	<i>ObjektName</i>
long	<i>ObjektTypen</i>
int	<i>Ausrichtung</i>

Die Routine setzt die horizontale Ausrichtung aller Objekte eines Reports oder bestimmter Zonen eines Reports.

*ZonenId* nennt die Bezeichnung der Zone, deren Objekte beeinflusst werden sollen. Wenn mehrere Zonen dieselbe Bezeichnung haben, werden alle Zonen angesprochen, deren Bezeichnung *ZonenId* entspricht. Wenn *ZonenId* mit einem Stern endet, werden alle Zonen angesprochen, deren Bezeichnung mit *Zoneld* beginnt (Wildcard). Die Angabe "\*" ist eine Wildcard, die alle Bezeichnungen matcht. Mit dieser Angabe werden alle Objekte aller Zonen angesprochen.

*ObjektName* nennt den Namen des Objekts, das beeinflusst werden soll. Wenn mehrere Objekte denselben Namen haben, werden sie alle angesprochen. Wenn *ObjektName* mit einem Stern endet, werden alle Objekte angesprochen, deren Name mit *ObjektName* beginnt (Wildcard). Die Angabe "\*" ist eine Wildcard, die alle Objektnamen matcht. Mit dieser Angabe werden alle Objekte angesprochen.

**ObjektTypen** bezeichnet die Liste aller Objekttypen, die beeinflusst werden sollen. Es können mehrere Objekttypen miteinander verodert werden. Die Liste aller möglichen Objekttypen ist im Anhang OBJ zusammengefaßt.

**Ausrichtung** bestimmt die gewünschte Ausrichtung der durch **ZonenId** und **ObjektName** bestimmten Objekte. Folgende Werte können angegeben werden:

```
ALIGN_LEFT
ALIGN_HORICENTER
ALIGN_RIGHT
```

Objekte, die über keine Ausrichtung verfügen, bleiben unverändert.

Die Routine liefert keinen Funktionswert zurück.

## Rep\_SetObjectsFormat

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>char *</b>	<b><i>ZonenId</i></b>
<b>char *</b>	<b><i>ObjektName</i></b>
<b>long</b>	<b><i>ObjektTypen</i></b>
<b>char *</b>	<b><i>FormatString</i></b>

Die Routine setzt die Formatierungsanweisung aller Objekte eines Reports oder bestimmter Zonen eines Reports.

**ZonenId** nennt die Bezeichnung der Zone, deren Objekte beeinflusst werden sollen. Wenn mehrere Zonen dieselbe Bezeichnung haben, werden alle Zonen angesprochen, deren Bezeichnung **ZonenId** entspricht. Wenn **ZonenId** mit einem Stern endet, werden alle Zonen angesprochen, deren Bezeichnung mit **ZonenId** beginnt (Wildcard). Die Angabe "\*" ist eine Wildcard, die alle Bezeichnungen matcht. Mit dieser Angabe werden alle Objekte aller Zonen angesprochen.

**ObjektName** nennt den Namen des Objekts, das beeinflusst werden soll. Wenn mehrere Objekte denselben Namen haben, werden sie alle angesprochen. Wenn **ObjektName** mit einem Stern endet, werden alle Objekte angesprochen, deren Name mit **ObjektName** beginnt (Wildcard). Die Angabe "\*" ist eine Wildcard, die alle Objektnamen matcht. Mit dieser Angabe werden alle Objekte angesprochen.

**ObjektTypen** bezeichnet die Liste aller Objekttypen, die beeinflusst werden sollen. Es können mehrere Objekttypen miteinander verodert werden. Die Liste aller möglichen Objekttypen ist im Anhang OBJ zusammengefaßt.

**FormatString** definiert die Format-Anweisung, die dem Objekt zugeordnet werden soll. Die Format-Anweisungen für Objekte vom Typ Gleitkomma, Festkomma, Datum und Uhrzeit werden in einem gesonderten Kapitel beschrieben.

Objekte, die über keine Formatierung verfügen, werden nicht beeinflusst.

Die Routine liefert keinen Funktionswert zurück.

## Rep\_SetPageBgrBitmap

<b>HWND</b>	<i>hwndReport</i>
<b>char *</b>	<i>BitmapDatei</i>

Diese Funktion setzt eine neue Bitmap als Seitenhintergrund. Die Bitmap wird als Dateiname angegeben.

Die Bitmap ist nur aktiv, wenn mittels der Funktion *Rep\_SetPageBgrMode* auch das Bit `REPORT_PAGEBGR_BITMAP` gesetzt wird.

**Die Funktion liefert zurück, ob die Bitmap erfolgreich geladen wurde**

0	Die Bitmap wurde erfolgreich geladen
-1	Es ist ein Fehler aufgetreten

### **Verwandte Funktionen**

Rep\_EditPageBackground  
Rep\_GetPageBgrMode  
Rep\_SetPageBgrMode  
Rep\_GetPageBgrColor  
Rep\_SetPageBgrColor

## Rep\_SetPageBgrColor

<b>HWND</b>	<i>hwndReport</i>
<b>COLORREF</b>	<i>Farbe</i>

Diese Funktion setzt eine neue Hintergrundfarbe für die Seiten des Reports.

Die Farbe ist nur aktiv, wenn mittels der Funktion *Rep\_SetPageBgrMode* auch das Bit `REPORT_PAGEBGR_SOLID` gesetzt wird.

**Die Funktion hat keinen Funktionswert.**

### **Verwandte Funktionen**

Rep\_EditPageBackground  
Rep\_SetPageBgrBitmap  
Rep\_GetPageBgrMode  
Rep\_SetPageBgrMode  
Rep\_GetPageBgrColor

## Rep\_SetPageBgrMode

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>long</b>	<b>Bits</b>

Diese Funktion setzt eine Liste von Status-Bits, die den Zustand des Seitenhintergrunds desinieren.

### ***Folgende Bits können gesetzt werden***

REPORT\_PAGEBGR\_SOLID

Der Seitenhintergrund wird wie der aktuell eingestellten Hintergrundfarbe gefärbt

REPORT\_PAGEBGR\_BITMAP

Es wird die aktuell eingestellte Bitmap ausgegeben.

REPORT\_PAGEBGR\_FIXRATIO

Wenn eine Bitmap aktiv ist, wird diese immer im Original-Seitenverhältnis dargestellt.

REPORT\_PAGEBGR\_CENTER

Wenn eine Bitmap aktiv ist, wird diese immer mittig auf der Seite dargestellt.

***Die Funktion liefert keinen Funktionswert zurück.***

### ***Verwandte Funktionen***

Rep\_EditPageBackground

Rep\_SetPageBgrBitmap

Rep\_GetPageBgrMode

Rep\_GetPageBgrColor

Rep\_SetPageBgrColor

## Rep\_SetPrinterNameAndPort

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>char *</b>	<b><i>Name</i></b>
<b>char *</b>	<b><i>Port</i></b>

Dies Funktion setzt in einem geöffneten Report (***hwndReport***) den Namen des Druckers (***Name***) und optional des Anschlusses (***Port***) auf welchem eine Ausgabe erfolgen soll. Die Angabe eines Leerstrings für den Namen bewirkt, daß kein Drucker im Report festgelegt wird, das heißt, daß der jeweils konfigurierte Standarddrucker benutzt wird.

*Die Funktion liefert einen Funktionswert vom Typ int zurück.*

- 1 es ist ein Fehler aufgetreten
- 0 es wurde der Standarddrucker eingestellt
- 1 es wurde der Drucker **Name** eingestellt

## Rep\_SetProperty

HWND	<i>hwndReport</i>
int	<i>Property</i>
long	<i>Value</i>
char *	<i>Buffer</i>

Diese Funktion setzt eine von zahlreichen Eigenschaften eines Reports. Eine Eigenschaft ist entweder ein numerischer Wert oder eine Zeichenfolge.

Der Parameter **Property** bestimmt, welche Eigenschaft gesetzt werden soll.

Im Anhang REPPROP sind alle Eigenschaften des Reports beschrieben. Bitte beachten Sie, daß nicht alle Eigenschaften gesetzt werden können. Manche Eigenschaften können nur ausgelesen werden.

Die Parameter **Value** und **Buffer** werden verwendet, um die zu setzende Eigenschaft zu übergeben. **Value** wird verwendet, um numerische Eigenschaften zu übergeben und **Buffer** wird verwendet, um String-Eigenschaften zu übergeben.

**Die Funktion liefert keinen Funktionswert zurück**

#### **Verwandte Funktionen**

Rep\_GetProperty

## **Rep\_SetReportMacro**

```
HWND      hwndReport
int        MacroType
const char * Code
```

Diese Funktion definiert einen neuen Code für eines der im Report abgelegten Makros.

Der Parameter **MacroType** bestimmt, welches Makro neu definiert wird:

<b>WERT</b>	<b>Bedeutung</b>
REPMAC_AFTERREADRECORD	Nach dem Einlesen des Datensatzes
REPMAC_ONSTARTOFRECORD	Vor der Ausgabe des Datensatzes
REPMAC_ONSTARTOFREPHEAD	Vor der Ausgabe des Reportkopfes
REPMAC_ONSTARTOFFINALELEM	Vor der Ausgabe des Schlußelementes
REPMAC_ONSTARTOFPAGEHEAD	Vor der Ausgabe des Seitenkopfes
REPMAC_ONSTARTOFPAGEFOOT	Vor der Ausgabe des Seitenfußes
REPMAC_ONSTARTOFGROUPHEAD	Vor der Ausgabe des Gruppenkopfes
REPMAC_ONSTARTOFGROUPFOOT	Vor der Ausgabe des Gruppenfußes

#### **Funktionswert**

Die Funktion liefert keinen Funktionswert

#### **Verwandte Funktionen**

Rep\_GetReportMacro

## Rep\_SpacingDialog

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>int *</b>	<b><i>Modus</i></b>

Diese Funktion öffnet den Dialog zum Angleichen der Breite oder Höhe mehrerer selektierter Objekte eines Reports.

**Modus** ist eine durch logisches *Oder* kombinierte Liste eines oderer mehrerer der folgenden Werte:

ARR_ADJUSTWIDTH	Breite angleichen
ARR_ADJUSTHEIGHT	Höhe angleichen

Wenn der Dialog bestätigt wurde, wird auf dem Parameter **Modus** der vom Benutzer ausgewählte Zustand zurückgeliefert. **Modus ist eine Zeigervariable!**

Diese Funktion selbst führt keine Änderung durch. Erst durch Aufruf der Funktion **Rep\_ArrangeSelectedObjects** und Übergabe der von **Rep\_SpacingDialog** erhaltenen Werte werden die selektierten Objekte wie angegeben arrangiert.

### **Funktionswert vom Typ int**

0	Der Dialog wurde bestätigt. <b>Modus</b> liefert den ausgewählten Wert.
-1	Der Dialog wurde abgebrochen.

### **Verwandte Funktionen**

Rep\_AlignDialog  
Rep\_ArrangeDialog  
Rep\_ArrangeSelectedObjects

## Rep\_SetUserCallback

```
HWND hwndReport  
long (CALLBACK * UserCallback)  
      (HWND hRep, const char *Event, REP_CALLBACK_DATA *Data, long lParam)  
long CallbackFlags
```

Diese Funktion meldet beim Report eine Callback-Funktion an, die während der Formatierung (Druck, Vorschau) bei verschiedenen Ereignissen - ähnlich den Makroereignissen - aufgerufen wird.

**UserCallback** ist die anzumeldende Callback-Funktion. Diese wird mit folgenden Parametern aufgerufen:

- Der Handle des Reportfensters
- Ein String, der den Typ des Ereignisses definiert
- Ein Zeiger auf eine Struktur vom Typ **REP\_CALLBACK\_DATA**
- Eine Bitliste, in der verschiedene Flags gesetzt werden können.

**CallbackFlags** ist für spätere Zwecke reserviert und muss den Wert 0 haben.

Folgenden Strings können als Grund für den Aufruf der Callback als Ereignis übermittelt werden:

```
OnStartReport  
OnEndReport  
BeforePrintSingleCopy  
AfterReadRecord  
BeforePrintReportHeader  
BeforePrintPageHeader  
BeforePrintPageFooter  
BeforePrintRecord  
BeforePrintFinalElement  
BeforePrintGroupHeader  
BeforePrintGroupFooter
```

Die Namen der Ereignisse sind selbsterklärend und werden daher nicht näher beschrieben. Im Rahmen von Weiterentwicklungen muss davon ausgegangen werden, dass später eventuell weitere Ereignisse definiert werden.

Die Struktur **REP\_CALLBACK\_DATA** übermittelt verschiedene Laufzeitinformationen an die Callback-Funktion. Es muss davon ausgegangen werden, dass diese Struktur später erweitert wird. Aus diesem Grund wird zur Erkennung der Version die aktuelle Länge der Struktur in Bytes als erste Variable der Struktur übermittelt.

Augenblicklich liefert die Struktur folgende Variablen:

```
int Size  
int CurrentRecord  
int CurrentPage  
int TotalPages  
int PassType
```

**Size** gibt die Länge der Struktur in Bytes an.

**CurrentRecord** nennt die Nummer des Datensatzes, der gerade ausgegeben wird.

**CurrentPage** nennt die Nummer der Seite, die gerade formatiert wird.

**TotalPages** nennt die Gesamtzahl der Seiten des Reports. Diese Variable ist nur dann korrekt besetzt, wenn sich im Reportformular mindestens an einer Stelle ein Objekt vom Typ *Seitennummer* befindet, welches die Gesamtanzahl der Seiten anzeigt. Denn nur in diesem Fall wird dieser Wert vom Report zuvor ermittelt. Aus Optimierungsgründen entfällt diese Ermittlung nämlich, wenn der Wert nicht benötigt werden sollte.

**PassType** hier wird der Typ des Durchlaufs übermittelt. Dieser ist im Moment immer REP\_PASS\_PRINT, da während aller anderen Durchläufe die Callback-Routine nicht aufgerufen wird..

#### **Funktionswert**

Die Funktion liefert keinen Funktionswert

## **Rep\_Wizzard**

<b>HWND</b>	<b><i>hwndReport</i></b>
<b>char *</b>	<b><i>Descriptor</i></b>
<b>long</b>	<b><i>Flags</i></b>
<b>char *</b>	<b><i>CustomBitmap</i></b>

Diese Funktion öffnet einen Wizzard-Dialog. Der Benutzer wird durch eine Reihe von Unterdialogen geführt, die die Gestaltung eines neuen Reports ermöglichen. Wenn am Ende der Dialog bestätigt wird, wird der Report, dessen Handle als Parameter übergeben wird, neu gestaltet. Alle Zonen und Objekte, die bereits in dem Report enthalten sind, werden entfernt. In der Regel wird diese Funktion direkt im Anschluß an das Öffnen eines neuen, leeren Reports mittels der Funktion *Rep\_Open* aufgerufen.

**hwndReport** ist der Handle des Reports, der neu gestaltet werden soll.

**Descriptor** ist ein Descriptor-String für eine Datenquelle. Dieser Parameter wird nur ausgewertet, wenn das Bit *REPORT\_WIZZARD\_USEDESCRIPTOR* im Parameter *Flags* gesetzt ist. In diesem Fall fragt der Wizzard keine Datenquelle an, sondern setzt direkt auf der hier angegebenen Datenquelle auf.

**Flags** ist eine Bitliste, in der folgende Bits gesetzt werden können:

#### **REPORT\_WIZZARD\_USEDESCRIPTOR**

Wenn dieses Bit gesetzt ist, wird die auf Parameter *Descriptor* angegebene Datenquelle verwendet. Wenn dieses Bit nicht gesetzt ist, generiert der Wizzard einen Report für den Zugriff auf eine ODBC-Datenbank. Der Name der ODBC-Datenquelle und der Name der Tabelle werden im Wizzard angefragt.

**CustomBitmap** Wenn auf diesem Parameter der Name einer Datei übergeben wird, die eine gültige Bitmap enthält, wird diese Bitmap anstelle der voreingestellten Bitmap zur Animation in den Wizzard eingeblendet. Die Bitmap sollte im Format 137 x 255 oder zumindest dieser Größe nahekommen, da der Dialog dynamisch zur Laufzeit aufgebaut wird und die Größen der einzelnen Komponenten aufeinander abgestimmt sind. Wenn die angegeben Bitmap nicht gefunden wird, erscheint die Standard-Bitmap.

**Funktionswert vom Typ int**

IDOK	Der Dialog wurde bestätigt. Der Report wurde neu gestaltet.
IDCANCEL	Der Dialog wurde abgebrochen oder es ist ein Fehler aufgetreten.

## Rep\_Zoom

HWND	<i>hwndReport</i>
int	<i>Mode</i>

Diese Funktion stellt den Zoom-Faktor der Bildschirmdarstellung des Reports ein. Der Parameter **Modus** steuert, welcher Zoom-Faktor gewählt wird:

<b>REP_ZOOMMODE_WINDOWWIDTH</b>	Zoomen auf optimale Fensterbreite
<b>REP_ZOOMMODE_DIALOG</b>	Der Faktor wird im Dialog angefragt
<b>10 ... 300</b>	Absoluter Zoom-Faktor in Prozent

**Funktionswert vom Typ int**

0	Der Zoom-Faktor wurde umgestellt.
-1	Fehler oder Dialog abgebrochen.

## 5. Zonenfunktionen

Dieses Kapitel beschreibt alle Funktionen, die die Kontrolle einer einzelnen Report-Zone ermöglichen. Diese Funktionen werden *Zonenfunktionen* genannt.

Alle *Zonenfunktionen* werden mit einem *Zonen-Handle* versorgt, welcher die Zone, auf die sich die Funktion bezieht, beschreibt. Ein Zonen-Handle ist eine Variable vom Typ **HREPZONE**. Verschiedene Funktionen liefern einen Zonen-Handle als Funktionswert zurück. Beispielsweise liefert die Funktion **Rep\_GetCurrentZoneHandle** den Handle der augenblicklich aktiven Zone zurück.

### RepZone\_DeleteObject

<b>HREPZONE</b>	<b><i>hRepZone</i></b>
<b>HROB</b>	<b><i>hObject</i></b>
<b>long</b>	<b><i>Flags</i></b>

Diese Funktion löscht das Objekt mit dem Handle ***hObject*** aus der Zone mit dem Handle ***hRepZone***. Der Parameter ***Flags*** muß 0 sein und ist für spätere Zwecke reserviert.

**Die Funktion liefert keinen Funktionswert zurück.**

### RepZone\_GetIndicatorField

<b>HREPZONE</b>	<b><i>hRepZone</i></b>
<b>char *</b>	<b><i>Name</i></b>
<b>int</b>	<b><i>NameSize</i></b>
<b>char *</b>	<b><i>Content</i></b>
<b>int</b>	<b><i>ContentSize</i></b>

Diese Funktion liefert den assoziierten Namen und Inhalt des Datenfeldes zurück, auf den sich die Zone ***hRepZone*** bezieht, falls eine gleichartige Zone mehrfach vorhanden ist.

Die Parameter ***Name*** und ***NameSize*** bezeichnen den Puffer und seine Größe, in dem der Name des Datenfeldes abgelegt werden soll, mit dem die Zone assoziiert ist. Die Parameter ***Content*** und ***ContentSize*** bezeichnen den Puffer und seine Größe, in dem der Inhalt des Datenfeldes abgelegt werden soll, mit dem die Zone Assoziiert ist.

Die Funktion liefert keinen Funktionswert zurück.

#### **Verwandte Funktionen**

RepZone\_SetIndicatorField

## RepZone\_GetName

HREPZONE	<i>hRepZone</i>
char *	<i>Result</i>
int	<i>Size</i>

Diese Funktion liefert den Namen der Zone zurück.

Die Funktion liefert keinen Funktionswert zurück.

### *Verwandte Funktionen*

RepZone\_SetName

## RepZone\_ResetOccurenceBits

HREPZONE	<i>hRepZone</i>
long	<i>Bits</i>

Mit dieser Funktion wird die seitenbezogene Ausgabe der durch *hRepZone* angegebenen Zone konfiguriert. Wurde mittels der Funktion *RepZone\_SetOccurenceBits* die Ausgabe für die Zone explizit aus- oder angeschaltet, so kann der Zustand mit *RepZone\_ResetOccurenceBits* wieder in den Ursprungszustand zurückgesetzt werden. Zur Erläuterung des Bitfeldes *Bits* siehe *RepZone\_SetOccurenceBits*.

Die Funktion liefert keinen Funktionswert zurück.

### *Verwandte Funktionen*

RepZone\_SetOccurenceBits

## RepZone\_SetIndicatorField

HREPZONE	<i>hRepZone</i>
const char *	<i>Name</i>
const char *	<i>Content</i>

Diese Funktion setzt den assoziierten Namen und Inhalt des Datenfeldes, auf das sich die Zone *hRepZone* bezieht, falls eine gleichartige Zone mehrfach vorhanden ist.

Die Funktion liefert keinen Funktionswert zurück.

### *Verwandte Funktionen*

RepZone\_GetIndicatorField

**RepZone\_SetName**

```
HREPZONE hRepZone  
Const char * NewName
```

Diese Funktion setzt einen neuen Namen für die Zone *hRepZone*.

Die Funktion liefert keinen Funktionswert zurück.

***Verwandte Funktionen***

RepZone\_GetName

**RepZone\_SetOccurenceBits**

**HREPZONE**    *hRepZone*  
**long**                *Bits*

Mit dieser Funktion wird die seitenbezogene Ausgabe der durch *hRepZone* angegebenen Zone konfiguriert. *Bits* ist ein Bitmuster welches durch Setzen der Bits die jeweilige Ausgabe aus- oder anschaltet. Mit Hilfe der Funktion *RepZone\_ResetOccurenceBits* kann die Ausgabe der Zone wieder normiert werden. Folgende Werte können in *Bits* kombiniert werden:

ZOCC_NOT_ON_FIRST_PAGE	Keine Ausgabe auf der ersten Seite
ZOCC_NOT_ON_LAST_PAGE	Keine Ausgabe auf der letzten Seite (nur für den Seitenfuß möglich).
ZOCC_NOT_ON_INNER_EVEN_PAGES	Keine Ausgabe auf geraden Folgeseiten
ZOCC_NOT_ON_INNER_ODD_PAGES	Keine Ausgabe auf ungeraden Folgeseiten
ZOCC_NOT_AT_REPORT_END	Keine Ausgabe nach Druck des letzten Datensatzes (nur für Gruppenfuß sinnvoll)
ZOCC_GROUP_ON_EVERYPAGE	Die Ausgabe erfolgt auch zu Beginn jeder Seite (nur für Gruppenkopf sinnvoll)
ZOCC_NOT_BEFORE_FIELD_NOT_EMPTY	Die Ausgabe erfolgt erstmals, wenn das assoziierte Datenfeld nicht leer ist (nur für Gruppenkopf sinnvoll).
veraltete nicht mehr zu verwendende Werte:	ersetzt durch:
ZOCC_NOTONFIRSTPAGE	ZOCC_NOT_ON_FIRST_PAGE
ZOCC_NOTONLASTPAGE	ZOCC_NOT_ON_LAST_PAGE
ZOCC_NOTATREPORTEND	ZOCC_NOT_AT_REPORT_END
ZOCC_NOTBEFOREFIELDNOTEMPTY	ZOCC_NOT_BEFORE_FIELD_NOT_EMPTY

Die Funktion liefert keinen Funktionswert zurück.

**Verwandte Funktionen**

RepZone\_ResetOccurenceBits

## 6. Objektfunktionen

Dieses Kapitel beschreibt alle Funktionen, die die Kontrolle eines einzelnen Report-Objekts ermöglichen. Diese Funktionen werden *Objektfunktionen* genannt.

Alle *Objektfunktionen* werden mit einem *Objekt-Handle* versorgt, welcher das Objekt, auf das sich die Funktion bezieht, beschreibt. Ein Objekt-Handle ist eine Variable vom Typ **HROB**. Verschiedene Funktionen liefern einen Objekt-Handle als Funktionswert zurück. Beispielsweise liefert die Funktion **Rep\_GetPrimarySelectedObject** den Handle des augenblicklich vorrangig markierten Objekts zurück. Darüberhinaus werden in manchen Notification-Messages die Handles von Objekten übergeben.

Beispielsweise kann ein Objekt, direkt nachdem es angelegt wurde, von außen verändert werden, da der Report nach dem Anlegen dem Host-Fenster eine Notification-Message vom Typ **REPMSG\_MODIFYNEWOBJECT** sendet. Diese erhält den Handle des Objekts als Parameter und mittels der im folgenden beschriebenen Funktionen kann das frisch angelegte Objekt nach Belieben verändert werden.

Näheres über das Senden von Notification-Messages wird in dem hierfür vorgesehenen Kapitel beschrieben.

### RepObj\_GetFontDisplayString

<b>HROB</b>	<b><i>hRob</i></b>
<b>char *</b>	<b><i>Destination</i></b>
<b>int</b>	<b><i>Size</i></b>

Diese Funktion liefert den Anzeigetext zur Darstellung des Schriftnamens und der Schriftgröße des Objekts zurück, das durch den Handle ***hRob*** definiert wird.

Der Parameter ***Destination*** zeigt auf einen Puffer, auf dem der Anzeigetext zurückgeliefert wird. Es werden nicht mehr als ***Size*** Zeichen übertragen. Das Ergebnis wird mit einem Nullbyte abgeschlossen – auch wenn es abgeschnitten wird.

Diese Funktion wird hauptsächlich von Status-Anzeige-Modulen benutzt.

Die Funktion liefert keinen Funktionswert zurück.

#### **Verwandte Funktionen**

RepObj\_GetLongProperty  
RepObj\_GetFontProperty  
RepObj\_GetTypeDisplayString  
RepObj\_GetSizeDisplayString  
RepObj\_GetPositionDisplayString

## RepObj\_GetFontProperty

<b>HROB</b>	<b>hRob</b>
<b>Int</b>	<b>Property</b>

Diese Funktion liefert eine von verschiedenen Fonteigenschaften des Objekts mit dem Handle **hRob** als ganzzahligen Wert zurück. Die Liste aller möglichen Fonteigenschaften ist im Anhang FOBJPROP (Font Properties für Objekte) beschrieben.

Wenn eine Property angegeben wird, die nicht definiert oder zumindest für den vorliegenden Objekttyp ohne Bedeutung ist, wird der Wert 0 zurückgeliefert.

Die Funktion liefert den gewünschten Wert als **long** zurück.

### *Verwandte Funktionen*

RepObj\_SetFontProperty  
RepObj\_GetLongProperty  
RepObj\_GetStringProperty

## RepObj\_GetLongProperty

<b>HROB</b>	<b>hRob</b>
<b>Int</b>	<b>Property</b>

Diese Funktion liefert eine von verschiedenen numerischen Eigenschaften des Objekts mit dem Handle **hRob** als ganzzahligen Wert zurück. Die Liste aller möglichen ganzzahligen Eigenschaften ist im Anhang LOBJPROP (Long Properties für Objekte) beschrieben.

Wenn eine Property angegeben wird, die nicht definiert oder zumindest für den vorliegenden Objekttyp ohne Bedeutung ist, wird der Wert 0 zurückgeliefert.

Die Funktion liefert den gewünschten Wert als **long** zurück.

### *Verwandte Funktionen*

RepObj\_SetLongProperty  
RepObj\_GetFontProperty  
RepObj\_GetStringProperty

## RepObj\_GetOutline

<b>HROB</b>	<b>hRob</b>
<b>LONGRECT *</b>	<b>Outline</b>

Diese Funktion liefert die Größe und Position des Objekts **hRob** zurück. Die Position von Objekten, die relativ zu anderen Objekten angeordnet sind, kann mit dieser Funktion nicht ermittelt werden.

Die Größe und Position des Objekts werden auf die Strukturvariablen der Struktur **Outline** vom Typ **LONGRECT** geschrieben, deren Adresse als Parameter übergeben wird. Die Variablen sind sämtlich vom Typ **long** und werden wie folgt besetzt:

<b>left</b>	linker Rand des Objekts in Mikrometern relativ zur Zone
<b>top</b>	oberer Rand des Objekts in Mikrometern relativ zur Zone
<b>right</b>	rechter Rand des Objekts in Mikrometern relativ zur Zone
<b>bottom</b>	unterer Rand des Objekts in Mikrometern relativ zur Zone

Die Funktion liefert keinen Funktionswert zurück.

#### Verwandte Funktionen

Rep\_FindNextObject  
 RepObj\_GetFontFaceName  
 RepObj\_GetFontHeight  
 RepObj\_GetType

### RepObj\_GetPositionDisplayString

<b>HROB</b>	<b>hRob</b>
<b>char *</b>	<b>Destination</b>
<b>int</b>	<b>Size</b>
<b>LONGRECT *</b>	<b>Outline</b>

Diese Funktion liefert den Anzeigetext zur Darstellung der Position des Objekts zurück, das durch den Handle **hRob** definiert wird.

Der Parameter **Destination** zeigt auf einen Puffer, auf dem der Anzeigetext zurückgeliefert wird. Es werden nicht mehr als **Size** Zeichen übertragen. Das Ergebnis wird mit einem Nullbyte abgeschlossen – auch wenn es abgeschnitten wird.

Wird für **Outline** der Wert NULL übergeben, so wird die aktuelle Position des Objektes zurückgeliefert, anderenfalls wird die in **Outline** angegebene Position in den Anzeigetext umgewandelt (siehe **RepObj\_GetOutline**).

Diese Funktion wird hauptsächlich von Status-Anzeige-Modulen benutzt.

Die Funktion liefert keinen Funktionswert zurück.

#### Verwandte Funktionen

RepObj\_GetOutline  
 RepObj\_GetTypeDisplayString  
 RepObj\_GetSizeDisplayString  
 RepObj\_GetFontDisplayString

## RepObj\_GetSizeDisplayString

<b>HROB</b>	<b>hRob</b>
<b>char *</b>	<b>Destination</b>
<b>int</b>	<b>Size</b>
<b>LONGRECT *</b>	<b>Outline</b>

Diese Funktion liefert den Anzeigetext zur Darstellung der Größe des Objekts zurück, das durch den Handle **hRob** definiert wird.

Der Parameter **Destination** zeigt auf einen Puffer, auf dem der Anzeigetext zurückgeliefert wird. Es werden nicht mehr als **Size** Zeichen übertragen. Das Ergebnis wird mit einem Nullbyte abgeschlossen – auch wenn es abgeschnitten wird.

Wird für **Outline** der Wert NULL übergeben, so wird die aktuelle Größe des Objektes zurückgeliefert, andernfalls wird die in **Outline** angegebene Größe in den Anzeigetext umgewandelt (siehe **RepObj\_GetOutline**). Diese Funktion wird hauptsächlich von Status-Anzeige-Modulen benutzt.

Die Funktion liefert keinen Funktionswert zurück.

### Verwandte Funktionen

RepObj\_GetOutline  
RepObj\_GetTypeDisplayString  
RepObj\_GetPositionDisplayString  
RepObj\_GetFontDisplayString

## RepObj\_GetStringProperty

<b>HROB</b>	<b>hRob</b>
<b>int</b>	<b>Property</b>
<b>char *</b>	<b>Result</b>
<b>int</b>	<b>Size</b>

Diese Funktion liefert eine von verschiedenen String-Eigenschaften des Objekts mit dem Handle **hRob** als nullterminierte Zeichenkette zurück. Die Liste aller möglichen ganzzahligen Eigenschaften ist im Anhang SOBJPROP (String Properties für Objekte) beschrieben.

Das Ergebnis wird im Puffer **Result** abgelegt. Wenn das Ergebnis länger als **Size** Zeichen ist, wird es abgeschnitten. In jedem Fall ist das Ergebnis nullterminiert.

Die Funktion hat keinen Funktionswert.

### Verwandte Funktionen

RepObj\_SetStringProperty  
RepObj\_GetFontProperty  
RepObj\_GetLongProperty

## RepObj\_GetTypeDisplayString

<b>HROB</b>	<i><b>hRob</b></i>
<b>char *</b>	<i><b>Destination</b></i>
<b>int</b>	<i><b>Size</b></i>

Diese Funktion liefert den typenspezifischen Anzeigetext ("Datenfeld", "Bild", "Datum",...) des Objekts zurück, das durch den Handle ***hRob*** definiert wird. Der Typ eines Objektes kann mittels ***RepObj\_GetType*** abgefragt werden.

Der Parameter ***Destination*** zeigt auf einen Puffer, auf dem der Anzeigetext zurückgeliefert wird. Es werden nicht mehr als ***Size*** Zeichen übertragen. Das Ergebnis wird mit einem Nullbyte abgeschlossen – auch wenn es abgeschnitten wird.

Diese Funktion wird hauptsächlich von Status-Anzeige-Modulen benutzt.

Die Funktion liefert keinen Funktionswert zurück.

### ***Verwandte Funktionen***

RepObj\_GetType  
RepObj\_GetSizeDisplayString  
RepObj\_GetPositionDisplayString  
RepObj\_GetFontDisplayString

## RepObj\_SetFontProperty

<b>HROB</b>	<b>hRob</b>
<b>int</b>	<b>Property</b>
<b>long</b>	<b>Value</b>

Diese Funktion setzt eine von verschiedenen Fonteigenschaften des Objekts mit dem Handle **hRob**. Die Liste aller möglichen Fonteigenschaften ist im Anhang FOBJPROP (Font Properties für Objekte) beschrieben.

Der Wert der Property wird auf dem Parameter **Value** als **long** übergeben. Wenn eine Property angegeben wird, die nicht definiert ist oder zumindest für den vorliegenden Objekttyp ohne Bedeutung ist, bleibt die Funktion ohne Wirkung.

Die Funktion hat keinen Funktionswert.

### *Verwandte Funktionen*

RepObj\_GetFontProperty  
RepObj\_SetLongProperty  
RepObj\_SetStringProperty

## RepObj\_SetLongProperty

<b>HROB</b>	<b>hRob</b>
<b>int</b>	<b>Property</b>
<b>long</b>	<b>Value</b>

Diese Funktion setzt eine von verschiedenen ganzzahligen Eigenschaften des Objekts mit dem Handle **hRob**. Die Liste aller möglichen ganzzahligen Eigenschaften ist im Anhang LOBJPROP (Long Properties für Objekte) beschrieben.

Der Wert der Property wird auf dem Parameter **Value** als **long** übergeben. Wenn eine Property angegeben wird, die nicht definiert ist oder die für den vorliegenden Objekttyp ohne Bedeutung ist, bleibt die Funktion ohne Wirkung.

Die Funktion hat keinen Funktionswert.

### *Verwandte Funktionen*

RepObj\_GetLongProperty  
RepObj\_SetFontProperty  
RepObj\_SetStringProperty

## RepObj\_SetOutline

<b>HROB</b>	<b>hRob</b>
<b>LONGRECT *</b>	<b>Outline</b>

Diese Funktion setzt eine neue Größe und Position für das Objekts **hRob**. Die Position von Objekten, die relativ zu anderen Objekten angeordnet sind, kann mit dieser Funktion nicht verändert werden.

Die gewünschte Größe und Position des Objekts müssen vor Aufruf der Funktion auf die Strukturvariablen der Struktur **Outline** vom Typ **LONGRECT** zugewiesen werden, deren Adresse als Parameter übergeben wird. Die Variablen sind sämtlich vom Typ **long** und müssen wie folgt besetzt werden:

<b>left</b>	linker Rand des Objekts in Mikrometern relativ zur Zone
<b>top</b>	oberer Rand des Objekts in Mikrometern relativ zur Zone
<b>right</b>	rechter Rand des Objekts in Mikrometern relativ zur Zone
<b>bottom</b>	unterer Rand des Objekts in Mikrometern relativ zur Zone

Die Funktion liefert keinen Funktionswert zurück.

### **Verwandte Funktionen**

Rep\_FindNextObject  
RepObj\_GetOutline

## RepObj\_SetStringProperty

<b>HROB</b>	<b>hRob</b>
<b>int</b>	<b>Property</b>
<b>const char *</b>	<b>Value</b>

Diese Funktion setzt eine von verschiedenen String-Eigenschaften des Objekts mit dem Handle **hRob**. Die Liste aller möglichen String-Eigenschaften ist im Anhang (String Properties für Objekte) beschrieben.

Der Wert der Property wird auf dem Parameter **Value** als zeiger auf eine Zeichenkette übergeben. Wenn eine Property angegeben wird, die nicht definiert ist oder die für den vorliegenden Objekttyp ohne Bedeutung ist, bleibt die Funktion ohne Wirkung.

Die Funktion hat keinen Funktionswert.

### **Verwandte Funktionen**

RepObj\_GetStringProperty  
RepObj\_SetFontProperty  
RepObj\_SetLongProperty

## 7. Formatdeskriptoren

Für die automatische Formatierung von Zahlen, Datums- und Uhrzeitangaben muß eine detaillierte Beschreibung des gewünschten Formates definiert werden. Um eine möglichst system- und maschinenunabhängige Darstellung zu gewährleisten, wurde eine Formatbeschreibung in Stringform gewählt.

Insgesamt werden vier verschiedene Formate definiert:

- Format für ganze Zahlen**
- Format für Gleitkommazahlen**
- Format für Datumsangaben**
- Format für Uhrzeiten**

Diese vier Formate werden in den folgenden Kapiteln erläutert. Sie sind inhaltlich verwandt, unterscheiden sich jedoch wesentlich voneinander.

### 7.1. Datumsformat

Ein Datumsformat ist ein String, der aus einer Aneinanderreihung von Platzhaltern für die Komponenten *Wochentag*, *Tag*, *Monat* und *Jahr* besteht. Vor, zwischen und hinter den Platzhaltern kann ein beliebiger Text stehen, der Bestandteil des fertig formatierten Datums wird.

Jeder Platzhalter beginnt mit einem Prozentzeichen. Dann folgt ein Buchstabe, der den Platzhaltertyp bestimmt. Es sind folgende Platzhalter definiert, die ein Datumsformat enthalten kann:

<b>%Wx</b>	Wochentagsname
<b>%Dx</b>	Tag
<b>%Mx</b>	Monat
<b>%Yx</b>	Jahresangabe

Dahinter folgt eine Variantenbezeichnung x. Die Variantenbezeichnung ist eine einstellige Ziffer und bedeutet je nach Platzhalter:

<b>%W0</b>	Keine Wochentagseinblendung
<b>%W1</b>	Wochentag als zweibuchstabiges Kürzel einblenden
<b>%W2</b>	Wochentag als vollen Namen einblenden
<b>%D0</b>	Keine Einblendung des Tages
<b>%D1</b>	Tag als möglichst kurze Zahl ohne führende Null einblenden
<b>%D2</b>	Tag als zweistellige Zahl, ggf. mit einer führenden Null einblenden
<b>%D3</b>	Tag als Zahl mit englischem Ordinalindex (1st, 2nd, 3rd, 4th, etc.)

<b>%M0</b>	Keine Einblendung des Monats
<b>%M1</b>	Monat als möglichst kurze Zahl ohne führende Null einblenden
<b>%M2</b>	Monat als zweistellige Zahl, ggf. mit einer führenden Null einblenden
<b>%M3</b>	Monatsname als dreibuchstabile Abkürzung einblenden
<b>%M4</b>	Monatsname in voller Länge einblenden
<b>%Y0</b>	Keine Einblendung des Jahres
<b>%Y1</b>	Jahr als zweistellige Zahl ohne Jahrhundert einblenden
<b>%Y2</b>	Jahr als vierstellige Zahl mit Jahrhundert einblenden

Durch die Reihenfolge der Platzhalter können auch die englischsprachige Anordnung (Monat, Tag, Jahr) und die sortierfreundige Anordnung (Jahr, Monat, Tag) bestimmt werden. insgesamt werden folgende Anordnungen der Platzhalter unterstützt:

**Tag, Monat, Jahr**  
**Monat, Tag, Jahr**  
**Jahr, Monat, Tag**

Zwischen den Platzhaltern dürfen beliebige Zeichen oder Textfolgen stehen. Die Anordnung der Wochentagsbezeichnung (%W) ist beliebig.

Folgende Beispiele zeigen die Darstellung des Datums 3. Januar 1955 mit verschiedenen Datumsformaten:

<b>Format</b>	<b>Darstellung</b>
%W0%D1.%M1.%Y2	3.1.1955
%W0%D2/%M2/%Y1	03/01/55
%W0%D1-%M3-%Y2	3-Jan-1955
%W0%M3, %D1, %Y2	Jan, 3, 1955
%W2, den %D1. %M4 %Y2	Montag, den 3. Januar 1955
%M4, %D3, %Y2	Januar, 3rd, 1955

## Verweis auf das System-Datumsformat

Anstelle eines expliziten Formats kann der Formatstring auch einen Verweis auf das im System länderspezifisch vorgegebene System-Datumsformat enthalten, welches in den Ländereinstellungen definiert werden kann. Unter Windows steht ein kurzes und ein langes System-Datumsformat zur Verfügung.

In diesem Fall besteht der gesamte Formatstring nur aus einer der beiden Zeichenfolgen:

<b>%SS</b>	Kurzes System-Datumsformat verwenden
<b>%SL</b>	Langes System-Datumsformat verwendet

Sollte der Formatstring weitere Zeichen beinhalten, werden diese ignoriert.

## 7.2. Uhrzeitformat

Ein Uhrzeitformat ist ein String, der aus einer Aneinanderreihung von Platzhaltern für die Komponenten *Stunden*, *Minuten* und *Sekunden* besteht. Vor, zwischen und hinter den Platzhaltern kann ein beliebiger Text stehen, der Bestandteil der fertig formatierten Uhrzeit wird.

Jeder Platzhalter beginnt mit einem Prozentzeichen. Dann folgt ein Buchstabe, der den Platzhaltertyp bestimmt. Es sind folgende Platzhalter definiert, die ein Uhrzeitformat enthalten kann:

<b>%Hx</b>	Stunden (24-Stunden-Anzeige)
<b>%hx</b>	Stunden (12-Stunden-Anzeige)
<b>%Ux</b>	Minuten
<b>%Sx</b>	Sekunden

Dahinter folgt eine Variantenbezeichnung x. Die Variantenbezeichnung ist eine einstellige Ziffer und bedeutet je nach Platzhalter:

<b>%H0</b>	Keine Stundenanzeige
<b>%H1</b>	Stunden als möglichst kurze Zahl ohne führende Null einblenden
<b>%H2</b>	Stunden zweistellig, ggf. mit einer führenden Null einblenden

Entsprechendes gilt für die 12-Stunden-Anzeige (**%hx**).

<b>%U0</b>	Keine Einblendung der Minuten
<b>%U1</b>	Minuten als möglichst kurze Zahl ohne führende Null einblenden
<b>%U2</b>	Minuten zweistellig, ggf. mit einer führenden Null einblenden
<b>%S0</b>	Keine Einblendung der Sekunden
<b>%S1</b>	Sekunden als möglichst kurze Zahl ohne führende Null einblenden
<b>%S2</b>	Sekunden zweistellig, ggf. mit einer führenden Null einblenden

Die Reihenfolge der Platzhalter ist zwar im Prinzip beliebig, jedoch wird in der Praxis nur die gewohnte Reihenfolge (Stunden, Minuten, Sekunden) auftreten.

Zwischen den Platzhaltern dürfen beliebige Zeichen oder Textfolgen stehen.

Folgende Beispiele zeigen die Darstellung der Uhrzeit 13:03:22 mit verschiedenen Zeitformaten:

<b>Format</b>	<b>Darstellung</b>
<code>%H1:%U2%S0</code>	13:03
<code>%H1:%U2:%S2</code>	13:03:22
<code>%H1 Uhr %U2%S0</code>	13 Uhr 03

## Verweis auf das System-Uhrzeitformat

Anstelle eines expliziten Formats kann der Formatstring auch einen Verweis auf das im System länderspezifisch vorgegebene System-Uhrzeitformat enthalten, welches in den Ländereinstellungen definiert werden kann. In diesem Fall besteht der gesamte Formatstring nur aus der Zeichenfolge:

**%SS** System-Uhrzeitformat

Sollte der Formatstring weitere Zeichen beinhalten, werden diese ignoriert.

## 7.3. Ganzzahlformat

Anders als die Datums- und Zeitformate - aber ähnlich dem Gleitkommaformat - bestehen Ganzzahlformate lediglich aus einer Reihe aneinandergereihter Steueranweisungen. Vor, zwischen und hinter den Steueranweisungen darf kein anderer Text stehen.

Alle Steueranweisungen beginnen mit einem Prozentzeichen. Dahinter folgt ein Buchstabe, der den Typ der Anweisung bestimmt. Folgende Steueranweisungen sind möglich:

<b>%Wx</b>	Definition der Stellenzahl
<b>%Tx</b>	Definition des Tausendertrenners
<b>%Bx</b>	Definition eines Prefix-Strings, der vor der Zahl eingeblendet wird
<b>%Ex</b>	Definition eines Postfix-Strings, der hinter der Zahl eingeblendet wird
<b>%+</b>	Anweisung, daß auch bei positiven Zahlen ein Vorzeichen erscheint
<b>%-</b>	Anweisung, daß bei negativen Zahlen das Vorzeichen unterdrückt wird

Je nach Typ der Anweisung folgt hinter dem Kennbuchstaben noch weitere Information.

Die Anweisung **%Wx** bestimmt die Anzahl Stellen, die die ganze Zahl haben soll. Die Stellenzahl folgt dem **%W** als einstellige Ziffer. Wenn die zu formatierende Zahl weniger Stellen hat, als gewünscht, werden führende Nullen erzeugt.

Die Anweisung **%Tx** bestimmt den Tausendertrenner, mit dem Tausendergruppen voneinander getrennt werden sollen. Die Angabe **x** kann aus einem oder mehreren Zeichen bestehen. Im Normalfall wird in Deutschland nur ein Zeichen verwendet, nämlich der Punkt. Wenn kein Tausendertrenner erwünscht ist, so wird für **x** einfach nichts angegeben, sondern direkt mit der nächsten Steueranweisung fortgefahren.

Die Anweisung **%Bx** (begin) und **%Ex** (end) definieren Zeichenfolgen **x**, die bis zum nächsten Prozentzeichen reichen. Diese Zeichenfolgen werden als Prefix bzw. Postfix vor bzw. hinter die formatierte Zahl gekettet. Wenn kein Prefix oder Postfix erwünscht ist, wird für **x** einfach nichts angegeben und direkt mit der nächsten Steueranweisung fortgefahren.

Die Anweisung **%+** bewirkt, daß positive Zahlen mit einem Plus-Zeichen dargestellt werden. Negative Zahlen werden ohnehin mit einem Minus-Zeichen versehen.

Folgende Beispiele zeigen die Darstellung der Zahl 1234 mit verschiedenen Ganzzahlformaten:

Format	Darstellung
%T.	1.234
%T	1234
%BDM %T. %EZaster	DM 1.234 Zaster
%+%W9%T,%EMio DM	+000,001,234 Mio DM

## 7.4. Gleitkommazahlformat

Anders als die Datums- und Zeitformate - aber ähnlich dem Ganzzahlformat - bestehen Gleitkommazahlformate lediglich aus einer Reihe aneinandergereihter Steueranweisungen. Vor, zwischen und hinter den Steueranweisungen darf kein anderer Text stehen.

Alle Steueranweisungen beginnen mit einem Prozentzeichen. Dahinter folgt ein Buchstabe, der den Typ der Anweisung bestimmt. Folgende Steueranweisungen sind möglich:

<b>%F<sub>x</sub></b>	Definition der Nachkommastellenzahl
<b>%T<sub>x</sub></b>	Definition des Tausendertrenners
<b>%D<sub>x</sub></b>	Definition des Dezimalkommata
<b>%B<sub>x</sub></b>	Definition eines Prefix-Strings, der vor der Zahl eingeblendet wird
<b>%E<sub>x</sub></b>	Definition eines Postfix-Strings, der hinter der Zahl eingeblendet wird
<b>%+</b>	Anweisung, daß bei positiven Zahlen ein Vorzeichen erscheint
<b>%-</b>	Anweisung, daß bei negativen Zahlen das Vorzeichen unterdrückt wird

Die Anweisung **%F<sub>x</sub>** bestimmt die Anzahl der Nachkommastellen, mit der die Zahl ausgegeben werden soll. Die Anzahl Nachkommastellen *x* wird als ein- oder zweistellige Zahl angegeben.

Die Anweisung **%T<sub>x</sub>** bestimmt den Tausendertrenner, mit dem Tausendergruppen voneinander getrennt werden sollen. Die Angabe *x* kann aus einem oder mehreren Zeichen bestehen. Im Normalfall wird in Deutschland nur ein Zeichen verwendet, nämlich der Punkt. Wenn kein Tausendertrenner erwünscht ist, so wird für *x* einfach nichts angegeben, sondern direkt mit der nächsten Steueranweisung fortgefahren.

Die Anweisung **%D<sub>x</sub>** bestimmt das Dezimalkomma. *x* besteht genau aus einem Zeichen, welches das Dezimalkomma repräsentiert. Im Normalfall wird in Deutschland das Komma verwendet. Wenn diese Angabe fehlt, wird ein Komma als Voreinstellung benutzt.

Die Anweisung **%B<sub>x</sub>** (begin) und **%E<sub>x</sub>** (end) definieren Zeichenfolgen *x*, die bis zum nächsten Prozentzeichen reichen. Diese Zeichenfolgen werden als Prefix bzw. Postfix vor bzw. hinter die formatierte Zahl gekettet. Wenn kein Prefix oder Postfix erwünscht ist, wird für *x* einfach nichts angegeben und direkt mit der nächsten Steueranweisung fortgefahren.

Die Anweisung **%+** bewirkt, daß positive Zahlen mit einem Plus-Zeichen dargestellt werden. Negative Zahlen werden ohnehin mit einem Minus-Zeichen versehen.

Folgende Beispiele zeigen die Darstellung der Zahl 12345,67 mit verschiedenen Gleitkommazahlformaten:

<b>Format</b>	<b>Darstellung</b>
%T.	12.345,67
%T.%F5	12.345,67000
%BDM %T%F0	DM 12346

## 8. Datenquellen für den Report

Der Waimea Report kann die Daten, die aufgelistet werden sollen, aus verschiedenen Datenquellen beziehen. Folgende Datenquellen werden unterstützt:

- ANSI-Dateien
- ASCII-Dateien
- Tex-Ass-Selektionsdateien
- DLL-Callback-Funktion
- ODBC-Datenbanken
- Simulierte Daten

### 8.1. ANSI-Dateien

Eine ANSI-Datei als Datenquelle für den Report muß die im folgenden beschriebene Struktur aufweisen. Die Datei muß zeilenweise organisiert sein. Jede Zeile der Datei enthält einen Datensatz. Die einzelnen Datenfelder innerhalb eines Datensatzes werden durch einen Tabulator (hex 09) voneinander getrennt. Die erste Zeile der Datei enthält die Datenfeldnamen. Auch diese sind durch Tabulatoren voneinander getrennt. Alle weiteren Zeilen der Datei enthalten Datensätze.

### 8.2. ASCII-Dateien

Eine ASCII-Datei als Datenquelle für den Report hat dieselbe Struktur, wie eine ANSI-Datei. Es wird lediglich der ASCII-Zeichensatz anstelle des ANSI-Zeichensatzes zugrundegelegt.

### 8.3. Tex-Ass-Selektionsdateien

Neben den oben beschriebenen Dateiformaten, kann der Report das proprietäre Dateiformat der Selektionsdateien des Produktes Tex-Ass for Windows direkt einlesen und verarbeiten. Das Format ist herstellerseitig beschrieben.

### 8.4. ODBC-Datenbanken

Der Report kann auf jede Datenbank, die mit einem ODBC-Treiber unter Windows installiert ist, direkt zugreifen.

### 8.5. DLL-Callback-Funktion (datensatzweise)

Für die Einbindung in DLL-strukturierte Programmpakete kann der Report die Datensätze, die gedruckt werden sollen, durch Aufrufen einer Callback-Funktion in einer beigestellten DLL lesen. Der Name der DLL wird als Datenquelle konfiguriert. In der DLL muß sich eine Funktion mit dem Namen **DataSourceAccess** befinden. Diese hat folgende Parameter:

***DataSourceAccess (long hwndRep, char \*Modus, char \*Buffer, long LngBuffer)***

Durch die Parameter **Buffer** und **LngBuffer** überreicht der Report der Funktion die Länge und Adresse eines Speicherbereiches, in den die Funktion Informationen übertragen soll. Die Art der Informationen wird durch den Parameter **Modus** definiert:

- "Names" Die Funktion liefert die Namen der von der DLL unterstützten Datenfelder zurück. Die Namen werden, durch Tabulatoren getrennt, miteinander verkettet. Die fertige Namenskette wird in den vom Report bereitgestellten Puffer **Buffer** kopiert. Die maximale Länge des Puffers darf nicht überschritten werden.
- "Open" Mit diesem Aufruf fordert der Report den ersten Datensatz an. Die Datenfelder müssen von der Funktion, durch Tabulatoren getrennt, in den bereitgestellten Puffer geschrieben werden. Die maximale Länge des Puffers darf nicht überschritten werden. Der Aufruf ist als Initialaufruf gedacht. Er erfolgt pro Ausdruck nur einmal und gibt der DLL-Funktion Gelegenheit, Initialisierungen vorzunehmen. Die Funktion liefert den Wert 0 zurück, wenn der Datensatz korrekt bereitgestellt ist. Wenn keine Datensätze verfügbar sind, muß der Wert -1 zurückgeliefert werden.
- "Next" Mit diesem Aufruf fordert der Report den nächsten Datensatz an. Die Datenfelder müssen von der Funktion, durch Tabulatoren getrennt, in den bereitgestellten Puffer geschrieben werden. Die maximale Länge des Puffers darf nicht überschritten werden. Der Aufruf erfolgt nicht, bevor nicht ein Aufruf "Open" stattgefunden hat. Die Funktion liefert den Wert 0 zurück, wenn ein weiterer Datensatz korrekt bereitgestellt wurde. Wenn keine Datensätze mehr verfügbar sind (end of file), wird der Wert -1 zurückgeliefert.
- "BufferSize" Die Funktion liefert die Größe des Puffers zurück, der von der DLL angelegt wird, um den Datentransfer zwischen DLL und Report abzuwickeln. Wenn die Funktion einen Wert zurückliefert, der kleiner als 30000 ist, wird ein Puffer mit einer Mindestlänge von 30000 Zeichen angelegt. Der hier zurückgelieferte Wert wird in allen anderen Funktionsaufrufen als Parameter **LngBuffer** übergeben.
- "ProgInd" Die Funktion liefert einen Wert für den aktuellen Zustand des **Progress-Indicators** zurück. Der Wert muß zwischen 0 und 1000 liegen.

## 8.6. DLL-Callback-Funktion (feldweise)

Alternativ zur obengenannten Schnittstelle zum Lesen ganzer Datensätze aus einer DLL wird auch eine Schnittstelle für das feldweise Lesen der Daten mittels einer DLL-Schnittstelle angeboten. Die Benutzung dieser Schnittstelle hat durch weg eine höhere Performance, als die satzweise Leseschnittstelle.

Der Name der DLL wird als Datenquelle konfiguriert. In der DLL muß sich eine Funktion mit dem Namen **DataSourceAccess** befinden. Diese hat folgende Parameter:

***DataSourceAccess (long hwndRep, char \*Modus, char \*FieldName, char \*Buffer, long LngBuffer)***

Durch die Parameter **Buffer** und **LngBuffer** überreicht der Report der Funktion die Länge und Adresse eines Speicherbereiches, in den die Funktion Informationen übertragen soll. Die Art der Informationen wird durch den Parameter **Modus** definiert:

- "Names" Die Funktion liefert die Namen aller von der DLL unterstützten Datenfelder zurück. Die Namen werden, durch Tabulatoren getrennt, miteinander verkettet. Die fertige Namenskette wird in den vom Report bereitgestellten Puffer **Buffer** kopiert. Die maximale Länge des Puffers darf nicht überschritten werden. Der Report fragt zuvor die Größe des Puffers bei der DLL mittels eines Aufrufs mit dem Modus *BufferSize* an. Der Parameter *FieldName* ist ohne Bedeutung.
- "Open" Mit diesem Aufruf fordert der Report den ersten Datensatz an. Die Datenfeldinhalte des ersten Datensatzes stehen nach diesem Aufruf zum Abruf durch Aufrufe mit dem Modus *GetField* bereit. Die Funktion liefert den Wert 0 zurück, wenn der Datensatz korrekt bereitgestellt ist. Der Parameter *FieldName* ist ohne Bedeutung. Wenn keine Datensätze verfügbar sind, muß der Wert -1 zurückgeliefert werden.
- "Next" Mit diesem Aufruf fordert der Report den nächsten Datensatz an. Die Datenfeldinhalte des nächsten Datensatzes stehen nach diesem Aufruf zum Abruf durch Aufrufe mit dem Modus *GetField* bereit. Der Aufruf erfolgt nicht, bevor nicht ein Aufruf "Open" stattgefunden hat. Die Funktion liefert den Wert 0 zurück, wenn ein weiterer Datensatz korrekt bereitgestellt wurde. Der Parameter *FieldName* ist ohne Bedeutung. Wenn keine Datensätze mehr verfügbar sind (end of file), wird der Wert -1 zurückgeliefert.
- "GetField" Mit diesem Aufruf fordert der Report den Inhalt des Feldes an, dessen Name auf Parameter *FieldName* übergeben wird. Der Aufruf dieser Funktion darf nicht stattfinden, bevor der Report den Datenzugriff mittels Modus *Open* eröffnet hat. Die Funktion liefert den Wert 0 zurück, wenn der Datenfeldinhalt korrekt bereitgestellt wurde. Wenn keine Datensätze mehr verfügbar sind (end of file), wird der Wert -1 zurückgeliefert.
- "BufferSize" Die Funktion liefert die Größe des Puffers zurück, der von der DLL angelegt wird, um den Datentransfer zwischen DLL und Report abzuwickeln. Wenn die Funktion einen Wert zurückliefert, der kleiner als 30000 ist, wird ein Puffer mit einer Mindestlänge von 30000 Zeichen angelegt.
- Wenn Parameter *FieldName* einen Namen übergibt, ist dies der Name des Feldes, für dessen Transfer der Puffer angelegt wird.
- Wenn *FieldName* NULL ist oder auf einen leeren Namen verweist, dann wird der Puffer für allgemeine Zwecke angefragt – z.B. für das Lesen der Feldnamen mittels Modus *Names*.
- "ProgInd" Die Funktion liefert einen Wert für den aktuellen Zustand des **Progress-Indicators** zurück. Der Wert muß zwischen 0 und 1000 liegen.

## 8.7. Simulation von Daten

Neben den oben beschriebenen Zugriffen auf *echte* Datenquellen bietet der Report die Möglichkeit, Daten zu simulieren. Die Anzahl der zu simulierenden Datensätze kann vom Benutzer vorgegeben werden. Die Simulation ist besonders nützlich, wenn ein Report-Layout erstellt wird und mittels der Druckseitenvorschau ein optischer Eindruck des Layouts gewonnen werden soll, ohne reale Daten zu verwenden.

## 8.8. Auswählen der Datenquelle

Art und Name der gewünschten Datenquelle können im Layout-Editor mittels eines Dialoges vom Benutzer eingestellt werden. Sie werden zusammen mit dem Reportformular gespeichert.

Darüberhinaus können Name und Art der Datenquelle auch mittels folgender Funktionen abgefragt oder gesetzt werden, ohne einen Dialog zu eröffnen:

- ***Rep\_GetDataSourceType***
- ***Rep\_GetDataSourceItem***
- ***Rep\_SetDataSourceType***
- ***Rep\_SetDataSourceItem.***

## 9. Betrieb des Reports als Druck-Server

In den vorangegangenen Kapiteln wurde eine Vielzahl von Interface-Routinen beschrieben, die den gesamten Leistungsumfang des Reports umfassen. Das gesamte Interface erlaubt es, sowohl den Report-Editor, als auch den Betrieb des Reports als reinen Druck-Server abzubilden.

Fast alle Funktionen des Interfaces dienen dem Betrieb als Editor. Für den Betrieb als reinen Druck-Server werden lediglich die folgenden Schnittstellen benötigt:

*Rep\_Preview\_File*  
*Rep\_Preview\_Report*  
*Rep\_Preview\_Report\_NonModal*  
*Rep\_Print\_File*  
*Rep\_Print\_Report*

Diese Funktionen gehören zu den wenigen Funktionen des Report-Interfaces, die aufgerufen werden, ohne den Handle eines Report-Fensters als Parameter zu übergeben. Sie werden in der Regel aufgerufen, ohne daß zuvor ein Report-Fenster eröffnet wurde.

Die Beschreibung der Funktionen erläutert, welche Parameter zur Verfügung gestellt werden können. Sowohl bei der *Druckvorschau* als auch beim *Druck* kann die Angabe einer Datenquelle weggelassen werden. In diesem Fall wird die im Reportformular eingestellte Datenquelle verwendet. Wenn jedoch eine Datenquelle angegeben wird, so wird diese **anstelle** der im Report eingestellten Datenquelle verwendet. Der Report selbst wird dabei nicht verändert.

## 10. Compilieren und Linken

Dieses Kapitel erläutert die notwendigen Komponenten für Compiler und Linker sowie die Bibliotheken und Headerdateien, die für eine Rahmenapplikation angegeben werden müssen, in die ein Report eingebunden werden soll.

### 10.1. Header-Dateien

Die Rahmenapplikation, in die der Report integriert wird, sollte in allen Quellen, die das Report-Interface verwenden, die Include-Dateien ***waimea.h*** und ***repdll.h*** (in dieser Reihenfolge) aufrufen. Diese beinhalten alle *Konstantendefinitionen*, alle *Strukturdefinitionen* und die *Prototypes* aller Funktionen des Report-Interfaces mit den entsprechenden Parameterdeklarationen.

Die Headerdateien werden bei der Installation im Unterverzeichnis ***develop*** des Installationsverzeichnis angelegt.

### 10.2. Linker

In der 32-Bit-Version muß die Bibliothek ***repdll32.lib*** in das Projekt integriert werden. Es handelt sich um eine implizite Bibliothek, die lediglich die Einsprünge in die Funktionen der dynamischen Bibliothek ***repdll32.dll*** koordiniert.

Die dynamische Bibliothek ***repdll32.dll*** sollten sich im Verzeichnis der Rahmenapplikation befinden.

## Anhang REPPROP      Report Properties

Die folgende Tabelle erläutert alle Properties, die für den Report mittels der Funktionen *Rep\_SetProperty* und *RepObj\_GetProperty* gesetzt oder ausgelesen werden können.

Die meisten Properties können sowohl ausgelesen als auch gesetzt werden. Manche hingegen können nur gelesen, andere wiederum nur gesetzt werden. Eine Property kann ein numerischer Wert oder ein String sein.

Manche Properties verändern Eigenschaften des Reports, die fest im Report gespeichert werden, andere wiederum beeinflussen nur Laufzeitparameter.

Die folgende Liste zeigt alle Properties, die definiert sind:

Property	Bedeutung
REP_PROP_ORDER_FIELDNAMES	<p>Legt fest, ob die Anzeige der Feldnamen, die mittels der Funktion <i>Rep_DeclareAvailableFields</i> beim Report als Liste verfügbarer Feldnamen angemeldet wurde, sortiert oder unsortiert erfolgt, wenn der Anwender auf einen der Buttons mit der Aufschrift <i>Datenfelder...</i> drückt.</p> <p>1 = sortiert 0 = unsortiert</p>
REP_PROP_HELPMODE	<p>Legt die Art und Weise fest, in der die Hilfe abgewickelt wird, wenn der Anwender in einem der zahlreichen Dialoge des Reports auf den Knopf <i>Hilfe...</i> drückt. Der Hilfe-Modus kann einen der folgenden Werte annehmen:</p> <p><b>REP_HELPMODE_DISABLE</b> Die Hilfe-Funktion ist abgeschaltet. Der Button <i>Hilfe</i> wird automatisch ausgeblendet</p> <p><b>REP_HELPMODE_STANDARD</b> Die Hilfe-Funktion ist im Standard-Betrieb. Wenn der Anwender auf <i>Hilfe</i> drückt, wird eine kontextorientierte Hilfe mittels des Windows-Hilfesystems angezeigt.</p> <p><b>REP_HELPMODE_MESSAGE</b> Anstelle der Anzeige einer Hilfe wird eine Notification-Message vom Type <i>REPMSG_HELP</i> oder <i>REPMSG_MACRO_HELP</i> an die Host-Anwendung gesendet. Diese überträgt die Adresse eines eindeutigen Kontext-Strings auf dem Parameter <i>IParam</i>. Genauer ist im Anhang NOTIF beschrieben.</p>

Property	Bedeutung
REP_PROP_PREVIEW_ZOOM	Gibt den Zoom-Faktor für die Vorschau an. Der Zoom-Faktor wird im Reportformular gespeichert und auch bei späteren Aufrufen der Druckvorschau als Voreinstellung benutzt.
REP_PROP_TRANSLATE_BRACKSEQS	Diese Eigenschaft steuert, ob in mehrzeiligen Texten die folgenden Sequenzen als Steuerzeichen interpretiert werden: <p>&lt;CR&gt; als Wagenrücklauf (hex 0D)  &lt;LF&gt; als Zeilenschaltung (hex 0A)  &lt;TAB&gt; als Tabulator (hex 09)</p> <p>1 = die Sequenzen werden interpretiert  0 = die Sequenzen werden nicht interpretiert und wie Text behandelt (Voreinstellung)</p>
REP_PROP_PAGENR_OFFSET	Legt den Offset zurück, der auf alle Seitennummern aufaddiert wird, damit Reports auch mit Seitennummern beginnen können die größer als 1 sind. Die Voreinstellung des Offsets ist 0. In diesem Fall werden die Seiten des Reports ab 1 nummeriert.
REP_PROP_DEFAULT_LANGUAGE	Legt die augenblicklich im Report eingestellte Standardsprache als Landeskenntung (z.B. 49 für deutsch) fest. Das ist die Sprache, in der alle Texte des Reports dargestellt werden, falls für die aktuell eingestellte Sprache keine spezifischen Texte verfügbar sind. Der Wert 0 bedeutet, daß für den Report keine Standardsprache eingestellt ist.
REP_PROP_NEGATIVE_OBJECT_COORDINATES	Legt fest, ob die Eigenschaft <i>negative Koordinaten für Objekte</i> aktiviert ist oder nicht. Wenn diese Eigenschaft aktiviert ist, dürfen Objekte auch oben oder links aus einer Zone herausragen, andernfalls nicht. <p>1 = negative Koordinaten erlaubt  2 = keine negativen Koordinaten erlaubt</p>
REP_PROP_NO_PROGRESS_INDICATORS	Legt fest, ob die Fortschrittsanzeige beim Drucken und bei der Vorschau unterdrückt wird: <p>1 = Fortschrittsanzeige unterdrücken  0 = Fortschrittsanzeige normal anzeigen</p>

Property	Bedeutung
REP_PROP_PRINT_HDC	<p>Mit dieser Eigenschaft kann ein Druck-Device-Kontext von außen vorgegeben bzw. auch wieder ausgelesen werden (siehe auch REP_PROP_PRINT_DEVMODE).</p> <p>Wenn sowohl ein DeviceMode als auch ein HDC vorgegeben sind, unterbleibt die Anfrage des Reports nach einem Drucker. Es wird dann auf jeden Fall der vorgegebene HDC verwendet.</p>
REP_PROP_PRINT_FROMPAGE	<p>Steuert den Wert <i>Von Seite</i> beim Drucken. Die Eigenschaft wird benutzt, um das entsprechende Control des Drucken-Dialoges vorzubesetzen. Wenn der Drucken-Dialog durch Setzen der Eigenschaft REP_PROP_PRINT_HDC abgeschaltet wurde, entscheidet diese Eigenschaft, welche Seite zuerst gedruckt wird. Der Wert -1 bedeutet, daß alle Seiten gedruckt werden. Dies ist auch die Voreinstellung des Wertes.</p>
REP_PROP_PRINT_TOPAGE	<p>Steuert den Wert <i>Bis Seite</i> beim Drucken. Die Eigenschaft wird verwendet, um das entsprechende Control des Drucken-Dialoges vorzubesetzen. Wenn der Drucken-Dialog durch Setzen der Eigenschaft REP_PROP_PRINT_HDC abgeschaltet wurde, entscheidet diese Eigenschaft bis zu welcher Seite gedruckt wird. Der Wert -1 bedeutet, daß alle Seiten gedruckt werden. Dies ist auch die Voreinstellung des Wertes.</p>
REP_PROP_PRINT_COPIES	<p>Steuert den Wert <i>Anzahl Kopien</i> beim Drucken. Diese Eigenschaft wird verwendet, um das entsprechende Control des Drucken-Dialoges vorzubesetzen. Wenn der Drucken-Dialog durch Setzen der Eigenschaft REP_PROP_PRINT_HDC abgeschaltet wurde, entscheidet diese Eigenschaft wieviele Kopien gedruckt werden. Die Voreinstellung ist 1.</p>
REP_PROP_PRINT_DEVMODE	<p>Mit dieser Eigenschaft kann ein DeviceMode für den Drucker von außen vorgegeben bzw. auch wieder ausgelesen werden (siehe auch REP_PROP_PRINT_HDC).</p> <p>Wenn sowohl ein DeviceMode als auch ein HDC vorgegeben sind, unterbleibt die Anfrage des Reports nach einem Drucker. Es wird dann auf jeden Fall der vorgegebene HDC und der verwendet.</p> <p>Die Eigenschaft ist ein Zeiger auf eine Devicemode-Struktur (siehe Windows-API: struct DEVMODE). Der Wert NULL bedeutet, daß kein externer DeviceMode verwendet wird (Voreinstellung).</p>

Property	Bedeutung
REP_PROP_ PREVIEW_WND_HEIGHT	Zuletzt verwendete Höhe des Previewfensters in Pixeln. -1 bedeutet, daß das Previewfenster maximiert war.
REP_PROP_ PREVIEW_WND_WIDTH	Zuletzt verwendete Höhe des Previewfensters in Pixeln. -1 bedeutet, daß das Previewfenster maximiert war.
REP_PROP_SUCC_PREVIEW_PRINT	Durch Abfragen dieser Eigenschaft kann festgestellt werden, ob während der Druckseitenvorschau die Funktion <i>Drucken</i> ausgeführt wurde und ob der Druck erfolgreich durchgeführt wurde:  1 = es wurde gedruckt 0 = es wurde nicht oder nicht erfolgreich gedruckt  Diese Eigenschaft kann nur abgefragt ,aber nicht gesetzt werden.
REP_PROP_CUSTOMPRINT	Diese Eigenschaft steht in Verbindung mit dem Report-Open-Flag <i>ROFLAG_CUSTOMPRINT_IN_PREVIEW</i> . Wurde der Report mit dieser Option eröffnet, so wirkt das Auslösen der Drucken-Schaltfläche wie die Schaltfläche <i>Schließen</i> , jedoch wird hierbei diese Eigenschaft auf 1 gesetzt, sodaß die Hostapplikation nach dem Schließen des Previewfensters diese Eigenschaft abfragen und ggf. selbst den Druck starten kann.
REP_PROP_ DISABLE_FIELDS_BUTTONS	Legt fest, ob der Button <i>Datenfelder...</i> der in verschiedenen Dialogen des Reports angeboten wird, um ein Datenfeld auszuwählen, aktiviert ist oder nicht:  1 = Der Button ist aktiv und kann benutzt werden 0 = Der Button ist grau und inaktiv
REP_PROP_ CURRENTLY_PRINTED_PAGE	Durch Abfragen dieser Eigenschaft kann während eines Reportlaufs die aktuell gedruckte Seite abgefragt werden.  Diese Information kann z.B. von Datenbereitstellungsmodulen verwendet werden, um die bereitgestellten Daten zu modifizieren.  Diese Eigenschaft kann nur abgefragt, aber nicht gesetzt werden.

Property	Bedeutung
REP_PROP_DISABLED_COMPONENTS	<p>Mit dieser Option können einzelne Komponenten der Report-Oberfläche ein- und ausgeschaltet werden. Sämtliche Funktionalität, sowie die Visualisierung dergleichen, wird aus dem Programm entfernt, als wäre diese Komponente nicht implementiert.</p> <p>Folgende Komponenten können z.Zt. ausgeschaltet werden:</p> <p>REP_COMPONENT_FIELD_REPLICATION Es wird die Funktionalität der automatischen Feldwiederholung ein- bzw. ausgeschaltet.</p>
REP_PROP_PAGE_HEIGHT	Diese Eigenschaft repräsentiert die Blatthöhe in Mikrometern.
REP_PROP_PAGE_WIDTH	Diese Eigenschaft repräsentiert die Blattbreite in Mikrometern.
REP_PROP_DISABLE_ORIENTATION_HANDLING	<p>Legt fest, ob der Report beim Druck die Blattausrichtung (Portrait oder Landscape) gemäß den Angaben im Reportformular einstellt, oder ob er dieses unterläßt und sich darauf verläßt, daß dies bereits von der Hostanwendung geschehen ist:</p> <p>0 = Report stellt alles selbst ein (Voreinstellung) 1 = Vorgaben erfolgen von außen im DEVMODE</p> <p>Siehe auch REP_PROP_PRINT_HDC und REP_PROP_PRINT_DEVMODE</p>
REP_PROP_DISABLE_BIN_HANDLING	<p>Legt fest, ob der Report beim Druck die Schachtsteuerung gemäß den Angaben im Reportformular übernimmt, oder ob er dieses unterläßt und sich darauf verläßt, daß dies von der Hostanwendung vorgenommen wird:</p> <p>0 = Report stellt alles selbst ein (Voreinstellung) 1 = Vorgaben erfolgen von außen im DEVMODE</p> <p>Siehe auch REP_PROP_PRINT_HDC und REP_PROP_PRINT_DEVMODE</p>
REP_PROP_DECIMAL_POINT	Legt den vom Report verwendeten Dezimalpunkt fest.
REP_PROP_COPYLIST_DESCRIPTOR	Diese String-Eigenschaft repräsentiert die gesamte Ausfertigungsliste des Reports. Es handelt sich um einen Deskriptorstring. Dieser ist an gesonderter Stelle dokumentiert.

Property	Bedeutung
REP_PROP_AUTO_OBJECTNAME_MODE	Legt fest, ob ...
REP_PROP_DISABLE_DOUBLE_CLICK	Legt fest, ob im Reporteditor der Doppelklick ausgewertet wird oder inaktiv ist:  1 = Doppelklick ist deaktiviert 2 = Doppelklick wird normal ausgeführt
REP_PROP_DISABLE_RIGHT_MOUSE_BUTTON	Legt fest, ob im Reporteditor der Rechtsklick ausgewertet wird oder inaktiv ist:  1 = Rechtsklick ist deaktiviert 2 = Rechtsklick wird normal ausgeführt
REP_PROP_FIELD_BUFFER_SIZE	Legt fest, mit welcher Puffergröße der Report intern pro Datenfeld arbeitet. Voreinstellung ist 30000 Zeichen.
REP_PROP_EXTERNAL_EDITED_OBJECTS	Diese Eigenschaft repräsentiert eine Liste von Objekttypen in form einer Bitliste. Wenn ein Bit gesetzt ist, bedeutet dies, daß das Bearbeiten der Eigenschaften des entsprechenden Objekts nicht durch den dafür vorgesehenen Dialog erfolgt, sondern von der Hostapplikation extern durchgeführt wird.  Wenn die Funktion <i>Objekt ändern...</i> für ein Objekt aufgerufen wird, für das dieses Bit gesetzt ist, wird anstelle den Dialog zu öffnen die Notification-Message REPMSG_PERFORM_EDIT_OBJECT an das Hostfenster gesendet.  Es können alle oder einzelne der bekannten Objekttypen angegeben werden. Die Liste aller Objekttypen ist im Anhang OBJ dargestellt.

Property	Bedeutung
<p>REP_PROP_NO_STARTDOC_AND_ENDDOC</p>	<p>Mit dieser Eigenschaft kann kontrolliert werden, ob der Report beim Drucken die Windows-API-Funktionen <i>StartDoc</i> und <i>EndDoc</i> aufruft oder nicht. Jeder Aufruf von <i>StartDoc</i> erzeugt einen neuen Druckauftrag.</p> <p>Durch Abschalten der Aufrufe kann der Report veranlaßt werden, seinen Druck in einen bereits eröffneten Druckauftrag einzubringen. In diesem Fall muß ein HDC beim Report angemeldet werden (siehe REP_PROP_PRINT_HDC), in dem die Funktion <i>StartDoc</i> bereits ausgeführt wurde.</p> <p>Auf diese Weise können mehrere Report-Drucke in einen Druckauftrag weingebettet werden.</p> <p>0 = <i>StartDoc</i> und <i>EndDoc</i> durchführen (Voreinstellung)                      1 = <i>StartDoc</i> und <i>EndDoc</i> nicht durchführen</p>
<p>REP_PROP_DEFAULT_PICTURE_PATH</p>	<p>Diese String-Eigenschaft legt den Pfad fest, der beim Ausgeben von Bildern benutzt wird, wenn als Bildquelle eine Datei angegeben wurde, diese aber ohne expliziten Pfad spezifiziert ist.</p> <p>Die Voreinstellung dieser Eigenschaft ist der Pfad, in dem sich das Reportformular befindet.</p>
<p>REP_PROP_DISABLE_OBJECT_CREATING</p>	<p>Diese Eigenschaft repräsentiert eine Liste von Objekttypen in form einer Bitliste. Wenn ein Bit gesetzt ist, bedeutet dies, daß Objekte dieses Typs nicht mehr angelegt werden können.</p> <p>Insbesondere werden die entsprechenden Funktionen aus den entsprechenden Kontext-Menüs ausgeblendet.</p> <p>Es können alle oder einzelne der bekannten Objekttypen angegeben werden. Die Liste aller Objekttypen ist im Anhang OBJ dargestellt.</p> <p>Wenn das Bit OBJTYPE_AUX gesetzt wird, werden auch die Funktionen zum Anlegen eines beliebigen Objekts deaktiviert.</p>

<b>Property</b>	<b>Bedeutung</b>
REP_PROP_ DISABLED_OPERATIONS	<p>Diese Eigenschaft repräsentiert eine Liste von Operationen, die im Report deaktiviert werden. Wenn ein Bit gesetzt ist, bedeutet dies, daß die entsprechende Operation nicht mehr ausgeführt werden kann.</p> <p>Insbesondere werden die entsprechenden Funktionen aus den betroffenen Kontext-Menüs ausgeblendet.</p> <p>Folgende Operationen können derzeit deaktiviert werden:</p> <p>REP_DISABLE_CREATE_ZONE Es können keine neuen Zonen angelegt werden</p>

## Anhang LOBJPROP      Long Properties von Objekten

Die folgende Tabelle erläutert alle Long-Properties von Reportobjekten, die mittels der Funktionen *RepObj\_SetLongProperty* und *RepObj\_GetLongProperty* gesetzt oder ausgelesen werden können.

Nicht jede Property kann auf jedes Objekt angewendet werden. Es ergibt sich aus dem Sinnzusammenhang, welche Properties für die einzelnen Objekttypen relevant sind.

Diejenigen Properties, die spezifisch für bestimmte Objekttypen definiert sind, werden durch den Beschreibungstext als solche ausgewiesen.

Property	Bedeutung
REPOBJ_PROP_PRINTSTATE	<p>Status, der angibt unter welchen Bedingungen das Objekt gedruckt wird. Folgende Werte sind möglich:</p> <p>REP_PRINTSTATE_NEVER Das Objekt wird niemals gedruckt</p> <p>REP_PRINTSTATE_EVER Das Objekt wird auf jeden Fall gedruckt</p> <p>REP_PRINTSTATE_IF_REF_NOT_EMPTY Das Objekt wird nur gedruckt, wenn das Referenzobjekt, dessen Name mittels der String-Property REP_PROP_PRINTSTATE_REFOBJ eigestellt werden kann, nicht leer ist.</p> <p>REP_PRINTSTATE_IF_REF_EMPTY Das Objekt wird nur gedruckt, wenn das Referenzobjekt, dessen Name mittels der String-Property REP_PROP_PRINTSTATE_REFOBJ eigestellt werden kann, leer ist.</p>
REPOBJ_PROP_OUTLINE_LEFT	Linker Rand des Objekts in Mikrometern
REPOBJ_PROP_OUTLINE_TOP	Oberer Rand des Objekts in Mikrometern
REPOBJ_PROP_OUTLINE_RIGHT	Rechter Rand des Objekts in Mikrometern
REPOBJ_PROP_OUTLINE_BOTTOM	Unterer Rand des Objekts in Mikrometern

Property	Bedeutung
REOBJ_PROP_LEFTPOS_TYPE	<p>Bezug der linken Position. Folgende Werte sind möglich:</p> <p>POSTYPE_FIX            Position relativ zum Rand</p> <p>POSTYPE_OBJREL        Position relativ zum rechten Rand von einem Referenzobjekt, welches mittels der String-Property REOBJ_PROP_LEFTPOS_REF eingestellt werden kann.</p>
REOBJ_PROP_TOPPOS_TYPE	<p>Bezug der oberen Position. Folgende Werte sind möglich:</p> <p>POSTYPE_FIX            Position relativ zur Zone</p> <p>POSTYPE_OBJREL        Position relativ zum unteren Rand von einem Referenzobjekt, welches mittels der String-Property REOBJ_PROP_TOPPOS_REF eingestellt werden kann.</p>
REOBJ_PROP_OBJECT_TYPE	<p>Typ des Objekts. Die Liste aller möglichen Objekttypen ist im Anhang OBJ beschrieben.</p>
REOBJ_PROP_STATE_BITS	<p>Statusbits des Objekts. Es können eines oder mehrere Bits gleichzeitig gesetzt sein, sofern sich die Bedeutungen der Bits nicht widerspricht. Folgende Statusbits sind definiert:</p> <p>OBSTATE_FIXPOS            Position ist fixiert  OBSTATE_FIXSIZE           Größe ist fixiert  OBSTATE_MUSTBESQUARE    Outline ist quadratisch  OBSTATE_FIXRATIO         Outline hat festes Seitenverhältnis  OBSTATE_QUAD13            Positive Steigung (nur für Linien)  OBSTATE_TRANSPARENT     Hintergrund ist transparent  OBSTATE_MULTILINE        mehrzeilige texte möglich  OBSTATE_CAPSLOCK         Großbuchstaben  OBSTATE_SUPRSEPANTE     Trenner vor Leerfeldern weglassen  OBSTATE_SUPRSEPPOST     Trenner nach Leerfld. weglassen  OBSTATE_OPEN_TOP         Rand oben offen  OBSTATE_OPEN_BOTTOM     Rand unten offen  OBSTATE_OPEN_LEFT        Rand links offen  OBSTATE_OPEN_RIGHT       Rand rechts offen  OBSTATE_SEPARATION       Objekt hat eine Einteilung  OBSTATE_AUTO_ENLARGE_VERT    automatisch vertikal vergrößern  OBSTATE_AUTO_SHRINK_VERT    automatisch vertikal verkleinern  OBSTATE_AUTO_ENLARGE_HORI    automaisch horizontal vergrößern  OBSTATE_AUTO_SHRINK_HORI    automatisch horizontal verkleinern</p> <p>Die Statusbits sind im Anhang STB nochmals separat zusammengefaßt.</p>

Property	Bedeutung
REPOBJ_PROP_VERT_ALIGN	Vertikale Ausrichtung des Objektinhalts innerhalb der Objektumrahmung:  ALIGN_TOP            oben ALIGN_VERTCENTER    mittig ALIGN_BOTTOM        unten
REPOBJ_PROP_HORI_ALIGN	Horizontale Ausrichtung des Objektinhalts innerhalb der Objektumrahmung:  ALIGN_LEFT            links ALIGN_HORICENTER     zentriert ALIGN_RIGHT           rechts ALIGN_DECIMALTAB    am Dezimaltabulator
REPOBJ_PROP_DATA_TYPE	Datentyp des Objekts (nur für Datenfelder):  DATATYPE_TEXT        Text DATATYPE_INT          Ganzzahlen DATATYPE_FLOAT       Gleitkomazahlen DATATYPE_DATE        Datum DATATYPE_TIME        Uhrzeit
REPOBJ_PROP_FRAME_TYPE	Randtyp des Objekts. Bei Linienobjekten definiert der Randtyp die Linienform. Folgende Typen sind definiert:  FRAMETYPE_NONE      kein Rand FRAMETYPE_DOT       gepunkteter Rand FRAMETYPE_SOLID     durchgehende Linie
REPOBJ_PROP_COLOR_FRAME	Farbe des Objektrandes. Wenn das Objekt den Randtyp FRAMETYPE_NONE hat, ist dieser Wert ohne Bedeutung.
REPOBJ_PROP_INNER_FRAME	Abstand des Objektinneren vom Rand in Mikrometern. Diese Angabe bestimmt z.B. wieviel Platz zwischen dem Rand und der Schrift innerhalb des Objekts gelassen wird.
REPOBJ_PROP_OUTER_FRAME	Dicke des Objektrands in Mikrometern. Dieser Wert ist nur dann von Bedeutung, wenn der Rahmentyp nicht den Wert FRAMETYPE_NONE hat.
REPOBJ_PROP_DECTAB_DISTANCE	Position des Dezimaltabulators innerhalb eines Objekts in Mikrometern. Der Dezimaltabulator ist nur für Datenfelder relevant, für die die horizontale Ausrichtung auf den Wert ALIGN_DECIMALTAB eingestellt ist.

Property	Bedeutung																																
REPOBJ_PROP_COLOR_BGR	Hintergrundfarbe des des Objekts. Dieser Wert ist ohne Bedeutung, wenn das Objekt das Statusbit OBSTATE_TRANSPARENT hat. Dann ist der Hintergrund transparent.																																
REPOBJ_PROP_COLOR_PATTERN	Farbe der Schraffur des Objekthintergrunds. Der Wert ist nur von Bedeutung, wenn die Property REPOBJ_PROP_PATTERN_BGR nicht den Wert REP_PATTERN_NONE hat.																																
REPOBJ_PROP_PATTERN_BGR	<p>Typ der Schraffur des Hintergrunds. Folgende Werte sind möglich:</p> <table border="0" data-bbox="810 882 1477 1346"> <tr> <td>REP_PATTERN_NONE</td> <td>keine Schraffur</td> </tr> <tr> <td>REP_PATTERN_BDIAGONAL</td> <td>diagonal ///</td> </tr> <tr> <td>REP_PATTERN_FDIAGONAL</td> <td>diagonal \\</td> </tr> <tr> <td>REP_PATTERN_CROSS</td> <td>senkrecht und waagrecht</td> </tr> <tr> <td>REP_PATTERN_DIAGCROSS</td> <td>diagonal in beiden Richtungen</td> </tr> <tr> <td>REP_PATTERN_HORIZONTAL</td> <td>waagrecht</td> </tr> <tr> <td>REP_PATTERN_VERTICAL</td> <td>senkrecht</td> </tr> <tr> <td>REP_PATTERN_GRAY_75_0</td> <td></td> </tr> <tr> <td>REP_PATTERN_GRAY_50_0</td> <td></td> </tr> <tr> <td>REP_PATTERN_GRAY_25_0</td> <td></td> </tr> <tr> <td>REP_PATTERN_GRAY_12_5</td> <td></td> </tr> <tr> <td>REP_PATTERN_GRAY_6_25</td> <td></td> </tr> <tr> <td>REP_PATTERN_GRAY_3_12</td> <td></td> </tr> <tr> <td>REP_PATTERN_KONFETTI</td> <td></td> </tr> <tr> <td>REP_PATTERN_GEFLECHT</td> <td></td> </tr> <tr> <td>REP_PATTERN_KUGELN</td> <td></td> </tr> </table>	REP_PATTERN_NONE	keine Schraffur	REP_PATTERN_BDIAGONAL	diagonal ///	REP_PATTERN_FDIAGONAL	diagonal \\	REP_PATTERN_CROSS	senkrecht und waagrecht	REP_PATTERN_DIAGCROSS	diagonal in beiden Richtungen	REP_PATTERN_HORIZONTAL	waagrecht	REP_PATTERN_VERTICAL	senkrecht	REP_PATTERN_GRAY_75_0		REP_PATTERN_GRAY_50_0		REP_PATTERN_GRAY_25_0		REP_PATTERN_GRAY_12_5		REP_PATTERN_GRAY_6_25		REP_PATTERN_GRAY_3_12		REP_PATTERN_KONFETTI		REP_PATTERN_GEFLECHT		REP_PATTERN_KUGELN	
REP_PATTERN_NONE	keine Schraffur																																
REP_PATTERN_BDIAGONAL	diagonal ///																																
REP_PATTERN_FDIAGONAL	diagonal \\																																
REP_PATTERN_CROSS	senkrecht und waagrecht																																
REP_PATTERN_DIAGCROSS	diagonal in beiden Richtungen																																
REP_PATTERN_HORIZONTAL	waagrecht																																
REP_PATTERN_VERTICAL	senkrecht																																
REP_PATTERN_GRAY_75_0																																	
REP_PATTERN_GRAY_50_0																																	
REP_PATTERN_GRAY_25_0																																	
REP_PATTERN_GRAY_12_5																																	
REP_PATTERN_GRAY_6_25																																	
REP_PATTERN_GRAY_3_12																																	
REP_PATTERN_KONFETTI																																	
REP_PATTERN_GEFLECHT																																	
REP_PATTERN_KUGELN																																	
REPOBJ_PROP_RECT_STYLE	<p>Diese Eigenschaft ist nur für Rechtecke relevant. Sie definiert den Stil des Rechtecks. Folgende Werte sind möglich:</p> <table border="0" data-bbox="810 1476 1417 1570"> <tr> <td>RECTSTYLE_FLAT</td> <td>flach</td> </tr> <tr> <td>RECTSTYLE_3DPOS</td> <td>dreidimensional erhaben</td> </tr> <tr> <td>RECTSTYLE_3DNEG</td> <td>dreidimensional vertieft</td> </tr> </table>	RECTSTYLE_FLAT	flach	RECTSTYLE_3DPOS	dreidimensional erhaben	RECTSTYLE_3DNEG	dreidimensional vertieft																										
RECTSTYLE_FLAT	flach																																
RECTSTYLE_3DPOS	dreidimensional erhaben																																
RECTSTYLE_3DNEG	dreidimensional vertieft																																
REPOBJ_PROP_RECT_CORNER	<p>Diese Eigenschaft ist nur für Rechtecke relevant. Sie definiert die Form der Ecken des Rechtecks. Folgende Werte sind möglich:</p> <table border="0" data-bbox="810 1733 1337 1827"> <tr> <td>CORNER_SHARP</td> <td>scharfe Ecken</td> </tr> <tr> <td>CORNER_MEDIUM</td> <td>leicht abgerundet</td> </tr> <tr> <td>CORNER_ROUND</td> <td>stark abgerundet</td> </tr> </table>	CORNER_SHARP	scharfe Ecken	CORNER_MEDIUM	leicht abgerundet	CORNER_ROUND	stark abgerundet																										
CORNER_SHARP	scharfe Ecken																																
CORNER_MEDIUM	leicht abgerundet																																
CORNER_ROUND	stark abgerundet																																

Property	Bedeutung								
REPOBJ_PROP_RECT_SPECIAL	<p>Diese Eigenschaft ist nur für Rechtecke relevant. Sie definiert die Enblendung von dreieckigen Pfeilspitzen an den Innenseiten des Rechtecks. Die Werte repräsentieren einzelne Bits einer Bitliste. Es können auch mehrere Werte gleichzeitig angegeben werden. Folgende Werte sind möglich:</p> <table data-bbox="810 663 1262 790"> <tr> <td>SPECIAL_LEFT</td> <td>Pfeil links</td> </tr> <tr> <td>SPECIAL_TOP</td> <td>Pfeil oben</td> </tr> <tr> <td>SPECIAL_RIGHT</td> <td>Pfeil rechts</td> </tr> <tr> <td>SPECIAL_BOTTOM</td> <td>Pfeil unten</td> </tr> </table>	SPECIAL_LEFT	Pfeil links	SPECIAL_TOP	Pfeil oben	SPECIAL_RIGHT	Pfeil rechts	SPECIAL_BOTTOM	Pfeil unten
SPECIAL_LEFT	Pfeil links								
SPECIAL_TOP	Pfeil oben								
SPECIAL_RIGHT	Pfeil rechts								
SPECIAL_BOTTOM	Pfeil unten								
REPOBJ_PROP_RECT_SHADOW	<p>Diese Eigenschaft ist nur für Rechtecke relevant. Sie definiert die Breite des Schattens bei 3D-Rechtecken in Mikrometern.</p>								
REPOBJ_PROP_PICT_ROP	<p>Diese Eigenschaft ist nur für Bilder relevant. Sie definiert die Art der Rasteroperation, mit dem das Bild dargestellt wird. Folgende Werte sind möglich:</p> <table data-bbox="810 1048 1315 1111"> <tr> <td>WAIMEA_ROP_COPY</td> <td>überdeckend</td> </tr> <tr> <td>WAIMEA_ROP_AND</td> <td>durchscheinend</td> </tr> </table>	WAIMEA_ROP_COPY	überdeckend	WAIMEA_ROP_AND	durchscheinend				
WAIMEA_ROP_COPY	überdeckend								
WAIMEA_ROP_AND	durchscheinend								
REPOBJ_PROP_PICT_STRECHMODE	<p>Diese Eigenschaft ist nur für Bilder relevant. Sie definiert die Art der Methode, Bilder zu Vergrößern oder zu Verkleinern. Folgende Werte sind möglich:</p> <p>WAIMEA_STRETCH_COLORONCOLOR Optimal für Bilder und Fotos</p> <p>WAIMEA_STRETCH_BLACKONWHITE Optimal für Strichzeichnungen, die dunkel auf hell sind</p> <p>WAIMEA_STRETCH_WHITEONBLACK Optimal für Strichzeichnungen, die hell auf dunkel sind</p> <p>WAIMEA_STRETCH_HALFTONE Optimal zur Darstellung farbtreuer Flächen</p>								

Property	Bedeutung
REPOBJ_PROP_PICT_SIZEMODE	<p>Diese Eigenschaft ist nur für Bilder relevant. Sie definiert, ob Bilder vergrößert oder verkleinert oder ob sie in Originalgröße angezeigt werden sollen. Folgende Werte sind möglich:</p> <p>WAIMEA_PICSIZE_AUTO_FIXRATIO Bild in Objekt einpassen, Seitenverhältnis wahren</p> <p>WAIMEA_PICSIZE_AUTO_VARRATIO Bild in Objekt einpassen, Seitenverhältnis nicht wahren</p> <p>WAIMEA_PICSIZE_ORIGINAL Bild in Originalgröße darstellen</p> <p>WAIMEA_PICSIZE_ZOOM Bild in Originalgröße, aber prozentual skaliert darstellen</p>
REPOBJ_PROP_PICT_WMFMODE	<p>Diese Eigenschaft ist nur für Bilder relevant. Sie definiert, ob WMF-Dateien im Vektorformat ausgegeben werden sollen, oder ob diese vor der Darstellung in ein Pixel-Format konvertiert werden sollen. Folgende Werte sind möglich:</p> <p>REP_WMF_VECTOR                   im Vektorformat ausgeben REP_WMF_PIXELS                   in Pixelformat konvertieren</p>
REPOBJ_PROP_PICT_MASK	<p>Diese Eigenschaft ist nur für Bilder relevant. Sie definiert, ob das Bild maskiert werden soll. Folgende Werte sind möglich:</p> <p>WAIMEA_PICMASK_NONE           nicht maskieren WAIMEA_PICMASK_ELLIPSE       mit Ellipse maskieren WAIMEA_PICMASK_CIRCLE       mein Kreis maskieren</p>
REPOBJ_PROP_PICT_ZOOM	<p>Diese Eigenschaft ist nur für Bilder relevant. Sie definiert den Vergrößerungsfaktor in Prozent, falls die Eigenschaft REPOBJ_PROP_PICT_SIZEMODE den Wert WAIMEA_PICSIZE_ZOOM hat.</p>
REPOBJ_PROP_PICT_SOURCE	<p>Diese Eigenschaft ist nur für Bilder relevant. Sie definiert die Bildquelle. Folgende Werte sind möglich:</p> <p>REP_PICTSOURCE_EMBEDDED   eingebettete Bild REP_PICTSOURCE_FIXFILE     externe Datei REP_PICTSOURCE_DATA       variabel durch Daten</p>
REPOBJ_PROP_FIELD_MAXLNG	<p>Diese Eigenschaft ist nur für Datenfelder relevant. Sie definiert die maximal auszugebende Anzahl Zeichen.</p>

Property	Bedeutung
REPOBJ_PROP_RECNR_RANGE	<p>Diese Eigenschaft ist nur für Datenstznummer-Objekte relevant. Sie definiert den Bereich auf den sich die Datensatznummern beziehen. Folgende Werte sind möglich:</p> <p>RANGE_REPORT            auf den gesamten Report  RANGE_PAGE                auf die aktuelle Seite  RANGE_GROUP              auf die aktuelle Gruppe</p>
REPOBJ_PROP_PAGNR_TYPE	<p>Diese Eigenschaft ist nur für Seitennummer-Objekte relevant. Sie definiert die Art der Seitennummer. Folgende Werte sind möglich:</p> <p>PAGNR_TYPE_CURRENT    Seitennummer  PAGNR_TYPE_NEXT        Seitennummer plus 1  PAGNR_TYPE_LAST        Anzahl der Gesamtseiten</p>
REPOBJ_PROP_SALDO_RANGE	<p>Diese Eigenschaft ist nur für Saldo-Objekte relevant. Sie definiert den Bezugsraum des Saldos. Folgende Werte sind möglich:</p> <p>SALDORANGE_GROUP      aktuelle Gruppe  SALDORANGE_PAGE        aktuelle Seite  SALDORANGE_REPORT      gesamter Report</p>

## Anhang SOBJPROP      String Properties von Objekten

Die folgende Tabelle erläutert alle String-Properties von Reportobjekten, die mittels der Funktionen *RepObj\_SetStringProperty* und *RepObj\_GetStringProperty* gesetzt oder ausgelesen werden können.

Nicht jede Property kann auf jedes Objekt angewendet werden. Es ergibt sich aus dem Sinnzusammenhang, welche Properties für die einzelnen Objekttypen relevant sind.

Diejenigen Properties, die spezifisch für bestimmte Objekttypen definiert sind, werden durch den Beschreibungstext als solche ausgewiesen.

Einige wenige Properties können nur gesetzt, nicht aber wieder ausgelesen werden. Auch auf diese Eigenschaft wird im Beschreibungstext hingewiesen.

Property	Bedeutung
REPOBJ_PROP_NAME	Name des Objekts
REPOBJ_PROP_DATAFIELD	Name des Datenfeldes der Datenquelle, mit dem das Objekt verknüpft ist.  Wenn das Objekt ein Verbundfeld ist, besteht diese Eigenschaft nicht aus einem einzelnen Datenfeld, sondern aus einer Datenfeldliste, in der alle Datenfelder des Verbundfeldes durch Tabulatoren getrennt aufgelistet sind.
REPOBJ_PROP_TEXT	Inhalt des Objekts. Diese Property ist nur für bestimmte Objekte sinnvoll - z.B. für Textobjekte
REPOBJ_PROP_FORMAT	Formatdeskriptor, der die Formatierung des Objektinhalts definiert. Die Art des Deskriptors hängt davon ab, was der Inhalt des Objekts für einen Datentyp hat. Die Syntax von Formatdeskriptoren wird in einem gleichnamigen Kapitel genau erläutert.
REPOBJ_PROP_PRINTSTATE_REF	Name des Bezugsobjekts, von dem das Drucken des Objekts abhängig gemacht wird, wenn die Long-Property REPOBJ_PROP_PRINTSTATE einen Wert hat, die sich auf ein anderes Objekt bezieht.
REPOBJ_PROP_LEFTPOS_REF	Name des Bezugsobjekts, auf das sich die linke Position des Objekts bezieht, wenn die Long-Property REPOBJ_PROP_LEFTPOS_TYPE den Wert POSTYPE_OBJREL hat (das Objekt also relativ zu einem anderen positioniert ist).

Property	Bedeutung
REOBJ_PROP_TOPPOS_REF	Name des Bezugsobjekts, auf das sich die obere Position des Objekts bezieht, wenn die Long-Property REOBJ_PROP_TOPPOS_TYPE den Wert POSTYPE_OBJREL hat (das Objekt also relativ zu einem anderen positioniert ist).
REOBJ_PROP_DECTAB_COMMA	Dezimalkomma, welches bei Dezimaltabulierung verwendet wird.
REOBJ_PROP_SEPARATION_DESC	Deskriptor, der die Strichunterteilung eines Objekts mit Textinhalt definiert. Der Deskriptor ist nur relevant, wenn das Objekt das Statusbit OBSTATE_SEPARATION hat. Andernfalls wird das Objekt nicht unterteilt.
REOBJ_PROP_FONT_FACENAME	Name der Schriftart des Objekts.
REOBJ_PROP_PICT_FILE	Diese Eigenschaft ist nur für Bildobjekte relevant. Sie definiert den Namen der Datei, in der sich das Bild befindet, wenn die Bildquelle vom Typ REP_PICTSOURCE_FIXFILE ist (siehe Long Property REOBJ_PROP_PICT_SOURCE).
REOBJ_PROP_PICT_EMBEDDED	Diese Eigenschaft ist nur für Bildobjekte relevant. Durch das Setzen dieser Eigenschaft wird bewirkt, daß das in einem Bildobjekt eingebettete Bild aus der Datei neu geladen wird, die als Eigenschaft übergeben wird.  Diese Eigenschaft kann zwar mittels der Routine <i>RepObj_SetStringProperty</i> gesetzt werden, kann aber nicht mittels der Routine <i>RepObj_GetStringProperty</i> wieder ausgelesen werden.
REOBJ_PROP_SALDO_RGFELD	Diese Eigenschaft ist nur für Saldo-Objekte relevant. Sie repräsentiert den Namen des Datenfeldes, auf das sich der Saldo bezieht.
REOBJ_PROP_COMPOUND_SEPS	Diese Eigenschaft ist nur für Verbundfelder relevant. Sie repräsentiert die Liste aller Texte, die als Trenner zwischen den Datenfeldern des Verbundfeldes stehen. Die einzelnen Trenntexte sind durch Tabulatoren voneinander separiert.

**Anhang FOBJPROP****Font Properties von Objekten**

Die folgende Tabelle erläutert alle Font-Properties von Reportobjekten, die mittels der Funktionen *RepObj\_SetFontProperty* und *RepObj\_GetFontProperty* gesetzt oder ausgelesen werden können.

Nicht jede Property kann auf jedes Objekt angewendet werden. Es ergibt sich aus dem Sinnzusammenhang, welche Properties für die einzelnen Objekttypen relevant sind.

<b>Property</b>	<b>Bedeutung</b>
REPOBJ_FONTPROP_HEIGHT	Höhe des Fonts in Zehntelpunkten.
REPOBJ_FONTPROP_ITALIC	Gibt an, ob der Font kursiv ist:  1 = kursiv 0 = nicht kursiv
REPOBJ_FONTPROP_UNDERLINE	Gibt an, ob der Font unterstrichen ist:  1 = unterstrichen 0 = nicht unterstrichen
REPOBJ_FONTPROP_STRIKEOUT	Gibt an, ob der Font durchgestrichen ist:  1 = durchgestrichen 0 = nicht durchgestrichen
REPOBJ_FONTPROP_WEIGHT	Gibt an ob der Font fett ist oder nicht:  FW_BOLD = fett FW_NORMAL = nicht fett
REPOBJ_FONTPROP_ESCAPEMENT	Drehung der Schrift gegen den Uhrzeiger sinn in 1/10 Grad
REPOBJ_FONTPROP_COLOR	Farbe der Schrift
REPOBJ_FONTPROP_CHARSET	Zeichensatzcodierung. Mögliche Werte sind:  ANSI_CHARSET

**Anhang OBJ      Objekttypen**

Folgende Objekttypen werden im Report verwendet. Der Typ eines jeden Objekts kann insbesondere mittels der Funktion *RepObj\_GetLongProperty* und Angabe der Eigenschaft *REPOBJ\_PROP\_OBJECT\_TYPE* abgefragt werden.

<b>Objekttyp</b>	<b>Bedeutung</b>
OBJTYPE_TEXT	Beschriftungstext
OBJTYPE_RECT	Rechteck
OBJTYPE_LINE	Line
OBJTYPE_CIRCLE	Kreis oder Ellipse
OBJTYPE_PICTURE	Bild
OBJTYPE_FIELD	Datenfeld
OBJTYPE_DATE	Einblendung des aktuellen Datums
OBJTYPE_TIME	Einblendung der aktuellen Uhrzeit
OBJTYPE_SALDO	Saldierung
OBJTYPE_RECORDNR	Einblendung der laufenden Datensatznummer
OBJTYPE_PAGENR	Einblendung der Seitennummer
OBJTYPE_COMPOUND	Verbundfeld
OBJTYPE_COPYNAME	Einblendung des Ausfertigungsnamens
OBJTYPE_ZONESEP	Zonenteiler

## Anhang STAT      Statusbits von Objekten

Jedes Objekt kann mit einer Reihe von Statusbits ausgerüstet werden. Die gesamte Bitliste kann durch die Long-Property REPOBJ\_PROP\_STATE\_BITS kontrolliert werden.

Die folgende Tabelle zeigt eine Übersicht aller öffentlichen Statusbits:

Statusbit	Bedeutung
OBSTATE_FIXPOS	Das Objekt ist fixiert. Seine Position kann nicht verändert werden
OBSTATE_FIXSIZE	Die Größe des Objekts ist fixiert und kann nicht verändert werden
OBSTATE_MUSTBESQUARE	Das Objekt muß quadratisch sein. Im Falle eines Rechtecks bedeutet dies, daß das Rechteck ein Quadrat ist. Im Falle eines Kreises bedeutet dies, daß das Objekt ein echter Kreis und keine Ellipse ist.
OBSTATE_FIXRATIO	Die Seitenverhältnisse des Objekts dürfen nicht verändert werden. Diese Eigenschaft bedeutet in Verbindung mit dem Statusbit OBSTATE_MUSTBESQUARE, daß das Objekt nie wieder seine quadratische Form verlieren kann.  Wenn dieses Statusbit gesetzt ist, werden beim Markieren des Objekts die oberen und seitlichen <i>Sizeboxen</i> weggelassen.
OBSTATE_QUAD13	Diese Eigenschaft ist nur für Linien relevant. Wenn dieses Bit gesetzt ist, verläuft die Linie vom ersten in den dritten Quadranten – also von links unten nach rechts oben. Ansonsten eben vom zweiten und vierten Quadranten.
OBSTATE_TRANSPARENT	Wenn dieses Bit gesetzt ist, wird das Objekt immer mit einem transparenten Hintergrund dargestellt. In diesem Fall ist die eingestellte Hintergrundfarbe und –Schraffur irrelevant.
OBSTATE_MULTILINE	Dieses Bit ist nur für Datenfelder relevant. Es besagt, daß das Feld mehrzeilig gefüllt und dargestellt wird, falls erforderlich.
OBSTATE_CAPSLOCK	Dieses Bit ist für alle Objekte relevant, in denen Text dargestellt wird. Es besagt, daß jeglicher Text in Großbuchstaben ausgegeben wird.
OBSTATE_SUPRSEPANTE	Dieses Bit ist nur für Verbundfelder relevant. Es bedeutet, daß der Trenner vor leeren Feldern weggelassen werden soll.

---

<b>Statusbit</b>	<b>Bedeutung</b>
OBSTATE_SUPRSEPOST	Dieses Bit ist nur für Verbundfelder relevant. Es bedeutet, daß der Trenner nach leeren Feldern weggelassen werden soll.
OBSTATE_OPEN_TOP	Wenn dieses Bit gesetzt ist, wird der obere Rand bei Objekten, die einen durchgehenden Rand haben, weggelassen.
OBSTATE_OPEN_BOTTOM	Wenn dieses Bit gesetzt ist, wird der untere Rand bei Objekten, die einen durchgehenden Rand haben, weggelassen.
OBSTATE_OPEN_LEFT	Wenn dieses Bit gesetzt ist, wird der linke Rand bei Objekten, die einen durchgehenden Rand haben, weggelassen.
OBSTATE_OPEN_RIGHT	Wenn dieses Bit gesetzt ist, wird der rechte Rand bei Objekten, die einen durchgehenden Rand haben, weggelassen.
OBSTATE_SEPARTION	Dieses Bit ist nur für Objekte relevant, in denen Text dargestellt wird. Wenn es gesetzt ist, wird der Text mit einer Strichteilung ausgegeben. Die Form der Strichteilung wird durch einen Teilungsdeskriptor vorgegeben.
OBSTATE_AUTO_ENLARGE_VERT	Dieses Bit ist nur für Datenfelder relevant, die das Statusbit OBSTATE_MULTILINE haben. Es besagt, daß das Datenfeld bei Bedarf in der vertikalen Richtung vergrößert wird.
OBSTATE_AUTO_SHRINK_VERT	Dieses Bit ist nur für Datenfelder relevant, die das Statusbit OBSTATE_MULTILINE haben. Es besagt, daß das Datenfeld bei Bedarf in der vertikalen Richtung verkleinert wird.

## Anhang NOTIF      Notification Messages

Nachdem eine Rahmenapplikation ein Report-Fenster mittels der Funktion **Rep\_Open** eröffnet hat, kann sie durch Aufrufen der Funktionen, die das Report-Interface zur Verfügung stellt, Einfluß auf das eröffnete Fenster und den darin befindlichen Report nehmen.

Umgekehrt informiert der Report die Rahmenapplikation und das Host-Fenster, als dessen Unterfenster der Report eröffnet ist, über Veränderungen und Zustände im Report, indem er Messages versendet.

Die meisten Messages sind rein informativen Characters. Die Rahmenapplikation kann diese Informationen auswerten oder ignorieren.

Alle Messages werden an das Host-Fenster versendet, welches in der Funktion **Rep\_Open** als Parameter angegeben wurde.

Die Message-Nummer wird von der Rahmenapplikation definiert und ebenfalls in der Funktion **Rep\_Open** als Parameter übergeben. In der Regel wird die Message in der Rahmenapplikation durch folgende Anweisung definiert:

```
#define WM_REPNOTIFY      (WM_USER + 1)
```

Alle Messages werden unter dieser Message-Nummer gesendet. Der Message-Parameter **wParam** unterscheidet die verschiedenen Ereignisse aus deren Anlaß die Message gesendet wurde. Die folgende Liste zeigt alle Messages und ihre Parameter:

### Messages an das Host-Fenster

Message	Bedeutung
REPMSG_CREATEREPORT	Es wurde ein neues Reportfenster eröffnet.
REPMSG_DESTROYREPORT	Das Reportfenster wird geschlossen.
REPMSG_MODIFYNEWOBJECT	Es wird gerade ein neues Objekt angelegt. Der Handle des Objekts wird in Parameter <b>lParam</b> übergeben. Das Objekt kann jetzt – bevor es aktiv wird – durch die Rahmenapplikation modifiziert werden.
REPMSG_EDITZONELIST	Die Zonenliste wurde verändert.
REPMSG_EDITZONE	Die aktuelle Zone wurde bearbeitet.
REPMSG_CURRENTZONE	Eine neue Zone wurde zur aktiven Zone.

Message	Bedeutung
REPMSG_ADDOBJECT	Es wurde ein neues Objekt angelegt.
REPMSG_DELETEOBJECT	Es wurde ein oder mehrere Objekte gelöscht.
REPMSG_EDITOBJECT	Die Kenndaten eines Objekts wurden verändert.
REPMSG_SELECTOBJECT	Es wurde ein Objekt markiert.
REPMSG_MOVEORSIZEOBJECT	Die Größe und/oder Position eines Objekts wurde verändert.
REPMSG_ZOOMCHANGE	Der Zoomfaktor wurde verändert.
REPMSG_GRIDCHANGE	Das Raster wurde verändert.
REPMSG_STARTMOVEMODE	Es wurde begonnen, ein Objekt mit der Maus zu bewegen. Die Message wird beim Drücken der Maustaste gesendet. <b>IParam</b> zeigt auf ein Rechteck vom Typ <b>LONGRECT</b> , welches die augenblickliche Position und Größe des Objekts enthält.
REPMSG_UPDATEMOVEMODE	Es wird eine neue Zwischenposition beim Verschieben eines Objekts mit der Maus gemeldet. Der Vorgang ist noch nicht abgeschlossen. Die Maustaste ist immer noch gedrückt. <b>IParam</b> zeigt auf ein Rechteck vom Typ <b>LONGRECT</b> , welches die augenblickliche Position und Größe des Objekts enthält.
REPMSG_ENDMOVEMODE	Das Bewegen eines Objekts mit der Maus wurde abgeschlossen. Die Message wird versendet, wenn die Maustaste losgelassen wird.
REPMSG_STARTSIZEMODE	Es wurde begonnen, ein Objekt mit der Maus zu vergrößern oder zu verkleinern. Die Message wird beim Drücken der Maustaste gesendet. <b>IParam</b> zeigt auf ein Rechteck vom Typ <b>LONGRECT</b> , welches die augenblickliche Position und Größe des Objekts enthält.
REPMSG_UPDATESIZEMODE	Es wird eine neue Zwischenposition beim Vergrößern oder Verkleinern eines Objekts mit der Maus gemeldet. Der Vorgang ist noch nicht abgeschlossen. Die Maustaste ist immer noch gedrückt. <b>IParam</b> zeigt auf ein Rechteck vom Typ <b>LONGRECT</b> , welches die augenblickliche Position und Größe des Objekts enthält.

Message	Bedeutung
REPMSG_ENDSIZEMODE	Das Vergrößern oder Verkleinern eines Objekts mit der Maus wurde abgeschlossen. Die Message wird versendet, wenn die Maustaste losgelassen wird.
REPMSG_EDITORDERCHANGE	Die Reihenfolge der Objekte einer Zone wurde verändert.
REPMSG_LBUTTONDOWN	Es wurde im Editierfenster die linke Maustaste gedrückt.
REPMSG_KEYDOWN	Es wurde im Editierfenster eine Taste gedrückt, welche nicht von der Report-DLL selbst verarbeitet wurde. Der Wert der Taste ( <i>siehe Windows-API: VK_...</i> ) wird auf <b>IParam</b> übergeben.
REPMSG_REGREQUEST_PRINT	Diese Message wird gesendet, wenn ein Report für die Bildschirmausgabe (Preview) oder die Druckausgabe erzeugt werden soll. Die Hostapplikation kann hier Registrierungsschecks durchführen und über den <b>Returnwert</b> determinieren, ob der Report generiert werden darf oder nicht. Folgende Returnwerte sind möglich: <ul style="list-style-type: none"> <li>-1 Druck/Preview wird nicht ausgeführt.</li> <li>0 Druck/Preview wird normal ausgeführt.</li> <li>1 Druck/Preview wird als Demo ausgeführt, es wird ein Wasserzeichen eingedruckt.</li> </ul>
REPMSG_REGREQUEST_HTMLEXPORT	Diese Message wird gesendet, wenn ein Report für den HTML-Export erzeugt werden soll. Die Hostapplikation kann hier Registrierungschecks durchführen und über den <b>Returnwert</b> determinieren, ob der Report generiert werden darf oder nicht. Folgende Returnwerte sind möglich: <ul style="list-style-type: none"> <li>-1 HTML-Export wird nicht ausgeführt.</li> <li>X HTML-Export wird normal ausgeführt.</li> <li>1 HTML-Export wird als Demo ausgeführt, es wird ein Wasserzeichen eingearbeitet.</li> </ul> Der Returnwert für den HTML-Export (X) kann bei der Firma <b>Waimea Software GmbH</b> erfragt werden.
REPMSG_HELP	Diese Message wird gesendet, wenn ein Anwender auf einen Hilfe-Knopf in einem der Dialoge des Reports drückt und der Hilfe-Modus zuvor mittels der Funktion <i>Rep_SetProperties</i> auf den Wert <i>REP_HELPMODE_MESSAGE</i> eingestellt worden ist. Auf dem Parameter <b>IParam</b> wird die Adresse eines eindeutigen Kontext-Strings übergeben. Insbesondere jedem Dialog, der einen Hilfe-Knopf enthält, ist ein Kontext-String zugeordnet.

---

<b>Message</b>	<b>Bedeutung</b>
REPMSG_MACRO_HELP	Diese Message wird gesendet, wenn ein Anwender in der Liste der verfügbaren Makrofunktionen auf den Hilfe-Knopf drückt und der Hilfe-Modus zuvor mittels der Funktion <i>Rep_SetProperties</i> auf den Wert <i>REP_HELPMODE_MESSAGE</i> eingestellt worden ist. Auf dem Parameter <i>IParam</i> wird ein Zeiger auf den Namen des Makrobefehls übergeben, für den Hilfe erwünscht ist.
REPMSG_PERFORM_EDIT_OBJECT	Diese Message wird gesendet, wenn die Funktion <i>Objekteigenschaften</i> aufgerufen wird und das Bearbeiten von Objekten nicht vom Reporteditor durchgeführt werden soll, sondern durch eine externe Routine. Diese Eigenschaft kann durch die ganzzahlige Eigenschaft <i>REP_PROP_EXTERNAL_EDITED_OBJECTS</i> ein- oder ausgeschaltet werden.

**Anhang OBSOLETE****Veraltete Funktionen**

In diesem Anhang werden die Namen aller Funktionen genannt, die zwar von der Datei *repdll32.dll* exportiert werden, von Fremdapplikationen jedoch nicht benutzt werden sollten, da ein Support dieser Funktionen in Zukunft nicht gewährleistet ist. Es handelt sich entweder um veraltete Funktionen oder um interne Funktionen, die jederzeit inkompatibel geändert werden können.

Alle folgenden Funktionen sind veraltet:

- PageBgr\_SetBitmap
- RepObj\_GetDataField
- RepObj\_GetDataType
- RepObj\_GetFontFaceName
- RepObj\_GetFontHeight
- RepObj\_GetFormat
- RepObj\_GetHoriAlign
- RepObj\_GetMaxLng
- RepObj\_GetName
- RepObj\_GetText
- RepObj\_GetType
- RepObj\_GetVertAlign
- RepObj\_SetAlign
- RepObj\_SetBitmap
- RepObj\_SetDataField
- RepObj\_SetDataType
- RepObj\_SetFontFaceName
- RepObj\_SetFormat
- RepObj\_SetMaxLng
- RepObj\_SetName
- RepObj\_SetText
- Rep\_AdjustEmptyDataFieldNames
- Rep\_GetBatchMode
- Rep\_GetLtsModuleName
- Rep\_SetBatchMode
- Rep\_SetPageBgrLoadBitmapCallback

Die Leistungen aller veralteten Funktionen werden durch modernere Funktionen abgewickelt. Insbesondere sollten alle Eigenschaften von Objekten durch Verwendung der neuen Funktionen:

- RepObj\_SetLongProperty
- RepObj\_GetLongProperty

- RepObj\_SetStringProperty
- RepObj\_GetStringProperty

- RepObj\_SetFontProperty
- RepObj\_GetFontProperty

kontrolliert werden.

**Anhang EXAMPLE****Quellcode-Beispiel einer Einbindung**

Das folgende Quellcode-Beispiel erzeugt eine Datei mit 100 Datensätzen und den Datenfeldern **Name**, **Vorname** und **Telefon**. Die Datensätze werden automatisch generiert. Anschließend wird diese Datei als Report mit dem Report-Formular **beisp.rep** ausgedruckt.

```
// Unterdrückung von Warnungen für den Microsoft Compiler
#ifdef WIN32
#pragma warning (disable:4699 4001 4201 4209 4115 4214 4514)
#else
#pragma warning (disable:4699)
#endif

#include <windows.h>
#include <stdio.h>
#include <share.h>

#include <waimea.h> // Allgemeine Makrodefinitionen
#include <repdll.h> // Report-DLL-Schnittstellen

void PrintReport (void)
{
    char *TempFile;
    char *DataSource;
    FILE *Fs;
    int k;

    // Dateinamen bereitstellen
    TempFile = "C:/TMP.TMP";

    // Anlegen der Datei
    if ((Fs = _fsopen (TempFile, "w", _SH_DENYNO)) == NULL)
        return;

    // Kopfzeile erzeugen
    fprintf (Fs, "%s\t%s\t%s\n", "Name", "Vorname", "Telefon");

    // Erzeuge alle Datenzeilen
    for (k=0; k<100; ++k)
        fprintf (Fs, "Name-%d\tVorname-%d\tTelefon-%d\n", k, k, k);

    // Datei schließen
    fclose (Fs);

    // Report drucken
    DataSource = "TEXTFILE,NAME=C:/TMP.TMP,CODE=0,COLSEP=9,FIELDNAMES=1";
    Rep_Print_File (GetActiveWindow (), "beisp.rep", DataSource, (HDC) 0, 0, 0, 1, "Profile.ini", 0);

    // Datei wieder löschen
    unlink (TempFile);
}
```

Der obige Quellcode wird bei der Installation in der Datei **ReportPrint.cpp** im Unterverzeichnis **Sources\Cpp\PrintModul** des Zielverzeichnisses angelegt.