

RSE-Author - Benutzerhandbuch

Diese Hilfedatei stellt ein elektronisches Online-Handbuch zu dem Autorensystem RSE-Author dar. Dieses Handbuch enthält alle Informationen, die Sie zur Arbeit mit dem Autorensystem benötigen.

Falls Sie mit RSE-Author noch nicht vertraut sind, empfehlen wir Ihnen, zuerst die Einleitung und danach das Kapitel 'Programmieren mit RSE-Author' durchzuarbeiten. Beachten Sie bitte auch den Abschnitt 'Nutzungsbedingungen'.

Innerhalb der Entwicklungsumgebung von RSE-Author können Sie jederzeit durch Drücken der Taste F1 eine kontextsensitive Hilfe aufrufen.

Allgemeines

[Einleitung](#)

[Nutzungsbedingungen](#)

[Programmieren mit RSE-Author](#)

Komponenten des Autorensystems

[Entwicklungsumgebung \(author.exe\)](#)

[Laufzeitsystem \(runtime.exe\)](#)

[Patch-Programm \(expatch.exe\)](#)

[Grafikbibliotheken erstellen \(grflib.exe\)](#)

[Passwortschutz](#)

[Virenschutz](#)

Script-Sprache

[Befehlsreferenz](#)

Sonstiges

[Häufig gestellte Fragen \(FAQ\)](#)

[Was muß in Verbindung mit einem RSE-Author-Programm kopiert werden?](#)

[Weiterentwicklung des Autorensystems](#)

Einleitung

Wir freuen uns, daß Sie sich entschlossen haben, mit dem Autorensystem RSE-Author zu arbeiten. RSE-Author ist ein modernes und bedienungsfreundliches Entwicklungssystem, das es Ihnen erlaubt, die Möglichkeiten Ihres Computers zur Erstellung von multimedialen Präsentationsprogrammen voll auszuschöpfen.

Copyright

Das Autorensystem RSE-Author, alle mit RSE-Author mitgelieferten Programme, Dateien und Dokumentationen, sind für die Firma Reitz Software-Entwicklung durch nationale und internationale Copyright-Vorschriften geschützt.

Sobald Sie mit RSE-Author zu arbeiten beginnen, erkennen Sie automatisch die nachfolgend aufgeführten Lizenzvereinbarungen und alle damit verbundenen Copyright-Vorschriften an. Nur unter gleichzeitiger Anerkennung dieser Fakten sind Sie berechtigt, RSE-Author zu nutzen.

Lizenzumfang

RSE-Author und alle dazu gehörenden Programme, Dateien und Dokumentationen dürfen nur auf der Anzahl von Computern eingesetzt werden, für die eine gültige Lizenz erworben wurde.

Erstellen von Kopien

Das Erstellen von Kopien von RSE-Author ist nur zu Sicherheitszwecken erlaubt. Die Erstellung von Kopien zu anderen Zwecken ist untersagt und kann strafrechtlich verfolgt werden.

Haftung

Die Firma Reitz Software-Entwicklung schließt jede Haftung für Schäden aus, die durch vorhandene Angaben in der Dokumentation entstehen.

Obwohl das Autorensystem RSE-Author sehr sorgfältig getestet wurde, kann eine Haftung für mittelbare oder unmittelbare Folgeschäden aus Programmfehlern nicht übernommen werden.

Updates

Registrierte Anwender können neue Versionen im Rahmen eines Update-Service zu einem reduzierten Preis beim Hersteller erwerben. Als Berechtigungsnachweis gilt die beiliegende Registrierungskarte.

Support

Wir bitten um Verständnis dafür, daß wir für ein Programm, das zu einem so günstigen Preis angeboten wird wie RSE-Author, keinen telefonischen oder schriftlichen Support anbieten können.

Wenn Ihnen ein Modem zur Verfügung steht, können Sie über die weiter unten angegebene Mailbox mit uns in Verbindung treten. Wir unterhalten in dieser Mailbox eine Support-Area. Hier können Sie Fragen zu RSE-Author an uns richten. Wir bitten aber auch hier darum, keine Fragen zu stellen, deren Beantwortung sich aus der Lektüre dieses Handbuchs ergibt.

Obwohl wir RSE-Author sehr gründlich getestet haben, ist es unmöglich, ein Programm in allen denkbaren Konfigurationen zu testen. Sollten Sie deshalb noch Fehler in RSE-Author entdecken, dann teilen Sie uns diese bitte schriftlich mit. Geben Sie dabei die folgenden Informationen an.

- Ihre Systemkonfiguration (Rechner, Speicherausbau, Betriebssystem, Grafikkarte usw.)
- Eine detaillierte Schilderung des Fehlers und unter welchen Umständen er aufgetreten ist.

Hersteller

**Reitz Software-Entwicklung
Langendellschlag 58**

65199 Wiesbaden

Mailbox

Wir unterhalten als unverbindliche Service-Leistung einen Support-Bereich in der Switchboard-BBS (06173/935935). Hier können Sie Fragen und Anregungen an uns richten.

Warenzeichen

IBM, VGA, OS/2 sind eingetragene Warenzeichen der International Business Machines Corp.

MS-DOS, MS-Windows sind eingetragene Warenzeichen der Microsoft Corp.

PC Paintbrush ist ein eingetragenes Warenzeichen der Z-Soft Corp.

Alle in diesem Handbuch zusätzlich verwendeten Programmnamen etc. sind ebenfalls eingetragene Warenzeichen und dürfen nicht gewerblich oder in sonstiger Weise verwendet werden.

Nutzungsbedingungen

Nutzungsbedingungen für das Autorensystem

RSE-Author und alle dazu gehörenden Programme, Dateien und Dokumentationen dürfen nur auf der Anzahl von Computern eingesetzt werden, für die eine gültige Lizenz erworben wurde.

Die Weitergabe der Dateien runtime.exe, author.fnt, sv221bw.dll und tl221bw.dll an Dritte ist erlaubt.

Alle anderen Programme, Dateien und Dokumentationen, die zu RSE-Author gehören, sind durch nationale und internationale Copyright-Vorschriften geschützt und dürfen nicht weitergegeben werden.

Nutzungsbedingungen für die Beispiel-Applikationen

Die im Lieferumfang von RSE-Author mitgelieferten Beispiel-Applikationen können als Vorlage für eigene Applikationen verwendet werden. Die Beispiel-Applikationen als solche dürfen nicht weitergegeben werden. Die zu den Beispiel-Applikationen gehörenden Grafiken, Videos und sonstige Dateien dürfen ebenfalls nicht weitergegeben werden.

Nutzungsbedingungen für das beiliegende Bildmaterial im Photo-CD-Format

Die Nutzung des digitalen Bildmaterials im Photo-CD-Format, das auf der RSE-Author-CD enthalten ist, ist unter den folgenden Bedingungen gestattet:

Das Bildmaterial darf für die Herstellung von gedruckten Publikationen verwendet werden. Das Bildmaterial darf des weiteren für nichtkommerzielle Anwendungen im privaten Bereich verwendet werden. Jede über die vorigen Punkte hinausgehende Verwertung des Bildmaterials bedarf der vorherigen schriftlichen Zustimmung des Herstellers.

Entwicklung von portablen Programmen

Sollen Programme sowohl unter Windows als auch unter OS/2 ablaufen, sollten einige Punkte beachtet werden.

Generell gilt, daß Anwendungen, die mit der Windows-Version von RSE-Author erstellt wurden, ohne Änderung direkt von der OS/2-Version verarbeitet werden können. Das gilt natürlich auch umgekehrt.

Die Praxis hat allerdings gezeigt, daß eine hundertprozentige Kompatibilität der Applikationen aus folgenden Gründen nicht erreicht werden kann.

1. Unter OS/2 stehen nicht die gleichen Schriften wie unter Windows zur Verfügung. Selbst wenn Windows-Schriften auf die jeweils geeignetste OS/2-Schrift gemappt werden, kommt es meistens zu unschönen Abweichungen im Schriftbild.
2. Unter OS/2 stehen nicht die gleichen Füllmuster wie unter Windows zur Verfügung. Einige Muster fehlen ganz, die anderen Muster unterscheiden sich in der Dichte von ihren Windows-Pendants.
3. Unter OS/2 verhalten sich Dialogfenster geringfügig anders als unter Windows. Wenn in ein Dialogfenster nachträglich Controls (Schalter, Listboxen, Fenster) eingetragen werden, kommt es zu Flackereffekten.
4. Erfahrungsgemäß laufen Windows-AVI-Dateien nicht unter OS/2 und umgekehrt.
5. RSE-Author unterstützt unter OS/2 in der vorliegenden Version keine Paletten. Beim Laden von Bitmaps werden die Farben an die OS/2-Standardpalette angepaßt.

Um ein Programm für beide Plattformen zu entwickeln, sollten deshalb die folgenden Punkte beachtet werden:

1. Die Entwicklung sollte gleichzeitig für beide Plattformen erfolgen. Bei der Programmierung sollte man darauf achten, Schriften und Schriftgrößen zu verwenden, die auf beiden Plattformen möglichst ähnlich sind. Positionierung und Umbruch der Texte sollten so gewählt werden, daß das Ergebnis auf beiden Plattformen optisch ansprechend ist.
2. Das gleiche gilt für die Verwendung der Füllmuster. Verwenden Sie nur Füllmuster, die auf beiden Plattformen verfügbar sind. Alternativ können Sie ein Bitmap als Hintergrund für ein Fenster einsetzen. In dem Fall sollte der Fensterhintergrund auf 'unsichtbar' geschaltet werden.
3. Dialogfenster sollten grundsätzlich unsichtbar eröffnet werden und erst nach dem vollständigen Zeichnen sichtbar geschaltet werden. Das Sichtbarschalten sollte im Open-Script (wird erst nach dem vollständigen Fensteraufbau aufgerufen) des Fensters mit dem Befehl SHOW erfolgen. Dialogfenster sollten des weiteren nur für statische Statusmeldungen oder Abfragen verwendet werden.
4. Wird mit digitalem Video gearbeitet, sollte geprüft werden, ob das verwendete Format auch von beiden Betriebssystemen unterstützt wird. Ggfs. müssen für beide Plattformen systemspezifische Dateien erzeugt werden.
5. Wenn Sie mit palettenbehafteten Bitmaps arbeiten, dann passen Sie diese am besten mit einem Bildbearbeitungsprogramm an die Standardpalette von OS/2 an. Die

Standardpalette von OS/2 bietet 256 normalverteilte Farben an. Durch geschicktes Dithering läßt sich damit eine akzeptable Qualität erreichen.

Entwicklungsumgebung (author.exe)

Das Programm 'author.exe' ist das zentrale Programm des Autorensystems. Es enthält alle Komponenten, die zum Entwickeln, Testen und Ausführen von Programmen erforderlich sind.

Die Entwicklungsumgebung besteht aus den folgenden Teilen:

1. Editor (Bearbeiten und Editieren der Programme)
2. Hilfesystem (Online-Handbuch)
3. Laufzeitsystem (Ausführen des fertigen Programms)

Bedienung der Entwicklungsumgebung

Alle Teile der integrierten Entwicklungsumgebung können über eine komfortable Menüleiste bedient werden und zwar wahlweise mit der Tastatur oder mit der Maus. Im folgenden finden Sie eine Auflistung aller Funktionen der Menüleiste:

Datei

Neu
Öffnen
Speichern
Speichern unter
Drucken
Druckerinstallation
Beenden

Bearbeiten

Rückgängig
Ausschneiden
Kopieren
Einfügen
Löschen
Elemente nach vorne
Elemente nach hinten

Objekt

Seite
Fenster
Dokument
Einlagern

Element

Linie
Rechteck
Kreis
Grafik
Symbol
Text
Rahmen
Schalter
Feld

Listbox
Fenster

Optionen

Programm starten
Passwort bearbeiten
Gitter einstellen
Einstellungen speichern

Fenster

Horizontal ordnen
Vertikal ordnen
Überlappend ordnen
Symbole ordnen
Alle schließen

Hilfe

Index
Hilfe verwenden
Info
System

Laufzeitsystem (runtime.exe)

Nachdem ein Autor eine Applikation mit der integrierten Entwicklungsumgebung (author.exe) erstellt hat, möchte er natürlich sein Werk an Dritte weitergeben. Außerdem soll ein Programm ausführbar sein, ohne beim Programmstart jedesmal die Entwicklungsumgebung aktivieren zu müssen.

Die erzeugten Programme mit der Dateikennung '.PRG' können nicht selbständig ausgeführt werden. Die Programme benötigen ein sogenanntes 'Laufzeitsystem' zur Ausführung. Dieses Laufzeitsystem ist ein eigenständiges Programm namens 'runtime.exe'. Dieses Programm wird von der Entwicklungsumgebung beim Start eines Programms automatisch aufgerufen. Es kann aber auch unabhängig von der Entwicklungsumgebung verwendet werden.

Starten des Laufzeitsystems

Wird das Icon des Laufzeitsystems angeklickt, dann wird ein Abfragefenster aufgeblendet, in dem der Name des zu startenden Programms eingegeben werden kann.

Dem Laufzeitsystem kann als Parameter der Name des zu startenden Programms mitgegeben werden.

Beispiel: runtime demo

Soll ein Programm beim Anklicken des Icons des Laufzeitsystems automatisch gestartet werden, dann muß das Programm mit dem Laufzeitsystem im Dateimanager mittels der Funktion 'Verknüpfen' verbunden werden.

Zusätzlich besteht noch die Möglichkeit mit dem Programm 'expatch.exe' den Namen eines zu startenden Programms direkt in den Programmcode des Laufzeitsystems zu übertragen.

Patch-Programm (expatch.exe)

Mit diesem Tool kann der Dateiname eines RSE-Author-Programms direkt in den Programmcode des Laufzeitsystems übertragen werden. Beim Start des Laufzeitsystems wird dieses Programm dann automatisch gestartet. Das ist z.B. sinnvoll, wenn Applikationen ohne Installation direkt von einer CD-ROM ausführbar sein sollen.

Das Programm 'expatch.exe' muß unter DOS mit den folgenden Parametern aufgerufen werden:

expatch dateiname /programm [?]

dateiname

Name des RSE-Author-Programms, das in das Laufzeitsystem eingetragen werden soll.

/programm

Name des Laufzeitsystems.

?

Hilfetext anzeigen

Beispiel

1. **expatch myprog.prg /runtime.exe**

Der Dateiname 'myprog.prg' wird als zu startende Applikation in den Programmcode von 'runtime.exe' eingetragen. Das Laufzeitsystem und das auszuführende Programm müssen im gleichen Verzeichnis stehen.

2. **copy runtime.exe myprog.exe expatch myprog.prg /myprog.exe**

Das Laufzeitsystem wird umbenannt. Anschließend wird 'myprog.prg' als zu startende Applikation in das umbenannte Runtime eingetragen. Das Laufzeitsystem und das auszuführende Programm müssen im gleichen Verzeichnis stehen.

3. **expatch /runtime**

Der Dateiname der zu startenden Applikation wird aus 'runtime.exe' wieder entfernt.

Grafikbibliotheken erstellen (grflib.exe)

Dieses Programm faßt Bitmap-Grafiken (*.BMP) in einer sogenannten Grafikbibliothek zusammen. Eine solche Bibliothek ist leichter zu handhaben als mehrere dutzend oder hundert Einzelgrafiken. Außerdem können Grafiken in einer solchen Bibliothek nicht von Dritten verändert werden.

Das Programm 'grflib.exe' muß unter DOS mit den folgenden Parametern aufgerufen werden:

grflib dateiname [?] [/i] [@liste]

dateiname

Name der zu erzeugenden Grafikbibliothek

?

Hilfetext anzeigen

/i

Inhalt der über 'dateiname' angegebenen Grafikbibliothek anzeigen.

@liste

Durch das Zeichen Klammeraffe (@) wird der Name einer Textdatei eingeleitet, in welcher die Namen der BMP-Grafiken, die in die Bibliothek aufgenommen werden sollen, zeilenweise aufgelistet sind. Ist keine derartige Liste angegeben, werden alle Dateien mit der Dateikennung '.BMP' im aktuellen Verzeichnis in die Bibliothek eingetragen.

Um auf Grafiken in einer Bibliothek zugreifen zu können, muß innerhalb eines Programms die Bibliothek mit dem Befehl SetGraphicsLibrary als aktuelle Bibliothek eingestellt werden.

Beim Zugriff auf eine BMP-Datei aus einem Programm heraus wird zuerst versucht, die Grafik aus einer Grafik-Bibliothek zu laden. Ist keine Grafikbibliothek vorhanden oder ist die Grafik nicht in der Bibliothek enthalten, dann versucht das Laufzeitsystem die Grafik als externe Datei zu laden.

Das hat den Vorteil, daß während der Entwicklung eines Lernprogramms die Dateien als externe Grafiken geladen werden können. Erst zum Abschluß der Entwicklung wird eine Grafik-Bibliothek erzeugt, die dann mit dem Programm ausgeliefert wird.

Beispiel

1. grflib grafik

Erzeuge eine Grafik-Bibliothek namens 'grafik.lib'. Es werden alle Grafikdateien mit der Dateikennung '.BMP' im aktuellen Verzeichnis in die Bibliothek eingefügt.

2. grflib grafik @bmp.lst

Erzeuge eine Grafik-Bibliothek namens 'grafik.lib'. Die Dateinamen der BMP-Grafiken sind in der Datei 'bmp.lst' abgelegt.

3. grflib grafik /i

Zeige ein Verzeichnis aller Grafiken in der Bibliothek 'grafik.lib'.

Passwortschutz

Sie können Ihre Programme durch Vergabe eines Paßworts gegen die Einsichtnahme durch Dritte schützen. Um ein Paßwort zu vergeben, oder ein bereits eingestelltes Paßwort zu verändern, wählen Sie den Menüpunkt 'Paßwort bearbeiten' im Menü 'Optionen' aus.

Achten Sie bitte darauf, daß Sie Ihr Passwort nicht vergessen. Ohne Ihr Passwort können Sie Ihr Programm nicht mehr bearbeiten. Am besten vergeben Sie ein Passwort erst nach erfolgter Programmierung, bevor Sie Ihr Programm für den Vertrieb vorbereiten. Machen Sie am besten noch eine Sicherheitskopie ohne Passwort.

Die Firma Reitz-Software-Entwicklung wird unter keinen Umständen Passwörter aus eingesandten Programmdateien entfernen, da es für uns nicht nachvollziehbar ist, wem die Rechte dieser Dateien gehören.

Virenschutz

Die Programme 'author.exe' und 'runtime.exe' überprüfen sich beim Programmstart selbständig auf Veränderungen hin. Wenn dabei eine Veränderung der Programmdatei festgestellt wird, wird eine Warnung ausgegeben, da die Möglichkeit eines Befalls durch einen Computervirus besteht.

Diese Funktion kann auch unterdrückt werden, indem beim Programmstart der Aufrufparameter '/nv' angegeben wird.

Häufig gestellte Fragen (FAQ)

Im folgenden finden Sie Antworten zu Fragen, die nach unserer Erfahrung beim Anwender des öfteren auftreten.

1. Wie kann ich bei der Arbeit in der Entwicklungsumgebung den Arbeitsbereich eines Editorfensters vergrößern?
2. Wenn ich in der Entwicklungsumgebung ein Objekt teste, kann ich das Testobjekt durch Drücken der Taste ESC nicht schließen.
3. Die Schriften, die ich in einem von mir erstellten RSE-Author-Programm verwende, sehen anders aus, wenn das Programm auf einem anderen Rechner ausgeführt wird.
4. Wie kann ich ein Programm von der Bildschirmauflösung unabhängig programmieren?
5. Ich möchte eine Bitmap als Fensterhintergrund verwenden. Wie kann ich Seiten auf diesem Hintergrund ohne Flackereffekte durchblättern?
6. Wie kann ich in einem Textblock die Textattribute (kursiv, unterstrichen usw.) ändern?
7. Wie kann ich ein laufendes Programm über die Tastatur abbrechen?
8. Ich bekomme die Fehlermeldung 'Objektbezeichnung nicht bekannt', obwohl ich sicher bin, dieses Objekt/Element angelegt zu haben.
9. Wenn ich mit dem Befehl 'BackPage' die vorige Seite auflende, wird die Seite nicht korrekt wiederhergestellt.
10. Ich möchte mehrere Bitmaps mit 256 Farben auf dem Bildschirm darstellen. Wenn ich nach dem ersten Bitmap noch weitere Bitmaps lade, kommt es in den zuvor geladenen Grafiken zu Farbveränderungen.
11. Wie kann ich innerhalb der Entwicklungsumgebung komplette Objekte kopieren?
12. Wie kann ich innerhalb der Entwicklungsumgebung die Reihenfolge der Objekte im Listenfenster verändern?
13. Wie kann ich bei der Bearbeitung eines Objekts die Reihenfolge der Elemente gezielt verändern?
14. Wenn ich ein Programm zu Testzwecken starte, kann ich die gemachten Änderungen nicht mehr rückgängig machen.
15. Wenn ich eine Autodesk-Animator-Datei (FLI, FLC) mit dem VideoPlayer abspiele, kann ich die Datei nicht vor- oder zurückspulen. Außerdem werden im Control zur Anzeige der aktuellen Position wirre Zeichen angezeigt.

Wie kann ich bei der Arbeit in der Entwicklungsumgebung den Arbeitsbereich eines Editorfensters vergrößern?

Sie können mit der Leertaste das aktuelle Editorfenster in einen Vollbildmodus schalten. In diesem Modus steht die gesamte Bildschirmfläche für die Bearbeitung des Objekts zur Verfügung. Durch erneutes Drücken der Leertaste wird das Editorfenster wieder auf die vorige Größe verkleinert.

Falls die Umschaltung auf die Normalgröße nicht funktionieren sollte, klicken Sie mit der Maus irgendwo in die freie Fensterfläche und versuchen Sie es erneut.

Wenn ich in der Entwicklungsumgebung ein Objekt teste, kann ich das Testobjekt durch Drücken der Taste ESC nicht schließen.

Klicken Sie mit der Maus irgendwo in die freie Fensterfläche des Objekts und versuchen Sie es erneut.

Die Schriften, die ich in einem von mir erstellten RSE-Author-Programm verwende, sehen anders aus, wenn das Programm auf einem anderen Rechner ausgeführt wird.

Sie sollten bei der Erstellung Ihrer Programme grundsätzlich mit den Standardschriften von Windows arbeiten. Ebenso sollten die Schriftzuordnungen (logischer Schriftname <==> Schriftdatei) in der Datei 'win.ini' auf den Standardeinstellungen stehen.

Wenn Sie mit Schriften arbeiten, die nicht auf allen Windows-Systemen installiert sind, ist fast sicher mit Abweichungen zu rechnen, da auf anderen Rechnern diese Schrift nicht dargestellt werden kann. Ebenso ist mit Abweichungen zu rechnen, wenn Sie die Schriftzuordnungen in der Datei 'win.ini' verändert haben. Das kann ebenfalls passieren, wenn die Schriftzuordnungen auf dem Zielrechner verändert wurden.

Wenn Sie sich bei der Erstellung eines Programms auf das Vorhandensein von bestimmten Schriften und deren korrekter Zuordnung verlassen, sollten Sie dies in der Dokumentation Ihres Programms erwähnen.

Wie kann ich ein Programm von der Bildschirmauflösung unabhängig programmieren?

Generell gilt, daß Sie Ihr Programm in einem Fenster mit vordefinierter Größe ablaufen lassen müssen. Die einfachste Lösung ist, das MainWindow, welches den Container für alle Objekte des Programms darstellt, als zentriertes Standardfenster zu definieren. Bei höheren Auflösungen steht das Fenster dann mittig auf dem Bildschirm.

Wenn Ihr Programm nicht in einem Standardfenster laufen soll und Sie sicherstellen möchten, daß vom Bildschirmhintergrund nichts zu sehen ist, dann gehen Sie folgendermaßen vor:

1. Definieren Sie das MainWindow als Fenster vom Typ 0 (ChildWindow ohne Rahmen). Geben Sie für die X- und Y-Position, sowie für die Breite und Höhe des Fensters im Eigenschaften-Dialog jeweils 0 an. Wenn Breite und Höhe eines Fensters auf 0 gestellt sind, setzt das Laufzeitsystem bei der Ausführung automatisch die Breite und Höhe des Bildschirms ein, so daß ein solches Fenster immer den Bildschirm komplett ausfüllt.

Stellen Sie die Hintergrundfarbe des Fensters am besten auf Hellgrau ein.

2. Legen Sie ein Fenster namens WorkWindow (oder einem anderen Namen nach Ihrem Belieben) an, das die Auflösung hat, mit der Sie arbeiten möchten (z.B. 640 * 480). Dieses Fenster sollte auch vom Typ 0 (ChildWindow ohne Rahmen) sein. Laden Sie dieses Fenster im Open-Handler des MainWindow mit dem folgenden Befehl:

LoadWindow WorkWindow /center /border:24

Dieses Fenster sollte dann als Arbeitsfenster der Applikation dienen. Das MainWindow dient nur als Hintergrund und sollte keine Ausgaben erhalten.

Wenn Sie wie beschrieben vorgegangen sind, dann füllt das WorkWindow bei einer Auflösung von 640 * 480 den Bildschirm komplett aus. Bei höheren Auflösungen steht es mittig zentriert und von einem 3D-Rahmen umgeben auf dem Bildschirm. Darum herum liegt ein grauer Rand, der den Rest des Bildschirms ausfüllt. Vom Desktop ist nichts zu sehen.

Ich möchte eine Bitmap als Fensterhintergrund verwenden. Wie kann ich Seiten auf diesem Hintergrund ohne Flackereffekte durchblättern?

Definieren Sie bei der Bearbeitung des Fensterobjekts den Hintergrund als durchsichtig. Legen Sie ein Bitmap-Element an und setzen Sie es auf die Koordinaten 0/0. Die Bitmap sollte das Fenster komplett ausfüllen.

Wenn Sie dann die Seiten in diesem Fenster durchblättern, dann geben Sie beim Laden der Seite zusätzlich die folgenden optionalen Parameter an:

LoadPage Page001 /noclear /paint
NextPage Page??? /noclear /paint
PrevPage Page??? /noclear /paint

Beim Laden einer neuen Seite geschieht dann folgendes:

- 1.** Der Record (Aufzeichnung der Ausgaben) des Fensterhintergrunds wird nicht gelöscht. Die Hintergrund-Bitmap bleibt erhalten.
- 2.** Der Record der Seite wird gelöscht. Das physikalische Löschen des Fensters wird allerdings unterdrückt. Lediglich die Aufzeichnungen für den Wiederaufbau der Seite sind weg.
- 3.** Es wird ein Neuzeichnen des Fensters angestoßen. Da nur noch die Hintergrund-Bitmap vorhanden ist, wird diese neu ausgegeben und überschreibt dabei die alten Seiteninformationen.
- 4.** Die neue Seite wird geladen und auf der Bitmap als Hintergrund ausgegeben.

Wie kann ich in einem Textblock die Textattribute (kursiv, unterstrichen, usw.) ändern?

Während der Eingabe eines Textblocks können jederzeit die Attribute sowie die Hinter- und Vordergrundfarbe des Textes verändert werden. Das ist z.B. sinnvoll, wenn ein einzelnes Wort unterstrichen, kursiv oder in einer anderen Farbe dargestellt werden soll. Attributanweisungen werden innerhalb des Textes in geschweifte Klammern eingeschlossen {.....}. Den genauen Aufbau dieser Anweisungen finden Sie im Abschnitt [Textattribute](#).

Wie kann ich ein laufendes Programm über die Tastatur abbrechen?

Die Tastenkombination STRG-Q ist mit einer Abbruchfunktion belegt. Wird diese Tastenkombination gedrückt, fragt das Laufzeitsystem nach, ob das Programm beendet werden soll. Wird dies bestätigt, wird die Ausführung des aktuellen Programms beendet.

Ich bekomme die Fehlermeldung 'Obektbezeichnung nicht bekannt', obwohl ich sicher bin, dieses Objekt/Element angelegt zu haben.

Möglicherweise ist das Objekt/Element gelöscht worden. Bei Ausführung der Befehle 'LoadPage', 'NextPage', 'PrevPage' usw. werden z.B. alle maussensitiven Felder und Hypertextfelder im Fenster gelöscht. Wird danach versucht, einen Befehl auf eines dieser Elemente auszuführen, kann der oben beschriebene Fehler auftreten.

Wenn ich mit dem Befehl 'BackPage' die vorige Seite aufblende, wird die Seite nicht korrekt wiederhergestellt.

Achten Sie bitte darauf, daß alle zusätzlichen Ausgaben, die in eine Seite erfolgen sollen, im PageEnter-Handler der Seite vorgenommen werden. Dabei kann es sich z.B. um zusätzliche Seiten-Objekte oder Bitmaps handeln, die der aktuellen Seite mit Überblendeffekten hinzugefügt werden sollen.

Wenn Sie eine Seite im Click-Handler eines Schalters oder maussensitiven Feldes laden und danach in diesem Handler noch zusätzliche Ausgaben in die Seite machen, dann können diese Ausgaben mit 'BackPage' nicht wiederhergestellt werden. Deshalb sollten diese Ausgaben in den PageEnter-Handler der Seite verlagert werden.

Das gesagte gilt auch für das Umdefinieren von Schaltern. Wenn beim Aufblenden einer Seite die Funktionsweise von Schaltern mit 'SetScript' umdefiniert werden soll, dann sollte das grundsätzlich im PageEnter-Handler der Seite erfolgen.

Ich möchte mehrere Bitmaps mit 256 Farben auf dem Bildschirm darstellen. Wenn ich nach dem ersten Bitmap noch weitere Bitmaps lade, kommt es in den zuvor geladenen Grafiken zu Farbveränderungen.

Sie haben für Ihre Grafikkarte einen Treiber installiert, der nur 256 Farben unterstützt. Bitmaps mit 256 Farben enthalten eine Palette, welche die Rot-, Grün- und Blauanteile der Farben in der Bitmap definiert. Beim Laden der Bitmap wird diese Palette eingestellt, um die Bitmap farbrichtig anzeigen zu können. Wurden zuvor andere Bitmaps mit anderen Paletten geladen, können diese nicht mehr richtig dargestellt werden.

Für die Lösung dieses Problems gibt es drei Möglichkeiten:

1. Es wird immer nur eine Bitmap auf dem Bildschirm angezeigt.
2. Sie betreiben Ihren Rechner in einem High-Color (64K Farben) oder True-Color (16 Mio Farben) Modus.
3. Sie versehen Ihre Bitmaps mit einer einheitlichen Palette.

Nähere Informationen zum Umgang mit Bitmaps finden Sie im Abschnitt [Bitmaps](#) im Kapitel [Tips und Tricks](#).

Wie kann ich innerhalb der Entwicklungsumgebung komplette Objekte kopieren?

Über das Clipboard können nur Elementgruppen kopiert werden. Um komplette Objekte zu kopieren, kann man sich aber eines kleinen Tricks bedienen. Kopieren Sie Ihr Programm unter einem anderen Namen. Mittels der Funktion 'Einlagern' im Menü 'Objekt' können Sie dann die zu kopierenden Objekte in das bearbeitete Programm einfügen.

Wie kann ich innerhalb der Entwicklungsumgebung die Reihenfolge der Objekte im Listenfenster verändern?

Sie können einzelne Einträge in der Liste verschieben, indem Sie die Einträge selektieren und anschließend bei gedrückter SHIFT-Taste die Pfeilhoch-/Pfeilrunter-Taste drücken.

Wie kann ich bei der Bearbeitung eines Objekts die Reihenfolge der Elemente gezielt verändern?

Positionieren Sie den Mauszeiger über der Arbeitsfläche und betätigen Sie die rechte Maustaste. Darauhin wird in einem Dialogfenster eine Liste aller Elemente im aktuellen Editorfenster angezeigt. Sie können die Elemente in dieser Liste verschieben, indem Sie das gewünschte Element selektieren und anschließend bei gedrückter SHIFT-Taste die Pfeilhoch-/Pfeilrunter-Taste drücken.

Wenn ich ein Programm zu Testzwecken starte, kann ich die gemachten Änderungen nicht mehr rückgängig machen.

Vor dem Start eines Programms aus der Entwicklungsumgebung heraus wird das komplette Programm automatisch abgespeichert, da das Laufzeitsystem das Programm als Datei auf der Platte erwartet.

Wenn ich eine Autodesk-Animator-Datei (FLI, FLC) mit dem VideoPlayer abspiele, kann ich die Datei nicht vor- oder zurückspulen. Außerdem werden im Control zur Anzeige der aktuellen Position wirre Zeichen angezeigt.

Der Standard-MCI-Treiber unterstützt in Verbindung mit FLI/FLC-Dateien die Funktionen Vor- und Zurückspulen nur, indem im Hintergrund die Datei vom Anfang bis zu der gewünschten Position abgespielt wird, was sehr zeitaufwendig ist. Des weiteren liefert der Treiber keine Informationen über die aktuelle Position des Dateizeigers. FLI-Dateien sollten nicht über den VideoPlayer abgespielt werden. Diese Dateien machen Sinn in Verbindung mit dem Befehl PlayVideo, um animierte Trailer oder Logos anzuzeigen, die in einer Endlosschleife abgespielt werden.

Was muß in Verbindung mit einem RSE-Author-Programm kopiert werden?

Folgende Komponenten müssen in Verbindung mit einem RSE-Author-Programm kopiert werden:

- 1.** Das lauffähige RSE-Author-Programm mit der Dateikennung .PRG.
- 2.** Alle Grafiken, Grafikbibliotheken und sonstige Dateien, die zu dem Programm gehören.
- 3.** Das Laufzeitsystem (runtime.exe).
- 4.** Die Schriftendatei (author.fnt).
- 5.** Folgende DLL-Dateien

sv221bw.dll
tl221bw.dll

Die Weitergabe der Dateien runtime.exe, author.fnt, sv221bw.dll und tl221bw.dll an Dritte ist erlaubt.

Alle anderen Programme, Dateien und Dokumentationen, die zu RSE-Author gehören, sind durch nationale und internationale Copyright-Vorschriften geschützt und dürfen nicht weitergegeben werden.

Weiterentwicklung des Autorensystems

Das Autorensystem RSE-Author wird ständig weiterentwickelt und verbessert. Da es sich hier um ein deutsches Produkt handelt, besteht für Sie als Anwender die Möglichkeit, durch Ihre Anregungen und Ideen auf die Entwicklung Einfluß zu nehmen. Schreiben Sie uns einfach und teilen Sie uns Ihre Verbesserungsvorschläge mit. Die Adresse finden Sie in der Einleitung dieses Online-Handbuchs. Haben Sie bitte Verständnis dafür, daß nur Verbesserungsvorschläge aufgegriffen werden können, die für alle Anwender interessant sind. Funktionen, die nur von einer bestimmten Firma oder für ein bestimmtes Projekt benötigt werden, können in eine Standard-Software, wie sie RSE-Author darstellt, nicht aufgenommen werden.

Natürlich sind wir auch an Fehlermeldungen interessiert. Obwohl wir uns bemüht haben, das Autorensystem so gründlich wie möglich zu testen, ist es unmöglich, alle denkbaren Situationen, denen das Autorensystem im Laufe der Zeit ausgesetzt wird, herbeizuführen. Es wird deshalb sicher noch der eine oder andere Fehler auftreten. Falls das der Fall ist, teilen Sie uns dies bitte mit. Wir werden uns bemühen, jeden Fehler schnellstmöglich aus der Welt zu schaffen.

Im folgenden möchten wir Ihnen kurz vorstellen, wie die Entwicklung von RSE-Author weiter vorstatten gehen wird. Diese Angaben sind unverbindlich.

Objekte

Folgende Objekte kommen in den nächsten Versionen neu hinzu:

- Menü** (Interaktive Erstellung von kaskadierbaren Pulldown- und Popup-Menüs).
- Animation** (Interaktive Erstellung von Sprite- und Stapel-Animationen).
- Video** (Video-Dateien als eigenständige Objekte eröffnen)
- Audio** (Audio-Dateien als eigenständige Objekte eröffnen)
- Dll** (Dll's als eigenständige Objekte eröffnen und Funktionen darin aufrufen)

Elemente

Folgende neue Elemente sind für die nächsten Versionen geplant:

- CheckBox**
- RadioButton**
- Input-Field** (einzeilige Texteingabe)
- Editor-Field** (mehrzeilige Texteingabe)
- DropDown-ListBox**
- ComboBox**
- DropDown-ComboBox**

Script-Sprache

Erweiterung der Script-Sprache um folgende Sprach-Elemente:

- mehrdimensionale Variablen
- symbolische Labels und goto-Anweisung
- for-to-Schleife
- while-Schleife
- do-loop-Schleife
- if-then-else-Abfragen
- komplexe Analyse-Möglichkeiten für Variablen

- Erzeugung aller Objekte und Elemente aus der Script-Sprache heraus
- virtuelle Devices (unsichtbare Fenster im Arbeitsspeicher)
- Prozeduren
- Libraries
- File-IO
- Datenbank-Abfragen

Datenbank

Das Autorensystem wird des weiteren um eine dbase-kompatible Datenbank erweitert.

Unterstützung weiterer Betriebssystem-Plattformen

Bei der Entwicklung von RSE-Author wurde sorgfältig auf Plattform-Unabhängigkeit geachtet. Aus diesem Grund ist die Portierung des Autorensystems auf die folgenden System-Plattformen möglich:

Windows 3.x	-	Intel
Windows 95	-	Intel
OS/2 2.x	-	Intel
OS/2 Warp	-	Intel
Macintosh System 7.x	-	Macintosh 68k
Macintosh System 7.x	-	Macintosh PowerPC
Windows NT	-	Intel
Windows NT	-	DEC Alpha
Motif	-	SUN Sparc
Motif	-	IBM RS 6000
Motif	-	HP 9000
Motif	-	SCO Unix
Motif	-	UnixWare

Welche Plattformen über Windows hinaus unterstützt werden, hängt von der jeweiligen Nachfrage ab.

Programmierung

In den folgenden Abschnitten wird auf die Art und Weise eingegangen, in der mit dem Autorensystem RSE-Author Applikationen erstellt werden.

Dabei wird jeder Bereich der Programmierung ausführlich besprochen und aufgezeigt, wie man mit den Möglichkeiten des Autorensystems schnell und effizient die gestellten Probleme lösen kann.

Ein Programm setzt sich stets aus einer Reihe von Objekten (Seiten, Fenster usw.) zusammen, die wiederum eine Anzahl von Elementen (Linien, Kreise, Schalter usw.) enthalten. Die Verknüpfung der Objekte, sowie die Steuerung des Programms erfolgt über die integrierte Script-Sprache.

Eine Liste aller Objekte und Elemente, sowie der dazu gehörigen Befehle finden Sie im Befehls-Index dieses Handbuchs.

Sehen Sie sich bitte auch das beiliegende Demo-Programm an. Dieses Programm wurde so ausgelegt, daß es die Möglichkeiten des Autorensystems möglichst anschaulich demonstriert.

Themen dieses Kapitels

Grundlagen
Script-Sprache

Seite
Fenster
Dokument

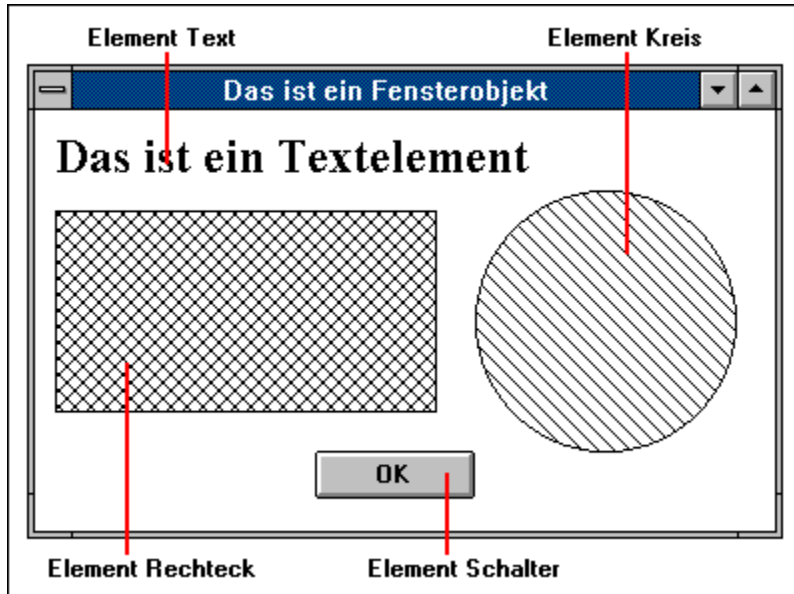
Schalter
Hyperlink
Hypertext
ListBox

Bitmaps
Video
Audio

Ein erstes Programm
Performance verbessern
Tips und Tricks

Grundlagen

Ein RSE-Author-Programm besteht aus einer Anzahl von Objekten. Dabei kann es sich um sogenannte Seiten, Fenster oder Dokumente handeln. Diese Objekte wiederum enthalten Elemente wie z.B. Linien, Kreise, Rechtecke oder auch Schalter, Listboxen usw.



Da sich bei modernen Betriebssystemen wie Windows oder OS/2 alle Programmein- und ausgaben innerhalb von Fenstern abspielt, muß für jedes Programm grundsätzlich mindestens ein Fenster eröffnet werden, in dem sich die Aktionen des Programms abspielen.

MainWindow

Das Autorenprogramm legt für jede neue Applikation ein Fensterobjekt namens 'MainWindow' an. Das ist das Hauptfenster des Programms, das grundsätzlich eröffnet wird. Sie können Position, Größe und Aussehen dieses Fensters frei einstellen und damit Ihren Wünschen anpassen. Von diesem Objekt aus erfolgen alle anderen Aktionen innerhalb eines RSE-Author-Programms.

Sie können das Objekt 'MainWindow' weder löschen noch umbenennen. Beachten Sie bitte, daß das MainWindow kein Dialogfenster sein darf. Selbst wenn Sie bei den Eigenschaften dieses Objekts den Typ Dialogfenster einstellen, wird das Fenster trotzdem als Standardfenster eröffnet.

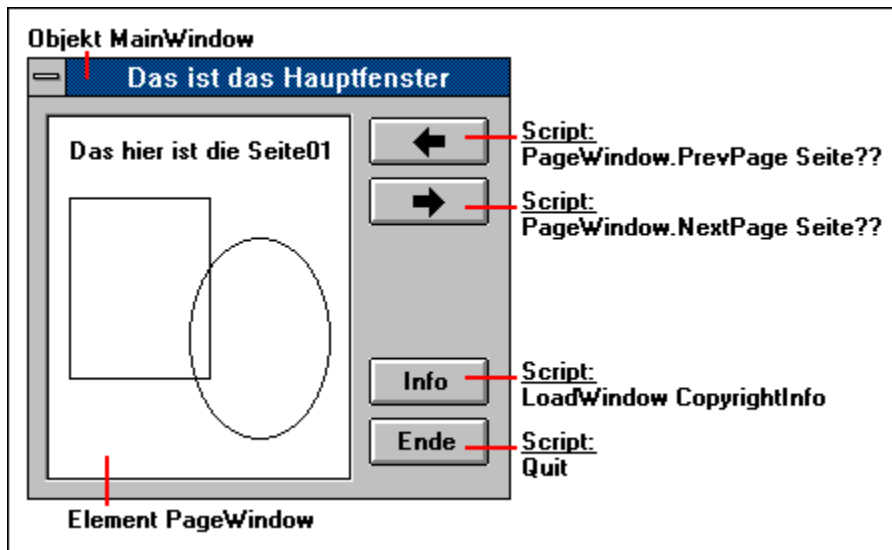
Das Schließen des Objekts 'MainWindow' beendet das Programm.

Seite

Ein Objekt vom Typ Seite stellt den Inhalt für ein Fenster dar. Ähnlich wie ein Buch kann ein RSE-Author-Programm beliebig viele Seiten enthalten, die innerhalb eines Fensters durchgeblättert werden können.

Die normale Vorgehensweise bei der Entwicklung eines Programms wird immer sein, die Informationen, die das Programm vermitteln soll, in einer Anzahl von Seiten abzulegen. Dann wird ein Fenster angelegt, in dem die Seiten durchgeblättert werden können. Das

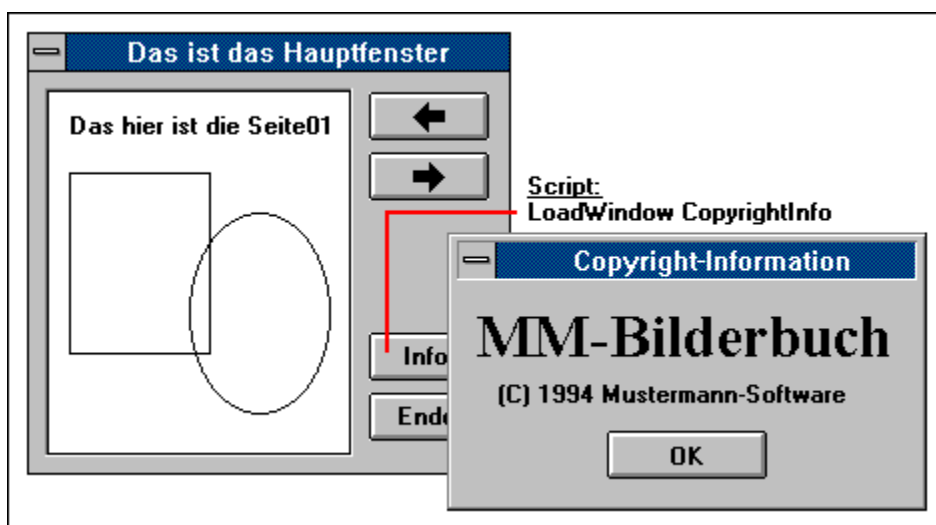
Laden und Weiterschalten der Seiten wird von der Script-Sprache mit den Befehlen LoadPage, NextPage und PrevPage erledigt.



Nehmen wir an, Sie haben 10 Seiten mit der Bezeichnung Seite01 bis Seite10 angelegt. Diese 10 Seiten möchten Sie in einem Unterfenster namens PageWindow, das ein Element Ihres Hauptfensters ist, durchblättern. Der prinzipielle Aufbau des Programms könnte dann wie oben aussehen.

Fenster

Alle Aktionen innerhalb eines Programms spielen sich in Objekten vom Typ Fenster ab. Beim Start eines Programms wird automatisch das Fenster 'MainWindow' eröffnet. Danach können durch Interaktionen des Benutzers (z.B. das Anklicken eines Schalters) weitere Fenster eröffnet werden. Das Eröffnen von weiteren Fensters wird von der Script-Sprache mit dem Befehl LoadWindow erledigt.



Dokument

Ein Objekt vom Typ Dokument ist für die Ausgabe auf einem Drucker vorgesehen. Bei der Bearbeitung dieses Objekts sind nicht alle Elemente verfügbar, da Bedienungselemente wie z.B. Schalter für die Ausgabe auf einen Drucker nicht benötigt werden. Des Weiteren werden bei der Bearbeitung eines Druckdokuments alle Koordinaten in zehntel Millimeter angegeben statt wie gewöhnlich in Pixeln.

Um ein Dokument auf den Drucker auszugeben, muß der Drucker mit dem Befehl OpenPrinter eröffnet werden. Danach kann eine oder auch mehrere Druckseiten mit dem Befehl LoadDocument geladen und danach mit PrintPage ausgegeben werden.

Für Dokumente können nur Bitmaps im BMP-Format verwendet werden.

Script-Sprache

Die Erstellung eines RSE-Author-Programms muß in zwei Stufen aufgeteilt werden.

- 1.** Erstellung aller zum Programm gehörenden Objekte.
- 2.** Verknüpfung der Objekte durch die integrierte Script-Sprache.

Die in RSE-Author integrierte Script-Sprache besteht aus einfachen Basic-ähnlichen Kommandos, die für die Steuerung eines Programms zuständig sind.

Eine genauere Erklärung, wie die Programmsteuerung mit der Script-Sprache funktioniert, finden Sie im Abschnitt Script-Sprache dieses Kapitels.

Script-Sprache

Einigen Objekten und Elementen können sogenannte 'Scripts' zugeordnet werden. Das sind kurze Basic-ähnliche Befehlssequenzen, die den Ablauf eines Programms steuern.

Ein Objekt oder Element kann ein oder mehrere Scripts enthalten, die beim Auftreten von bestimmten Systemereignissen (z.B. Öffnen eines Fensters, Anklicken eines Schalters) ausgeführt werden sollen. Im folgenden wird eine Liste der Objekte/Elemente und deren zugeordneten Scripts aufgeführt:

Objekt Fenster

Open-Script

Wird nach dem Eröffnen des Fensters aufgerufen.

Close-Script

Wird vor dem Schließen eines Fensters aufgerufen.

Move-Script

Wird nach dem Verschieben eines Fensters aufgerufen.

Resize-Script

Wird nach einer Größenänderung eines Fensters aufgerufen.

Click-Script

Wird nach dem Anklicken eines Fensters mit der Maus aufgerufen.

DoubleClick-Script

Wird nach dem Doppelklicken eines Fensters mit der Maus aufgerufen.

Objekt Seite

Enter-Script

Wird nach dem Laden einer Seite aufgerufen.

Exit-Script

Wird beim Verlassen einer Seite aufgerufen

Element Schalter

Click-Script

Wird nach der Betätigung des Schalters aufgerufen.

Element Feld

Click-Script

Wird nach dem Anklicken des Feldes aufgerufen.

DoubleClick-Script

Wird nach dem Doppelklicken eines Feldes aufgerufen.

MoveIn-Script

Wird nach dem Hineinbewegen des Mauszeigers in ein Feld aufgerufen.

MoveOut-Script

Wird nach dem Herausbewegen des Mauszeigers aus einem Feld aufgerufen.

Element HyperText

Click-Script

Wird nach dem Anklicken des Feldes aufgerufen.

DoubleClick-Script

Wird nach dem Doppelklicken eines Feldes aufgerufen.

MoveIn-Script

Wird nach dem Hineinbewegen des Mauszeigers in ein Feld aufgerufen.

MoveOut-Script

Wird nach dem Herausbewegen des Mauszeigers aus einem Feld aufgerufen.

Element ListBox

Select-Script

Wird nach der Auswahl eines Listeneintrags aufgerufen.

DoubleClick-Script

Wird nach dem Doppelklicken eines Listeneintrags aufgerufen.

Objektnamen und Aliasnamen

Jedem Objekt wird beim Anlegen ein Name zugeteilt. Das Hauptfenster trägt z.B. den Namen 'MainWindow'. Über diesen Namen kann das Objekt später auch innerhalb eines Programms angesprochen werden. Dem Namen des Objekts wird einfach durch einen Punkt getrennt, der auszuführende Befehl zugeordnet.

Beispiel: `MainWindow.DrawText "RSE-Author ist ein flexibles Autorensystem"`

In dem obigen Beispiel wird ein Text in ein Fenster ausgegeben. Durch die vorangestellte Objektbezeichnung weiß das Autorensystem, welches Fensterobjekt die Textausgabe erhalten soll.

Zusätzlich zu dem Namen kann ein Objekt noch ein sogenanntes Alias erhalten. Ein Alias ist ein Kurzname unter dem das Objekt angesprochen werden kann. Beim Anlegen von Objekten wird man in der Regel aussagekräftige Namen wählen, wie 'MainWindow' oder 'CopyrightInfo', um zu einem späteren Zeitpunkt am Objektnamen erkennen zu können, wozu das Objekt dient. Allerdings bedingen diese langen Namen viel Tipparbeit, wenn für jeden Befehl, der auf dieses Objekt wirken soll, der komplette Name vorangestellt werden soll. Deswegen besteht die Möglichkeit für ein Objekt noch einen Kurznamen zu vergeben. Das kann im Eigenschaftendialog des Objekts oder durch den Befehl ALIAS geschehen.

Beispiel: `MainWindow.Alias win
win.DrawText "RSE-Author ist ein flexibles Autorensystem"`

Folgende Objekte und Elemente können Namen zugeordnet bekommen:

Fenster
Seite
Dokument
Schalter

Feld
Hypertext
Listbox

Folgende Objekte und Elemente können sowohl Namen als auch Aliasnamen erhalten:

Fenster
Schalter
Feld
Hypertext
Listbox

Objektorientierte Programmierung

Die Script-Sprache arbeitet objektorientiert. Einer Objektbezeichnung folgt durch einen Punkt getrennt, der auszuführende Befehl. Der Befehl bezieht sich dann auf das bezeichnete Objekt.

Beispiel: MainWindow.LoadPage Seite532

Objektbezeichnungen können auch hierarchisch verknüpft werden, d.h. der Bezeichnung eines Fensters, folgt durch einen Punkt getrennt, die Bezeichnung eines Elements, danach folgt wieder durch einen Punkt getrennt, der eigentliche Befehl, der sich auf das Element bezieht. Dadurch ist es möglich, in verschiedenen Fenstern die gleichen Elementnamen zu verwenden.

Beispiel: MainWindow.button01.Enable ! Button01 in MainWindow wird aktiviert
SubWindow.button01.Enable! Button01 in SubWindow wird aktiviert

Die Objektbezeichnung kann auch weggelassen werden. In dem Fall bezieht sich der Befehl auf das Objekt oder Element, das die Ausführung des Scripts in Auftrag gegeben hat. Wird ein Befehl ohne Objektbezeichnung von einem Bedienungselement (Schalter, Feld, usw.) in Auftrag geben und falls dieses Element den Befehl nicht kennt, wird der Befehl zur Ausführung dem Fenster übergeben, welches das Element enthält. Ist der Befehl dem Fenster unbekannt, wird geprüft, ob es sich um einen allgemeinen Befehl handelt. Trifft auch das nicht zu, wird eine Fehlermeldung ausgegeben.

Beispiel:

In einem Schalter-Script sind folgende Befehle zu finden.

Sound warning
Close

Bei der Bearbeitung des ersten Befehls geht das Autorensystem folgendermaßen vor: Zuerst wird überprüft, ob das Element Schalter den Befehl SOUND kennt. Das ist nicht der Fall, also wird der Befehl zur Ausführung dem Fenster übergeben, in dem der Schalter sitzt. Das Autorensystem stellt fest, daß das Objekt Fenster den Befehl SOUND ebenfalls nicht kennt und übergibt den Befehl zum Schluß an die Applikationsebene, wo der Befehl bekannt ist und ausgeführt wird.

Jetzt wird der Befehl CLOSE bearbeitet. Der Befehl CLOSE ist dem Element Schalter nicht bekannt. Der Befehl wird wiederum dem Fensterobjekt übergeben, in dem der Schalter sitzt. Dem Fensterobjekt ist dieser Befehl bekannt und wird ausgeführt, d.h. das Fenster wird geschlossen.

Es ist also nicht immer notwendig, einem Befehl die Bezeichnung eines Objekts oder Elements voranzustellen. Soll der Befehl auf das Objekt oder Element wirken, welches das Befehlsscript enthält, in dem der Befehl sitzt, kann der Objektname auch weggelassen werden.

Fixe Parameter und optionale Parameter

Den meisten Befehlen folgen noch ein oder mehrere Parameter, die für die Ausführung des Befehls benötigt werden. Man unterscheidet dabei zwischen fixen und optionalen Parametern.

Fixe Parameter müssen angegeben werden. Fehlt ein fixer Parameter, führt das zu einem Syntaxfehler bei der Ausführung eines Programms. Dem Befehl DrawRectangle müssen z.B. vier fixe Parameter mitgegeben werden, welche die Koordinaten, sowie die Breite und Höhe des zu zeichnenden Rechtecks angeben.

DrawRectangle 100 100 400 300

Viele Befehle können zusätzlich noch sogenannte optionale Parameter erhalten. Ein solcher Parameter kann, muß aber nicht angegeben werden. Ein optionaler Parameter wird immer durch einen Schrägstrich ('/') eingeleitet. Dem Schrägstrich folgt der Namensteil des Parameters, der identifiziert, um welchen optionalen Parameter es sich handelt. Dem Namensteil können zusätzlich noch durch einen Doppelpunkt getrennt, ein oder auch mehrere Werte folgen. Im folgenden sehen Sie einige Beispiele.

DrawRectangle 100 100 400 300

DrawRectangle 100 100 400 300 /xround:20 /yround:20

LoadWindow SubWindow

LoadWindow SubWindow /center /border:24

LoadPage Page001

LoadPage Page001 /clear /fade:1:2:0

Seite

Seitenobjekte stellen praktisch die Inhalte für ein Fenster dar. Ähnlich wie in einem Buch, können Sie Seiten innerhalb eines Fensters durchblättern.

Um mit Seiten zu arbeiten, müssen Sie folgendermaßen vorgehen:

1. Eröffnen Sie innerhalb des Applikationsfensters ein Unterfenster, in das die Seiten geladen werden sollen.
2. Legen Sie eine Schalterleiste (oder sonstige Bedienungselemente) an, mit denen das Blättern zwischen den verschiedenen Seiten realisiert wird. Achten Sie darauf, daß keine Bedienungselemente in dem Fenster liegen, das die Seiten aufnehmen soll. Das Seitenfenster sollte nur Informationen (also Text, Grafik, Hyperlink- und Hypertextfelder) aufnehmen.
3. Belegen Sie die Schalter mit den nötigen Befehlen, um zwischen den Seiten zu blättern.
4. Sie können beim Bearbeiten eines Seitenobjekts zusätzlich Hyperlink- und/oder Hypertextfelder definieren, mit denen zu anderen Seiten gewechselt werden kann.

Zum Laden von Seiten stehen Ihnen die folgenden Befehle zur Verfügung:

LoadPage
FirstPage
LastPage
NextPage
PrevPage
BackPage
SelectPage

Hintergrund und Vordergrund

Bei der Verarbeitung von Seitenobjekten innerhalb eines Fensters muß man zwischen dem Hintergrund und dem Vordergrund des Fensters unterscheiden. Alle Elemente, die direkt bei der Erstellung eines Fensterobjekts in der Entwicklungsumgebung in dieses Fenster eingetragen werden, stehen im Hintergrund. Alle Elemente, die beim Laden eines Seitenobjekts nachträglich in dieses Fenster eingetragen werden, stehen im Vordergrund.

Diese Unterscheidung ist wichtig, da der Hintergrund und der Vordergrund unabhängig voneinander gelöscht werden können (siehe Befehle Clear und LoadPage).

Ein Beispiel:

Sie möchten innerhalb eines Fensters eine Bitmap-Grafik als Hintergrund für alle Ausgaben in dem Fenster verwenden. Beim Weiterblättern soll vor dem Hintergrund dieser Grafik die alte Seite entfernt und eine neue Seite auf diesen Hintergrund geladen werden.

So etwas kann realisiert werden, indem die Hintergrund-Bitmap bei der Erstellung des Fensterobjekts direkt in das Fenster eingetragen wird. Das Füllmuster für das Fenster sollte auf 'unsichtbar' eingestellt werden. Beim Laden einer neuen Seite in diesem Fenster wird lediglich der Vordergrund gelöscht. Die im Hintergrund eingetragene Bitmap bleibt stehen. Um unliebsame Flackereffekte zu vermeiden, sollten die Befehle zum Laden der Seite mit folgenden Parametern aufgerufen werden:

LoadPage Page032 /noclear /paint
NextPage Page??? /noclear /paint
PrevPage Page??? /noclear /paint

Die Aufteilung in Hinter- und Vordergrund wird erreicht, indem die Ausgaben in getrennten Metafiles aufgezeichnet werden. Dabei wird zwischen dem Page-Metafile und dem Window-Metafile unterschieden.

Den sogenannten Draw-Befehlen (DrawBitmap, DrawLine, DrawText usw.) kann mittels des optionalen Parameters '/record' mitgeteilt werden, ob die Ausgabe in den Hintergrund oder in den Vordergrund eingetragen werden soll. Standardmäßig wird eine solche Ausgabe in den Vordergrund eingetragen.

Scripts

Jeder Seite können zwei Befehls-Scripts zugeordnet werden, die nach dem Laden einer Seite (PageEntry-Script) und vor dem Verlassen (PageExit-Script) ausgeführt werden.

1. PageEntry-Script

Im PageEntry-Script sollten alle zusätzlichen Ausgaben und Einstellungen für die Seite vorgenommen werden. Wenn Sie z.B. nach dem Laden einer Seite mittels Überblendungen zusätzliche Bildobjekte einkopieren wollen, sollten Sie das in diesem Script tun. Hier können auch Schalter gesperrt, freigegeben oder umdefiniert werden. Es ist sehr wichtig, alle Aktionen, die mit der Seite zu tun haben, hier abzuwickeln, da nur so sicher gestellt werden kann, daß bei einem Rücksprung mit dem Befehl BackPage die Seite mit allen zusätzlich ausgegebenen Elementen wiederhergestellt werden kann.

2. PageExit-Script

Im PageExit-Script dagegen können z.B. gesperrte Schalter wieder freigegeben werden. Der PageExit-Script einer Seite wird unmittelbar vor dem Laden der nächsten Seite aufgerufen. Durch den Befehl Clear wird das aktuelle PageExit-Script gelöscht. Wird danach eine neue Seite geladen, wird das PageExit-Script der vorigen Seite nicht mehr ausgeführt.

Fenster

Einer der wichtigsten Aspekte in der Programmierung von RSE-Author-Programmen ist die Gestaltung der Bildschirmfenster. Alle Ausgaben, ob Text, Grafik oder Animation erfolgen in Fenstern auf dem Bildschirm.

Fenster werden in der Entwicklungsumgebung als Objekte erstellt. Die Eigenschaften des Fensters, sowie dessen Inhalte können bequem interaktiv definiert werden. Jedes Fenster erhält dabei einen eindeutigen Namen, unter dem das Fenster später innerhalb der Script-Sprache angesprochen werden kann. Auf das Fenster können Befehle ausgeführt werden (siehe [Befehls-Index](#)).

Mit dem Befehl [LoadWindow](#) kann ein Fenster auf den Bildschirm gebracht werden. Dem Befehl wird als Parameter einfach der Name des Fensterobjekts übergeben, das eröffnet werden soll. Die einzige Ausnahme ist das Fenster mit dem Namen 'MainWindow', das vom Laufzeitsystem beim Start eines Programms automatisch geöffnet wird.

Applikationsfenster

Dem Fenster mit der Bezeichnung 'MainWindow' kommt eine besondere Rolle zu. Ein Fensterobjekt mit diesem Namen wird vom Autorensystem beim Anlegen eines neuen Programms automatisch erzeugt. Dieses Objekt kann weder gelöscht noch umbenannt werden. Bei 'MainWindow' handelt es sich um das sogenannte Applikationsfenster, das vom Laufzeitsystem automatisch als erstes Fenster eröffnet wird. Dieses Fenster stellt praktisch den Container für ein Programm dar.

Wird das Applikationsfenster vom Benutzer geschlossen, dann wird das Lernprogramm automatisch beendet. Des weiteren liegen alle nachfolgend eröffneten Fenster immer vor dem Applikationsfenster.

Das Applikationsfenster darf kein Dialogfenster sein. Selbst wenn Sie dieses Objekt im Eigenschaften-Dialog als Dialogfenster einstellen, wird es später vom Programm aus als Standardfenster eröffnet.

Fenstertypen

Für die Entwicklung von Programmen stehen Ihnen die folgenden fünf Fenstertypen zur Verfügung:

- 0** - Fenster ohne Rahmen
- 1** - Fenster mit Rahmen
- 2** - Standardfenster
- 3** - Dialogfenster
- 4** - 3D-Dialogfenster

Der Typ eines Fensters wird im Eigenschaften-Dialog des Fensterobjekts innerhalb der Entwicklungsumgebung eingestellt.

Fenster vom Typ 0 sind einfache Flächen auf dem Bildschirm ohne Rahmen. Beim Eröffnen werden sie als Kindfenster in das aktuelle Fenster eingetragen. Fenster vom Typ 1 entsprechen den Fenstern vom Typ 0. Sie sind lediglich noch von einem einfachen Strichrahmen umgeben. Diese beiden Fenstertypen werden relativ zu ihrem Vaterfenster positioniert.

Das normale Arbeitsfenster ist das Fenster vom Typ 2. Es besitzt ein Fenstermenü in der

linken oberen Ecke, sowie eine Titelleiste. Es kann verschoben, in der Größe verändert, sowie minimiert und maximiert werden. Welche dieser Möglichkeiten für das Fenster erwünscht sind, kann ebenfalls im Eigenschaften-Dialog des Fensterobjekts eingestellt werden.

Fenster vom Typ 3 und 4 nennt man Dialogfenster. Dieser Fenstertyp sollte für Statusmeldungen, Copyright-Informationen usw. verwendet werden. Unter Windows erhält ein Dialogfenster vom Typ 4 einen 3D-Look. Unter OS/2 unterscheiden sich die beiden Typen nicht. Standardfenster und Dialogfenster werden immer relativ zum gesamten Bildschirm positioniert.

Vaterfenster

Jedes Fenster (außer dem Applikationsfenster) bekommt bei der Eröffnung ein sogenanntes Vaterfenster zugeteilt. Ein Fenster muß immer vor seinem Vaterfenster liegen. Standardmäßig wird das Applikationsfenster (MainWindow) als Vaterfenster eingesetzt. Diese Zuordnung kann beim Laden eines Fensters mit dem Befehl LoadWindow durch den optionalen Parameter /PARENT verändert werden.

Hintergrund und Vordergrund

Bei der Verarbeitung von Seitenobjekten innerhalb eines Fensters muß man zwischen dem Hintergrund und dem Vordergrund des Fensters unterscheiden. Alle Elemente, die direkt bei der Erstellung eines Fensterobjekts in der Entwicklungsumgebung in dieses Fenster eingetragen werden, stehen im Hintergrund. Alle Elemente, die beim Laden eines Seitenobjekts nachträglich in dieses Fenster eingetragen werden, stehen im Vordergrund.

Diese Unterscheidung ist wichtig, da der Hintergrund und der Vordergrund unabhängig voneinander gelöscht werden können (siehe Befehle Clear und LoadPage).

Ein Beispiel:

Sie möchten innerhalb eines Fensters eine Bitmap-Grafik als Hintergrund für alle Ausgaben in dem Fenster verwenden. Beim Weiterblättern soll vor dem Hintergrund dieser Grafik die alte Seite entfernt und eine neue Seite auf diesen Hintergrund geladen werden.

So etwas kann realisiert werden, indem die Hintergrund-Bitmap bei der Erstellung des Fensterobjekts direkt in das Fenster eingetragen wird. Das Füllmuster für das Fenster sollte auf 'unsichtbar' eingestellt werden. Beim Laden einer neuen Seite in diesem Fenster wird lediglich der Vordergrund gelöscht. Die im Hintergrund eingetragene Bitmap bleibt stehen. Um unliebsame Flackereffekte zu vermeiden, sollten die Befehle zum Laden der Seite mit folgenden Parametern aufgerufen werden:

```
LoadPage Page032 /noclear /paint  
NextPage Page??? /noclear /paint  
PrevPage Page??? /noclear /paint
```

Die Aufteilung in Hinter- und Vordergrund wird erreicht, indem die Ausgaben in getrennten Metafiles aufgezeichnet werden. Dabei wird zwischen dem Page-Metafile und dem Window-Metafile unterschieden.

Den sogenannten Draw-Befehlen (DrawBitmap, DrawLine, DrawText usw.) kann mittels des optionalen Parameters '/record' mitgeteilt werden, ob die Ausgabe in den Hintergrund oder in den Vordergrund eingetragen werden soll. Standardmäßig wird eine solche Ausgabe in den Vordergrund eingetragen.

Scripts

Jedem Fensterobjekt können bis zu sechs Befehlsscripts zugeordnet werden, die bei bestimmten Systemereignissen ausgeführt werden.

1. Open-Script

Wird nach dem Eröffnen eines Fensters aufgerufen.

2. Close-Script

Wird beim Schließen eines Fensters durch das System-Menü aufgerufen. Ist dieses Script vorhanden, muß das Fenster innerhalb des Scripts explizit durch den Befehl Close geschlossen werden.

3. Move-Script

Wird nach dem Verschieben eines Fensters aufgerufen.

4. Resize-Script

Wird nach dem Vergrößern/Verkleinern eines Fensters aufgerufen.

5. Click-Script

Wird nach dem Anklicken eines Fensters mit der Maus aufgerufen..

6. Dclick-Script

Wird nach dem Doppelklicken eines Fensters mit der Maus aufgerufen. Dieses Script kann nur ausgeführt werden, wenn kein Click-Script definiert wurde, da ansonsten beim Anklicken des Fensters grundsätzlich das Click-Script ausgeführt wird.

Dokument

RSE-Author bietet Ihnen die Möglichkeit, Ausgaben auf einen Drucker zu machen. Zu diesem Zweck können Sie in der Entwicklungsumgebung Objekte vom Typ 'Dokument' anlegen. Ein solches Objekt stellt eine Seite dar, die auf den Drucker ausgegeben werden soll. Im Unterschied zu Fenster- und Seitenobjekten können bei einem Dokument einige Elemente nicht verwendet werden. Das erklärt sich dadurch, daß z.B. ein Hypertextfeld auf einer Druckseite nicht viel Sinn macht. Des weiteren werden bei der Bearbeitung eines Dokuments alle Koordinaten in zehntel Millimeter statt wie üblich in Pixeln angegeben. Das ist notwendig, da die Auflösung von Druckern sehr unterschiedlich sein kann. Durch die Verwendung eines auflösungsunabhängigen Koordinatensystems ist gewährleistet, daß der Ausdruck auf allen Druckern und Auflösungen in der gleichen Größe erscheint.

Um etwas auf den Drucker auszugeben, müssen Sie folgendermaßen vorgehen:

1. Erstellen Sie alle Druckseiten, die zu der Ausgabe gehören, als Objekte vom Typ Dokument.
2. Legen Sie in Ihrem Programm ein Bedienungselement an, welches den Druckvorgang auslösen soll, z.B. einen Schalter mit der Aufschrift 'Drucken'.
3. Geben Sie im Click-Script des Bedienungselements die erforderlichen Befehle an, um den Ausdruck zu starten. Das könnte z.B. folgendermaßen aussehen:

```
OpenPrinter prn /option  
prn.LoadDocument Seite01  
prn.PrintPage  
prn.LoadDocument Seite02  
prn.PrintPage  
prn.LoadDocument Seite03  
prn.PrintPage  
prn.Close
```

Der allgemeine Befehl OpenPrinter eröffnet den Drucker. Mit dem Befehl LoadDocument wird eine Druckseite geladen. Der Befehl PrintPage druckt diese Seite aus. Mit dem Befehl Close wird der Drucker wieder geschlossen.

Schalter

Die meisten Programme, die unter einer grafischen Systemoberfläche wie Windows oder OS/2 laufen, werden durch Schalter bedient.

Ein solcher Schalter wird bei der Bearbeitung eines Seiten- oder Fensterobjekts als eigenständiges Element eingesetzt. Das Element erhält einen Namen und auf dieses Element können Befehle ausgeführt werden (siehe [Befehls-Index](#)).

RSE-Author bietet Ihnen drei verschiedene Schaltertypen an:

1. Textschalter

Dieser Schaltertyp hat einen frei definierbaren Text in der Schaltfläche. Die Schriftart kann frei eingestellt werden.

2. Grafikschalter

In diesen Schaltertyp können beliebige Bitmaps eingesetzt werden. Dabei kann für jeden Schalterzustand (nicht gedrückt, gedrückt, gesperrt) eine eigene Bitmap angegeben werden. Die Größe des Schalters ist durch die Abmessungen des Bitmaps vorgegeben.

Der Grafikschalter kann wieder in vier Untertypen unterteilt werden:

2.1. Grafikschalter mit 3D-Rahmen

Die Bitmap wird mit einem 3D-Rahmen versehen.

2.2. Grafikschalter ohne 3D-Rahmen

Die Bitmap wird ohne zusätzlichen Rahmen gezeichnet.

2.3. Grafikschalter ohne 3D-Rahmen, schwarz ist transparent

Die Bitmap wird ohne zusätzlichen Rahmen gezeichnet. Die Farbe 'Schwarz' wird als Transparenzfarbe verwendet. Damit können beliebig geformte Schalter (rund, oval usw.) realisiert werden.

2.4. Grafikschalter ohne 3D-Rahmen, weiß ist transparent

Die Bitmap wird ohne zusätzlichen Rahmen gezeichnet. Die Farbe 'Weiß' wird als Transparenzfarbe verwendet. Damit können beliebig geformte Schalter (rund, oval usw.) realisiert werden.

Bitmaps für Grafikschalter können innerhalb eines Programms mit dem Befehl [LoadBitmap](#) vorgeladen werden. Beim Anlegen eines Schalters während des Aufbaus eines Fensters oder einer Seite prüft das Autorensystem, ob die Bitmap, die auf den Schalter geladen werden soll, bereits im Arbeitsspeicher vorliegt. Ist das der Fall, wird die im Speicher befindliche Bitmap verwendet. Dadurch kann der Aufbau von Schalterelementen beschleunigt werden, indem häufig benötigte Bitmaps bereits beim Programmstart (Open-Handler des Objekts MainWindow) geladen werden.

3. Symbolschalter

In diesen Schaltertyp können vordefinierte Grafiksymbole (Pfeile) eingesetzt werden.

Script

Einem Schalter kann ein Befehls-Script zugeordnet werden, das beim Anklicken des Schalters ausgeführt werden soll.

Hyperlink

Ein Hyperlink (oder auch maussensitives Feld) ist ein beliebiger rechteckiger Bereich in einem Fenster. Durch Anklicken oder Hinein/Herausbewegen der Maus in diesem Bereich kann eine Aktion ausgelöst werden.

Ein derartiges maussensitives Feld wird bei der Bearbeitung eines Seiten- oder Fensterobjekts als eigenständiges Element eingesetzt. Das Element erhält einen Namen und auf dieses Element können Befehle ausgeführt werden (siehe [Befehls-Index](#)).

Bei der Bearbeitung von Bitmap-Elementen können Sie das Bitmap ebenfalls als Hyperlink-Feld markieren. In dem Fall wird beim Zeichnen des Bitmaps vom Laufzeitsystem automatisch um das Bitmap herum ein Hyperlink-Feld erzeugt. Diese Möglichkeit erspart beim Entwickeln etwas Arbeit, da es eine häufige Anforderung ist, kleine Grafiken (Thumbnails) mit Querverweisen zu anderen Bildschirmseiten zu versehen.

Scripts

Dem Feld können insgesamt vier verschiedene Befehls-Scripts zugeordnet werden.

1. Click-Script

Wird beim einfachen Anklicken des Feldes mit der Maus aufgerufen. In diesem Script kann dann z.B. mit dem Befehl [LoadPage](#) eine andere Bildschirmseite geladen werden.

2. Dclick-Script

Wird beim Doppelklicken des Feldes mit der Maus aufgerufen. Dieses Script kann nur ausgeführt werden, wenn kein Click-Script definiert wurde, da ansonsten beim Anklicken des Feldes grundsätzlich das Click-Script ausgeführt wird.

3. MoveIn-Script

Wird beim Hineinbewegen des Mauszeigers in ein Feld aufgerufen. Hier kann z.B. ein Text in einer Statuszeile ausgegeben werden, der das Objekt beschreibt, über den sich der Mauszeiger gerade bewegt. Des weiteren kann mit dem Befehl [Mark](#) das maussensitive Feld optisch hervorgehoben werden.

4. MoveOut-Script

Wird beim Herausbewegen des Mauszeigers aus einem Feld aufgerufen. Wurde im MoveIn-Script ein Text in eine Statuszeile geschrieben, kann dieser Text jetzt wieder gelöscht werden. Falls das Feld über den Befehl [Mark](#) optisch hervorgehoben wurde, kann die Hervorhebung mit dem Befehl [Unmark](#) wieder rückgängig gemacht werden.

Hypertext

Ein Hypertextfeld ist ein durch eine andere Farbe oder Schriftattribut hervorgehobenes Wort in einem Text. Durch Anklicken dieses Wortes kann eine Aktion ausgelöst werden, z.B. das Laden einer weiteren Bildschirmseite mit zusätzlichen Informationen zu diesem speziellen Begriff.

Hypertextfelder können bei der Bearbeitung eines Textelements erzeugt werden, indem das als Hypertext zu markierende Wort in eckige Klammern ([.....]) gesetzt wird. Klickt man in dem Bearbeitungsfenster für Textelemente dann den Schalter 'Aktualisieren' an, werden in einer Liste alle im Text markierten Hypertextfelder aufgelistet. Jedes dieser Hypertextfelder kann dann als eigenständiges Element bearbeitet werden. Das Element erhält einen Namen und auf dieses Element können Befehle ausgeführt werden (siehe Befehls-Index).

Scripts

Einem Hypertextfeld können insgesamt vier verschiedene Befehls-Scripts zugeordnet werden.

1. Click-Script

Wird beim einfachen Anklicken des Hypertextfeldes mit der Maus aufgerufen. In diesem Script kann dann z.B. mit dem Befehl LoadPage eine andere Bildschirmseite mit Informationen zu diesem Begriff geladen werden.

2. Dclick-Script

Wird beim Doppelklicken des Hypertextfeldes mit der Maus aufgerufen. Dieses Script kann nur ausgeführt werden, wenn kein Click-Script definiert wurde, da ansonsten beim Anklicken des Hypertextfeldes grundsätzlich das Click-Script ausgeführt wird.

3. MoveIn-Script

Wird beim Hineinbewegen des Mauszeigers in ein Hypertextfeld aufgerufen. Hier kann z.B. ein Text in einer Statuszeile ausgegeben werden, der eine Kurzinformation zu dem Wort ausgibt, über den sich der Mauszeiger gerade bewegt. Des weiteren kann mit dem Befehl Mark das Hypertextfeld optisch hervorgehoben werden.

4. MoveOut-Script

Wird beim Herausbewegen des Mauszeigers aus dem Hypertextfeld aufgerufen. Wurde im MoveIn-Script ein Text in eine Statuszeile geschrieben, kann dieser Text jetzt wieder gelöscht werden. Falls das Hypertextfeld über den Befehl Mark optisch hervorgehoben wurde, kann die Hervorhebung mit dem Befehl Unmark wieder rückgängig gemacht werden.

ListBox

Eine ListBox ist ein kleines Fenster, das Texteinträge in einer Liste anzeigt. Durch Selektion oder Doppelklicken dieser Einträge können Programmaktionen ausgelöst werden.

Eine ListBox wird bei der Bearbeitung eines Seiten- oder Fensterobjekts als eigenständiges Element eingesetzt. Das Element erhält einen Namen und auf dieses Element können Befehle ausgeführt werden (siehe [Befehls-Index](#)).

Bei der Bearbeitung der ListBox können beliebig viele Texteinträge eingefügt werden. Jeder einzelne Eintrag kann eigene Befehls-Scripts erhalten. Wird im Bearbeitungsdialog der Script-Schalter angeklickt, dann wird ein Editorfenster für den aktuell selektierten Listeneintrag aufgeblendet.

Scripts

Einer ListBox können insgesamt zwei verschiedene Befehls-Scripts zugeordnet werden.

1. Select-Script

Wird beim Selektieren eines Eintrags mit der Maus oder mit den Pfeiltasten aufgerufen. Hier kann z.B. ein Text in einer Statuszeile ausgegeben werden, der eine Kurzinformation zu dem Eintrag ausgibt,

2. Dclick-Script

Wird beim Doppelklicken eines Eintrags mit der Maus aufgerufen. In diesem Script kann dann z.B. mit dem Befehl [LoadPage](#) eine andere Bildschirmseite mit Informationen zu dem Eintrag geladen werden.

Bitmaps

Bitmaps sind eines der wichtigsten Gestaltungselemente für Präsentationsprogramme, wenn es darum geht, Informationen visuell zu vermitteln.

Bitmaps können mittels des Befehls LoadBitmap als eigenständige Objekte in den Speicher geladen werden. Von dort können Sie mittels des Befehls DrawBitmap beliebig oft in Bildschirmfenster ausgegeben werden. Mit dem Befehl Delete können Sie wieder aus dem Speicher entfernt werden.

Beachten Sie bitte auch den Abschnitt Bitmaps im Unterkapitel Tips und Tricks. Hier erhalten Sie zusätzliche Informationen über den Umgang mit Bitmaps.

Video

RSE-Author bietet drei verschiedene Möglichkeiten, Video-Clips in Programme einzubetten.

1. Der Befehl VideoPlayer eröffnet ein Abspielfenster für Video-Dateien. Dieses Fenster beinhaltet einen kleinen Bildschirm, in dem das Video abläuft und eine Schalterleiste mit Funktionen wie Play, Stop, Pause, Rewind, Forward usw. Der Anwender hat die Möglichkeit, sich das Video beliebig oft anzusehen und nach Belieben vor- und zurückzuspulen. Dieser Befehl sollte für AVI- und MPG-Dateien verwendet werden. Für Autodesk-Animator-Dateien (FLI, FLC) ist er nur bedingt geeignet.
2. Der Befehl PlayVideo lässt eine Video-Datei in einem beliebigen Fenster ablaufen.
3. Über den Befehl Mci können MCI-Befehle direkt an das Betriebssystem übergeben werden. Den Aufbau und die Syntax dieser Befehle können Sie der 'Video für Windows'-Dokumentation entnehmen.

Beachten Sie bitte auch den Abschnitt Video im Unterkapitel Tips und Tricks. Hier erhalten Sie zusätzliche Informationen über den Umgang mit digitalem Video.

Audio

RSE-Author bietet zwei verschiedene Möglichkeiten, digitales Audio in Programme einzubetten.

1. Der Befehl PlayAudio spielt eine beliebige Audio-Datei ab.
2. Über den Befehl Mci können MCI-Befehle direkt an das Betriebssystem übergeben werden. Den Aufbau und die Syntax dieser Befehle können Sie der 'Video für Windows'-Dokumentation entnehmen.

Beachten Sie bitte auch den Abschnitt Audio im Unterkapitel Tips und Tricks. Hier erhalten Sie zusätzliche Informationen über den Umgang mit digitalem Audio.

Ein erstes Programm

Im folgenden Abschnitt werden alle Schritte erläutert, die erforderlich sind, um ein erstes lauffähiges Programm mit dem Autorensystem RSE-Author zu erstellen.

Wir wollen ein kleines Programm erstellen, das es erlaubt, 10 verschiedene Seiten innerhalb eines Fensters durchzublättern. Das Hauptfenster soll aus folgenden Elementen bestehen:

- Eine Titelleiste, in welcher der Titel der aktuellen Seite angezeigt wird.
- Ein Seitenfenster, in dem die Seiten durchgeblättert werden.
- Eine Schalterleiste, in der alle zur Bedienung des Programms benötigten Schalter sitzen.

Beim Programmstart soll ein Hauptindex angezeigt werden, der über Hyperlinkfelder einen wahlfreien Zugriff auf die 10 Seiten erlaubt. Außerdem soll ein sequentielles Blättern zwischen den Seiten durch Anklicken der entsprechenden Schalter möglich sein.

1. Objekte anlegen

Legen Sie in einem neuen Programm zusätzlich zu dem bereits vorhandenen Objekt 'MainWindow' noch die folgenden Objekte an:

- Ein Fensterobjekt namens 'TitleWindow'
- Ein Fensterobjekt namens 'PageWindow'
- Ein Seitenobjekt namens 'MainIndex'
- 10 Seitenobjekte namens 'Page001', 'Page002',, 'Page010'
- Ein Fensterobjekt namens 'CopyrightInfo'

2. Bearbeiten des Objekts 'TitleWindow'

Rufen Sie den Bearbeitungsdiallog dieses Fensters auf. Setzen Sie das Feld 'Typ' auf den Wert 'Ohne Rahmen'. Das Fenster ist also als rahmenloses Kindfenster definiert.

Legen Sie dann in dem Fenster ein Text-Element an. In dem Bearbeitungs-Dialog des Text-Elements geben Sie als Text einfach nur einen Punkt an. Es geht hier nur darum, die Schriftart des Titelfensters zu definieren. Das tun wir, indem wir einen Dummy-Text einsetzen. Rufen Sie jetzt das Bearbeitungsfenster für die Schriftart ein und stellen Sie dort die Schrift ein, die für Ihre Überschriften verwendet werden soll.

3. Bearbeiten des Objekts 'PageWindow'

Rufen Sie den Bearbeitungsdiallog dieses Fensters auf. Setzen Sie das Feld 'Typ' ebenfalls auf den Wert 'Ohne Rahmen'.

4. Bearbeiten des Objekts 'MainWindow'

Jetzt können Sie darangehen, das Hauptfenster zu bearbeiten. Definieren Sie das Fensterobjekt innerhalb des Eigenschaften-Dialogs als Standardfenster. Stellen Sie hier außerdem ein, daß das Fenster nicht in der Größe veränderbar sein darf.

Legen Sie innerhalb des Fensters Schalter-Elemente an für die folgenden Funktionen:

- Zurückblättern
- Vorwärtsblättern
- Hauptindex aufrufen

- Copyright-Information aufrufen
- Programm beenden

Schalten Sie das Bearbeitungsfenster am besten in den Vollbildmodus, indem Sie die Leertaste drücken. Dadurch haben Sie die bearbeitete Fensterfläche besser im Überblick. Ein erneutes Drücken der Leertaste schaltet das Bearbeitungsfenster wieder in die normale Darstellung zurück (falls das nicht klappt, klicken Sie einfach mit der Maus in die freie Fensterfläche und versuchen Sie es noch einmal).

Ordnen Sie die Schalter im unteren Bereich von MainWindow nach Ihrem Belieben an. Welche Schaltertypen Sie verwenden und wie Sie diese optisch gestalten, bleibt Ihnen überlassen. Falls Sie die Schalterleiste farblich vom Rest des Fensters abheben wollen, legen Sie einfach ein Rechteck-Element hinter die Schalter.

Geben Sie jetzt die Befehls-Scripts für die Schalter ein.

Schalter für Zurückblättern

PageWindow.PrevPage Page??? /loop

Schalter für Vorwärtsblättern

PageWindow.NextPage Page??? /loop

Schalter für Hauptindex aufrufen

PageWindow.LoadPage Page??? /loop

Schalter für Copyright-Information aufrufen

LoadWindow CopyrightInfo

Schalter für Programm beenden

Quit

Legen Sie jetzt innerhalb von MainWindow zwei Fenster-Elemente an. Definieren Sie innerhalb dieser Elemente Querverweise auf das Objekt 'TitleWindow' und das Objekt 'PageWindow'.

Wenn Sie das wünschen, können Sie die beiden Unterfenster innerhalb des Bearbeitungsdialogs noch mit Rahmen versehen. Positionieren Sie diese Elemente (Titelleiste und das Seitenfenster), wie Sie es für richtig halten. Die beiden Unterfenster sollten natürlich nicht die Schalterleiste verdecken.

Geben Sie zum Schluß innerhalb des Eigenschaften-Dialogs des Objekts 'MainWindow' noch das Befehls-Script ein, das beim Eröffnen des Hauptfensters ausgeführt werden soll (Open-Script):

PageWindow.LoadPage MainIndex ! Hauptindex bei Programmstart laden

5. Bearbeiten des Hauptindex

Bearbeiten Sie jetzt das Seitenobjekt namens 'MainIndex'. Setzen Sie hier für jede der 10 Seiten entweder ein Hypertext- oder ein Hyperlinkfeld ein, durch dessen Anklicken später zu dieser Seite verzweigt werden soll. Ein Hypertextfeld könnte z.B. den Titel der Seite enthalten. Ein Hyperlinkfeld könnte aus einer kleinen Grafik bestehen, welche das Thema der Seite anzeigt.

Geben Sie im Click-Script der Hypertext- oder Hyperlinkfelder abhängig von der

Seitenbezeichnung folgenden Befehl ein:

```
PageWindow.LoadPage Page001
```

Setzen Sie die jeweils passende Seitenbezeichnung ein.

Geben Sie jetzt noch im Eigenschaften-Dialog des Seitenobjekts das PageEntry-Script ein, das beim Laden des Hauptindex ausgeführt werden soll:

```
TitleWindow.Clear  
TitleWindow.DrawText "Haupt-Index" /x:5
```

6. Bearbeiten der 10 Seiten

Geben Sie jetzt innerhalb dieser 10 Seiten Texte und Grafiken nach Ihrem Belieben ein.

Geben Sie im Eigenschaften-Dialog des Seitenobjekts das PageEntry-Script ein, das beim Laden dieser Seite ausgeführt werden soll:

```
TitleWindow.Clear  
TitleWindow.DrawText "Das hier ist die Seite ??? " /x:5
```

7. Bearbeiten der Copyright-Information

Dieses Fenster sollten Sie im Eigenschaften-Dialog des Objekts als 3D-Dialogfenster definieren. Fügen Sie in das Objekt einige Text-Elemente ein und geben Sie in diese Elemente eine Copyright-Meldung ein. Des weiteren wird noch ein OK-Schalter benötigt.

Für den OK-Schalter geben Sie als Befehls-Script folgenden Befehl ein:

```
Close
```

Zum Abschluß

Wenn Sie diese sieben Schritte durchführen, sollten Sie in der Lage sein, Ihr erstes RSE-Author-Programm zu schreiben und ein Gefühl für die Funktionsweise dieses Entwicklungssystems zu bekommen. Sehen Sie sich auch die mitgelieferten Demo-Programme an.

Performance verbessern

Im folgenden wollen wir Ihnen einige Hinweise geben, wie Sie die Ablaufgeschwindigkeit Ihres Programms verbessern können.

Zuerst noch einige allgemeine Hinweise:

Wenn Sie eine Applikation entwickeln, die später von einer CD laufen soll, behalten Sie immer im Auge, daß Daten und Grafiken von einer CD wesentlich langsamer geladen werden, als von einer Festplatte. Dieser Aspekt wird allzuleicht übersehen, wenn ein Programm entwickelt wird. Die Applikation, die in Verbindung mit einer schnellen SCSI-Platte ein absoluter Renner war, entpuppt sich dann leicht als Schnecke, wenn sie von CD ausgeführt wird.

Bei manchen Entwicklungssystemen verlangsamt die Ausführung der Befehls-Scripts die Ausführung der Programme erheblich. Unter RSE-Author müssen Sie sich darüber keine Gedanken machen. Bei der Geschwindigkeit von modernen Rechnern (ab 486er) kann der Zeitaufwand für das Interpretieren der Befehls-Scripts vernachlässigt werden. Sie können die Ablaufgeschwindigkeit der Scripts steigern, indem Sie auf Kommentare in den Scripts verzichten. Wir empfehlen aber dennoch, die Scripts gut zu kommentieren, da der Geschwindigkeitsgewinn so gering ausfällt, daß es nicht lohnt, deswegen auf die Übersichtlichkeit zu verzichten.

Bitmaps

Der Zeitaufwand für das Nachladen von Grafiken ist am stärksten für die Performance eines Programms verantwortlich. Um kurze Ladezeiten zu erreichen, sollten die folgenden Punkte beachtet werden:

- 1.** In der Regel können unkomprimierte Grafiken (BMP, unkomprimiertes TIF) schneller geladen werden, als komprimierte Grafiken (GIF, PCX, JPEG, komprimiertes TIF). Wenn möglich, sollten Sie deshalb das BMP-Format verwenden. Das JPEG-Format sollte nur in Verbindung mit Echtfarb-Bildern (16 Millionen Farben) und wenn es darauf ankommt, möglichst viele Bilder auf einem Datenträger unterzubringen (Bildarchive), verwendet werden.
- 2.** Es gilt generell, daß das Laden von vielen kleinen Grafiken sehr viel zeitaufwendiger ist, als das Laden einer großen Grafik. Sie sollten deshalb kleine Grafiken in einer großen Bitmap zusammenfassen, die dann mit einem einzigen Zugriff geladen wird.
- 3.** Selbst wenn Ihr Programm grundsätzlich auf Rechnern, die mit 64K oder 16 Mio Farben konfiguriert sind, laufen soll, sollten Sie prüfen, ob für Ihre Anwendung Grafiken mit 256 Farben ausreichend sind. Eine Grafik mit 256 Farben benötigt nur ein Drittel des Speicherplatzes, den ein Echtfarbbild mit 16 Mio Farben benötigt und kann dementsprechend schneller geladen werden.
- 4.** Das Skalieren von Bitmaps, sowie die transparente Verknüpfung von Bitmaps mit dem Hintergrund können ein Programm auch verlangsamen. Vor allem das Skalieren/Verknüpfen von sehr großflächigen Bitmaps kann die Ablaufgeschwindigkeit eines Programms drastisch herabsetzen. Deshalb sollten diese Funktionen nur sparsam eingesetzt werden. Eine Transparenzverknüpfung sollte, wenn möglich, nur mit kleinen Bitmaps durchgeführt werden. Statt eine Bitmap zur Laufzeit zu skalieren, sollte man prüfen, ob es nicht sinnvoller ist, die Bitmap bereits während der Entwicklung eines Programms mit einem Bildbearbeitungsprogramm in die gewünschte Größe zu bringen und in diesem Format abzuspeichern. Durch den großen Speicherplatz den eine CD bietet,

ist es ohne weiteres möglich, die gleiche Bitmap in verschiedenen Auflösungsstufen mitzuliefern.

5. Wenn Sie eine neue Seite laden, die auch Bitmaps enthält, sollten Sie den optionalen Parameter '/back' angeben. Die Seite wird dann im Arbeitsspeicher aufgebaut und anschließend in das Fenster kopiert. Das ist zwar nicht schneller, als das direkte Laden in das Fenster, sogar ein wenig langsamer, da aber während der Ladezeit die vorige Seite noch sichtbar ist, wirkt diese Vorgehensweise subjektiv schneller.

Video

Beim Starten einer Video-Datei vergehen meistens einige Sekunden, bis das erste Bild erscheint. Diese Zeit kann abgekürzt werden, indem Sie beim Start Ihres Programms (im Open-Handler des MainWindow) mittels des Befehls Mci eine kleine Dummy-Videodatei eröffnen. Dadurch werden die Programmkomponenten des MCI-Systems bereits in den Speicher geladen und stehen beim Starten einer anderen Videodatei bereits zur Verfügung. Beim Programmende müssen Sie die Dummy-Videodatei dann wieder schließen. Die Befehle könnten wie folgt aussehen:

Programmstart

```
Mci "open dummy.avi alias xxx"
```

Programmende

```
Mci "close xxx"
```

Tips und Tricks

Ein Autorensystem allein genügt noch nicht, um ein anspruchsvolles Multimedia-Programm zu erstellen. Woher bekommt man z.B. gutes Bildmaterial für das geplante Programm? Wie erstellt man eine Video-Datei? Oder wie vermeidet man Farbfehler, wenn man ein Programm für 256-Farb-Adapter erstellt?

Diese und einige andere Fragen möchten wir Ihnen in diesem Abschnitt beantworten. Wir greifen dabei auf die kleinen Tips und Tricks zurück, die sich in unserer eigenen täglichen Entwicklungsarbeit bewährt haben.

[Planung](#)

[Copyright](#)

[Bitmaps](#)

[Video](#)

[Audio](#)

[CD-ROM](#)

[Installation](#)

Planung

Den Aufwand für die Entwicklung eines wirklich guten Multimedia-Programms sollte man nicht unterschätzen. Informationen müssen recherchiert werden. Das Bild- und Videomaterial muß produziert und nachbearbeitet werden. Texte müssen geschrieben werden. Die Programmierung an sich nimmt verglichen damit eher wenig Zeit in Anspruch. Folgende Vorgehensweise hat sich bei der Entwicklung von Multimedia-Programmen bewährt.

- 1.** Erstellung eines Pflichtenheftes. Überlegen Sie genau, was Ihr Programm leisten soll und was alles darin enthalten sein soll. Legen Sie bereits jetzt eine Struktur für Ihr Programm fest. Schreiben Sie auf, welches Quellenmaterial Sie benötigen und wo Sie es bekommen können. Überlegen Sie auch, ob Sie das Bildmaterial selber produzieren oder einkaufen wollen und wie lange Sie dafür benötigen werden. Legen Sie einen Zeitplan für die verschiedenen Schritte der Entwicklung fest.
- 2.** Recherchieren Sie die benötigten Informationen für Ihr Programm.
- 3.** Beschaffen oder produzieren Sie das Bild- und Videomaterial.
- 4.** Schreiben Sie die dazugehörigen Texte.
- 5.** Zum Schluß kann die Programmierung erfolgen, welche das gesammelte Material in einem Programm zusammenfaßt.

Wenn Sie bei der Entwicklung Ihrer Applikationen methodisch vorgehen und sorgfältig planen, werden Sie schnell zu ansehnlichen Ergebnissen kommen. Dennoch gilt, daß die Entwicklung eines guten Programms viel Zeit und auch Geld in Anspruch nimmt.

Copyright

Bei der Beschaffung des Bildmaterials für Ihr Programm sollten Sie sorgfältig darauf achten, daß Sie nicht das Copyright von jemand anderem verletzen. Die Fotos und Videosequenzen, die auf CD-Bildersammlungen zu finden sind, sind fast immer Copyright-geschützt, d.h. Sie dürfen diese Bilder und Videos nicht für Ihre Zwecke verwenden. Es gibt zwar lizenzfreie Bildersammlungen, die Erlaubnis für die Weitergabe beschränkt sich aber meistens auf die Erzeugung von gedruckten Publikationen. Die Weitergabe als digitale Datei ist auch bei diesen Sammlungen oftmals nicht erlaubt. Erkundigen Sie sich also vor der Verwendung solchen Materials genau, ob die Verwendung zu dem von Ihnen geplanten Zweck auch wirklich erlaubt ist.

Das sicherste ist immer, Bild- und Videomaterial selbst zu produzieren. Das Copyright daran gehört dann Ihnen und Sie sind sicher, daß es keinen Ärger gibt.

Beherrigen Sie diesen Ratschlag bitte gut. Es sind schon ganze CD-ROM-Auflagen mit mehreren tausend Stück eingestampft worden, weil die Frage des Copyrights mißachtet wurde.

Bitmaps

In Multimedia-Programmen sind Bitmap-Grafiken das am häufigsten gebrauchte Medium. Woher bekommt man nun die Bitmaps für das geplante Programm und wie bearbeitet man sie am besten?

Die erste und einfachste Möglichkeit ist natürlich, Bitmaps mit einem Malprogramm selbst zu zeichnen, vorausgesetzt man kann zeichnen oder hat einen Grafiker zur Hand. Geeignete Grafikprogramme für den PC gibt es wie Sand am Meer. Das bekannteste ist sicherlich PC-Paintbrush, das standardmäßig im Lieferumfang von Windows enthalten ist.

Die zweite Möglichkeit besteht darin, mit einer Kamera Aufnahmen zu machen und diese in den Computer zu übertragen. Man kann dies entweder mit einer digitalen Kamera machen, welche die Aufnahmen direkt in einem computerlesbaren Format erzeugt, oder mit einer normalen Kleinbildkamera, deren Aufnahmen man später digitalisieren läßt (Photo-CD).

Meistens wird das Bildmaterial aus nachträglich digitalisierten Fotos bestehen, die dann für Dokumentationen, Kataloge oder elektronische Bücher verwendet werden.

Photo-CD

Die einfachste und qualitativ beste Möglichkeit, um das benötigte Bildmaterial zu erhalten, ist die Kodak-Photo-CD. Eine CD mit ca. 100 digitalisierten Aufnahmen kostet etwa 120 DM (Stand 12.94). Die Qualität dieser Bilder ist erfahrungsgemäß hervorragend.

Die Bilder können mit einer einfachen Kleinbildkamera gemacht werden. Nach dem Entwickeln des Filmes sucht man die Bilder aus, die digitalisiert werden sollen und gibt die Negative oder Dias einfach beim nächsten Photo-Fachhändler ab. Nach ca. 2 Wochen erhält man dann eine CD mit den gewünschten Aufnahmen zurück (es gibt aber auch Firmen, die einen Schnell-Service anbieten). Die Bilder können direkt von der CD geladen und mit einem Bildbearbeitungsprogramm in die gewünschte Form gebracht werden (nachscharfen, Größe verändern, Farben reduzieren usw.).

Bildbearbeitung

Ein Bildbearbeitungsprogramm, das sich bei uns in der täglichen Arbeit bewährt hat, ist TIFFY-PRO von Helge Hackbarth. Es handelt sich hierbei um ein einfach zu bedienendes und sehr preiswertes Programm, das alle wichtigen Funktionen für die Bildbearbeitung enthält. Die Bitmaps für das beiliegende Demo-Programm wurden z.B. mit TIFFY-PRO aus den Photo-CD-Vorlagen erzeugt.

Das Programm kann bei folgender Adresse bezogen werden:

**Helge Hackbarth
Hellgrund 84
D-22880 Wedel**

**Tel.: (040) 880 16 29
Fax: (040) 881 38 64
BBS: (040) 881 38 64 (von 18.00 bis 20.00 Uhr)**

Eine Shareware-Version von TIFFY-PRO ist im Lieferumfang der RSE-Author-CD enthalten.

Geschwindigkeit

Die Performance Ihrer Anwendung hängt in sehr hohem Maße davon ab, welches Bildformat und welche Farbtiefe Sie verwenden.

Wenn Speicherplatz keine oder nur eine geringe Rolle spielt (z.B. bei CD-ROM-Anwendungen), sollten Sie das BMP-Format verwenden, da dieses am schnellsten geladen werden kann und somit die Performance Ihrer Anwendung verbessert wird. Wenn möglich, sollten Sie die Bilder auf 256 Farben reduzieren, da eine Bitmap-Grafik mit dieser Farbtiefe nur ein Drittel des Speicherplatzes benötigt, den ein Echtfarbbild belegt. Das spart nicht nur Speicherplatz, sondern beschleunigt Ihr Programm auch erheblich, da das Laden der Bilddaten wesentlich schneller vonstatten geht.

Selbst wenn ein Programm nur im HighColour- oder TrueColour-Modus laufen soll, kann es sinnvoll sein, wegen der besseren Performance mit 256-Farb-Grafiken zu arbeiten.

Sie sollten außerdem vermeiden, viele kleine Bitmaps auf eine Bildschirmseite zu laden, da dadurch die Performance Ihres Programms ebenfalls schlechter wird. Fassen Sie die kleinen Bitmaps lieber zu einer großen Bitmap zusammen, die auf einen Schlag geladen wird.

Ein Beispiel: Sie wollen ein Programm schreiben, bei dem aus einem Index mit vielen kleinen Bildern zu weiteren Bildschirmseiten verzweigt wird. Dieser Index besteht aus 20 kleinen Bitmaps. Nehmen wir des weiteren an, das Programm soll von CD laufen. Ein CD-Laufwerk hat eine mittlere Zugriffszeit von ca. 300 Millisekunden. Um den Index mit 20 Mini-Bitmaps zu laden, werden also allein für die 20 Positionierungen des Lasers etwa 6 Sekunden benötigt. Dazu kommt noch die reine Ladezeit der Bitmap-Daten. Faßt man die kleinen Bitmaps aber in eine große Bitmap zusammen, dann hat man nur noch eine Positionierungszeit von einer Drittel Sekunde. Die große Bitmap wird also wesentlich schneller geladen.

Farbtiefe

Soll ein Programm eine möglichst optimale Bildqualität haben, sollten Sie Echtfarbbilder mit 16 Millionen Farben verwenden. Eine solche Applikation stellt aber hohe Ansprüche an die Hardware. Für viele Zwecke sind 256 Farben voll ausreichend.

Wenn Ihr Bildmaterial auf Photo-CD vorliegt, müssen Sie die Bilder mit einem Bildbearbeitungsprogramm auf 256 Farben reduzieren. Beachten Sie bitte, daß viele Programme nur sehr einfache Funktionen zur Farbreduktion anbieten, die nur sehr mäßige Resultate erzielen.

Ein Programm, das in dieser Hinsicht sehr gute Ergebnisse liefert, ist das weiter oben erwähnte TIFFY PRO. Sehen Sie sich auch hierzu das mitgelieferte Demo-Programm an, das sehr anschaulich demonstriert, welche Qualität mit 256 Farben erreicht werden kann.

Farbpaletten

Ein weiteres Problem, mit dem sich ein Entwickler von Multimedia-Programmen beschäftigen muß, sind Farbpaletten. Es setzen sich zwar immer mehr Grafikkarten mit High-Colour - oder TrueColour-Modi durch, aber der kleinste gemeinsame Nenner sind derzeit immer noch Grafikadapter, die gleichzeitig 256 Farben darstellen

Auch mit 256 Farben kann eine sehr gute Bildqualität erreicht werden, wenn man einige Regeln beachtet. Sehen Sie sich z.B. das beiliegende Demo-Programm an, das eine kleine Diashow zu einer Australien-Reise darstellt. Trotz der Beschränkung auf 256 Farben wurde hier eine gute Qualität erreicht.

Außerdem sollte man beachten, daß die Verwendung von 256-Farb-Grafiken die Performance eines Programms verbessert, da gegenüber Echtfarb-Bildern nur ein Drittel der Bilddaten geladen werden muß.

Wenn auf dem Bildschirm aber nur 256 Farben dargestellt werden können, kann es zu Farbveränderungen kommen, wenn mehrere Bitmaps mit 256 Farben gleichzeitig dargestellt werden sollen. Jede Bitmap hat eine eigene Farbpalette, in der die Farben, welche diese Bitmap verwendet, definiert werden. Wird eine Bitmap neu geladen, wird gleichzeitig deren Farbpalette eingestellt. Wenn sich diese Palette von den Paletten der zuvor geladenen Bitmaps unterscheidet, dann kommt es zu Farbveränderungen.

Durch einige Tricks kann man diesen Effekt vermeiden. Folgende Möglichkeiten stehen zur Verfügung:

- 1.** Die Programme laufen nur in einem HighColor- oder TrueColor-Modus. Da hier 64K oder sogar 16.7 Mio Farben dargestellt werden können, werden in diesen Modi Paletten sowieso ignoriert. Alle Bilder können farbecht geladen werden.
- 2.** Es ist immer nur eine palettenbehaftete Bitmap auf dem Bildschirm zu sehen. Bei einfachen Slide-Shows, bei denen Bild für Bild durchgeblättert wird, ist das durchaus ausreichend. Vor dem Laden der nächsten Bitmap wird der Bildschirm einfach gelöscht, so daß eine andere Palette realisiert werden kann.
- 3.** Sie erzeugen eine optimierte Farbpalette für mehrere Bitmaps. Sie benötigen dazu ein Bildbearbeitungsprogramm, das Farbreduktionen durchführen und die Palette einer Bitmap als externe Datei speichern kann. Das Programm muß des weiteren dazu in der Lage sein, eine Farbreduktion auf eine externe Palette durchzuführen. Ein Programm, das diese Anforderungen erfüllt, ist z.B. TIFFY PRO.
 - 3.1.** Legen Sie eine TrueColor-Grafik (24 Bit) an, die groß genug ist, um alle Bitmaps, die angepaßt werden sollen, aufzunehmen.
 - 3.2.** Fügen Sie die Bitmaps, die aneinander angepaßt werden sollen, über die Zwischenablage in die TrueColor-Grafik ein. Danach verfügen Sie über eine Grafik, in der alle Einzelgrafiken zusammengefaßt sind.
 - 3.3.** Führen Sie mit einem Bildbearbeitungsprogramm eine Farbreduktion auf die TrueColor-Grafik durch. Speichern Sie die erzeugte Farbpalette ab. Die TrueColor-Grafik wird danach nicht mehr benötigt und kann gelöscht werden.
 - 3.4.** Jetzt müssen Sie die Bitmaps, die angepaßt werden sollen, nacheinander einzeln laden. Zur Durchführung des nächsten Schrittes müssen auch diese Einzelgrafiken im TrueColor-Format (24 Bit) vorliegen. Ist das nicht der Fall, wandeln Sie die Bitmaps mit einem Bildbearbeitungsprogramm in dieses Format um.
 - 3.5.** Führen Sie dann für jede Bitmap eine Farbreduktion auf 256 Farben durch, wobei Sie die Bitmap auf die zuvor extern abgespeicherte Palette mappen. Speichern Sie die Bitmaps danach erneut ab.

Nach der Durchführung dieser fünf Schritte haben Sie mehrere Bitmaps mit einer optimalen einheitlichen Farbpalette.

Wenn Sie zusätzlich noch Videos ablaufen lassen wollen, müssen Sie noch einige zusätzliche Arbeitsschritte durchführen. Sie benötigen dazu einen Video-Editor, der es erlaubt, Einzelbilder aus einer Video-Datei zu extrahieren, sowie die Palette der Video-Datei zu

bearbeiten. Wenn Sie mit AVI-Dateien arbeiten, bietet z.B. 'Video für Windows' von Microsoft die erforderlichen Funktionen an.

- 3.2.1** Extrahieren Sie von jeder Szene der Videosequenz ein Einzelbild als Bitmap mit Palette. Fügen Sie diese Bitmaps zusätzlich in die TrueColor-Grafik ein.

- 3.6.** Nachdem Sie die Einzelbilder aus der Videosequenz ebenfalls mit der optimierten Palette versehen haben, übertragen Sie die Palette aus einem der Einzelbilder in die Videosequenz. Die Videosequenz verfügt danach über die gleiche Farbpalette wie die Bitmaps, die gleichzeitig auf den Bildschirm kommen sollen.

Video

Erst durch die Möglichkeit, Programme mit Videosequenzen auszustatten, ist Multimedia wirklich salonfähig geworden. Die Aussagekraft der laufenden Bilder ist sehr viel größer als die von statischen Standbildern.

Allerdings gibt es für die Darstellung von Video auf dem Computer-Bildschirm eine schier unüberschaubare Vielfalt an Hardware- und Software-Lösungen. Wir möchten Ihnen hier kurz die wichtigsten Grundlagen vermitteln.

Lösungen zur Video-Darstellung auf dem Computer müssen grundsätzlich zwei verschiedenen Hauptgruppen zugeordnet werden.

Analoges Video

Diese Form der Video-Darstellung war bis vor wenigen Jahren die einzige Möglichkeit, Video in Computer-Programme zu integrieren. In den Computer wird zu diesem Zweck eine sogenannte Video-Overlay-Karte eingebaut. An diese Karte wird eine analoge Videoquelle, wie z.B. ein Video-Recorder oder ein Laserdisc-Player angeschlossen. Die Overlay-Karte mischt dann das Signal der Grafikkarte mit dem analogen Videosignal zusammen und stellt das Ergebnis auf dem Monitor dar. Normalerweise wird dabei einfach eine bestimmte Farbe (i.d.R. blau oder schwarz) als transparent definiert. Überall, wo auf dem Computerbild diese Farbe auftaucht, ist dann das Videobild zu sehen.

Der Vorteil dieser Methode besteht in einer sehr guten Bildqualität verbunden mit der Möglichkeit Video bildschirmfüllend und ruckfrei abzuspielen. Für High-End-Anwendungen, die eine maximale Qualität bieten sollen, ist diese Methode am besten geeignet.

Der Nachteil liegt in den sehr hohen Kosten für ein derartiges Multimedia-System, da zusätzlich zum Computer noch eine Video-Overlay-Karte und ein geeigneter Video-Recorder oder Laserdisc-Player angeschafft werden muß. Des weiteren ist die Produktion von entsprechenden Videobändern oder Bildplatten sehr teuer.

Eine Ausnahme hierzu stellen TV-Overlay-Karten dar, die ein Fernsehsignal mit dem Grafikkartensignal zusammenmischen, da die analoge Bildquelle hier sehr preiswert aus der heimischen Antennenbuchse gezapft werden kann.

Digitales Video

Mit der Verbreitung von CD-Laufwerken als Speichermedium hat gleichzeitig das digitale Video seinen Siegeszug angetreten. Beim digitalen Video werden die Bilddaten in einer Datei abgelegt, die in komprimierter Form sämtliche Einzelbilder des Films nacheinander enthält. Eine solche Datei kann wie gewohnt auf andere Datenträger kopiert oder auch per DFÜ übertragen werden. Des weiteren können solche Videos mit der geeigneten Software direkt am Computer geschnitten und nachbearbeitet werden. Aufwendige Schnittsysteme, wie sie für analoges Video benötigt werden, entfallen hier, so daß sich jeder Computer-Anwender als Regisseur versuchen kann.

Durch die leichtere Handhabbarkeit und den Wegfall spezieller Zusatz-Hardware sind digitale Videos für den Hobby- und semiprofessionellen Bereich wesentlich besser geeignet als analoge Videos.

Es gibt eine ganze Reihe unterschiedlicher Datenformate für digitales Video. Im folgenden stellen wir Ihnen die wichtigsten Formate vor.

AVI

Dieses Dateiformat ist im PC-Bereich wohl das bekannteste. Das AVI-Format wurde von der Firma Microsoft zusammen mit dem Programmpaket 'Video für Windows' eingeführt. Es handelt sich dabei um eine Software-Lösung zum Abspielen von digitalen Videos.

Der Vorteil dieses Formats liegt in der weiten Verbreitung. Des weiteren lassen sich AVI-Videos mit relativ geringem Aufwand selbst herstellen. Viele neuere Grafikkarten verfügen über einen Video-Eingang und sind in der Lage, das eingehende Videosignal zu digitalisieren und im AVI-Format abzulegen.

Allerdings ist die Auflösung der AVI-Videos eher bescheiden. Auch auf sehr schnellen Rechnern sollte die Auflösung nicht 320*240 Punkte überschreiten, wenn das Video einigermaßen flüssig ablaufen soll. Außerdem benötigen AVI-Videos verglichen mit anderen Formaten (wie z.B. MPEG) relativ viel Speicherplatz.

Seit kurzem gibt es Grafikkarten, die über einen Zusatzchip verfügen, der AVI-Videos auch bildschirmfüllend in guter Qualität abspielen kann.

MPEG

MPEG-Dateien (Dateikennung '.MPG') bieten gegenüber AVI-Dateien eine höhere Datenkompression, höhere Auflösung und Abspielgeschwindigkeit. Die Herstellung solcher Dateien ist aber wesentlich aufwendiger als die Herstellung von AVI-Dateien. Erst seit kurzem (Stand 12/94) stehen Hardware-Lösungen zur Verfügung, um MPEG-Dateien auch mit PC zu erzeugen. Zuvor konnte dieses Format nur auf teuren Parallelrechnern oder sehr teuren PC-Steckkarten (ca. 40.000 DM) erzeugt werden.

MPEG-Dateien können sowohl durch reine Software-Treiber, als auch mit der Unterstützung spezieller MPEG-Decoder-Karten abgespielt werden. Eine derartige Decoder-Karte kann ein MPEG-Video bildschirmfüllend in sehr guter Qualität abspielen.

MPEG wird als das Videoformat der Zukunft angesehen, da es im Bereich des digitalen Videos über die besten Leistungsdaten verfügt. MPEG ist das Dateiformat, das für den geplanten Video-CD-Standard verwendet werden soll.

Für den Hobby- oder semiprofessionellen Bereich ist z.Z. aber noch das AVI-Format besser geeignet.

Quicktime

Das Quicktime-Format kommt aus der Macintosh-Welt und stellt dort das Pendant zu 'Video für Windows' dar. Die Leistung der Quicktime-Videos ist mit denen der AVI-Videos zu vergleichen.

FLI, FLC

Bei diesem Dateiformat handelt es sich eigentlich nicht um echte Videos. Mit diesem Format werden Trickfilme und Animationen abgespeichert, die von dem verbreiteten Programm 'Autodesk-Animator' und vielen anderen Animationsprogrammen erzeugt werden.

Wie erzeuge ich eigene Video-Dateien?

Wenn Sie ein eigenes Video produzieren wollen, müssen Sie zuerst einen Videofilm mit einer Standard-Videokamera aufnehmen. Dabei sollten Sie darauf achten, daß Sie über ein SVHS-

Gerät verfügen, da für die Erzeugung eines digitalen Videos möglichst hochwertiges Ausgangsmaterial verwendet werden sollte.

Mit der fertigen Videokassette können Sie dann auf zweierlei Weise verfahren:

1. Sie digitalisieren das Video selbst. Dazu benötigen Sie einen SVHS-Videorecorder, eine Grafikkarte mit Videoeingang, die das gewünschte Video-Format erzeugen kann, sowie die zur Digitalisierung nötige Software.
2. Sie schicken das Videoband an eine Firma, die Digitalisierungen im Auftrag durchführt. Die Digitalisierung von 1 Minute Video kostet abhängig vom Aufwand i.d.R. zwischen 50 und 150 DM.

Was wird für das Abspielen der Video-Dateien benötigt?

Zuerst benötigen Sie die erforderlichen Software-Treiber, die für das Abspielen des jeweiligen Video-Formats benötigt werden. Unter Windows und OS/2 handelt es sich dabei um sogenannte MCI-Treiber. MCI ist eine einheitliche Schnittstelle für die Ansteuerung von Video und Sound. Für praktisch jede Video- und Soundkarte, sowie für alle wichtigen Software-Video-Formate existieren entsprechende MCI-Treiber.

Falls das Abspielen des Videos noch zusätzlich durch eine Hardware-Lösung unterstützt werden soll, dann muß zusätzlich noch eine entsprechende Steckkarte (z.B. für das Abspielen von MPEG-Videos) eingebaut werden.

Audio

Eine Soundkarte gehört zur Standardausstattung eines multimedia-tauglichen PC. Mit Hilfe einer derartigen Steckkarte kann ein Multimedia-Programm mit Musikuntermalung und Sprachausgabe aufgewertet werden.

Die Audiodaten liegen i.d.R. als Dateien vor, die nach Belieben kopiert oder auch per DFÜ verschickt werden können. Die wichtigsten Dateiformate sind im folgenden aufgelistet.

WAV

Das bekannteste Dateiformat ist hierbei sicherlich das WAV-Format, das unter Windows z.B. zur Erzeugung der Systemtöne dient. WAV-Dateien enthalten digital aufgezeichnete Klanginformationen. Das kann sowohl Musik als auch Sprache sein. Das WAV-Format eignet sich vor allem für die Aufnahme von gesprochenen Kommentaren.

MOD

MOD-Dateien enthalten i.d.R. Musikstücke, die nicht digitalisiert wurden. Stattdessen werden bei diesem Format die Parameter für jeden Ton abgespeichert. Das benötigt erheblich weniger Speicherplatz als das WAV-Format. Dafür ist dieses Format für das Abspeichern von Sprache ungeeignet.

MIDI

In MIDI-Dateien werden wie in MOD-Dateien die Parameter der Töne abgespeichert. Ebenso wie MOD ist dieses Format gut für das platzsparende Abspeichern von Musikstücken geeignet. Für Sprache ist dieses Format aber auch ungeeignet.

Wenn Sie künstlerisch begabt sind, können Sie mit der entsprechenden Software eigene Musikstücke entwickeln. Ansonsten bieten fast alle größeren CD-Händler CD's an, die lizenzfreies Tonmaterial enthalten, das Sie nach Belieben in Ihre Programme einbinden können.

Wie erzeuge ich eigene Audio-Dateien?

Sie benötigen eine Soundkarte mit Mikrofonanschluß, sowie eine Software, die das Digitalisieren der aufgenommenen Tondaten erlaubt. Schließen Sie das Mikrofon an die Soundkarte an, starten Sie das Aufnahmeprogramm und sprechen Sie Ihre Texte in das Mikrofon. Die Aufnahme wird dann in einer WAV-Datei abgespeichert.

Was wird für das Abspielen der Audio-Dateien benötigt?

Neben einer Soundkarte werden entsprechende MCI-Treiber benötigt, welche die Kommunikation zwischen Soundkarte und Programm erledigen. Solche Treiber liegen jeder Soundkarte bei.

CD-ROM

Durch die CD-ROM als Datenträger ist Multimedia erst wirklich interessant geworden. Satt 660 Megabyte passen auf eine der silbernen Scheiben. Das ist mehr als genug Speicher für tausende von Photos, bis zu einer Stunde Video oder mehreren Stunden digitalisiertem Sound.

Wer eine umfangreiche Multimedia-Applikation entwickeln will, wird um die CD als Datenträger kaum herumkommen. Man kann zwar auch Disketten verwenden, die beschränkte Speicherkapazität dieses Mediums stellt aber eine sehr starke Einschränkung dar.

Wie erstellt man nun eigene CD's und was kostet es? Hier herrscht oftmals Verwirrung vor, denn einmal bekommt man gesagt, daß die Herstellung einer CD etwa 100 DM kostet, das andere Mal sollen es nur noch zwei DM pro CD sein. Um diesen Widerspruch aufzulösen, muß man wissen, daß es zwei unterschiedliche Möglichkeiten der Herstellung von CD's gibt.

Brennen von CD's in kleiner Auflage

Benötigt man nur wenige CD's (für die Archivierung von Daten oder ein Firmen-Mailing), kann man die CD's in kleiner Stückzahl mit einem sogenannten CD-Recorder schreiben. Ein solcher CD-Recorder ist ein spezieller CD-Player, der mit einem verstärkten Laserstrahl Informationen auf eine leere CD brennen kann. Das funktioniert natürlich nur einmal. Sind die Daten einmal geschrieben, können Sie nicht mehr gelöscht oder überschrieben werden.

Ein solcher CD-Recorder kostet i.d.R. zwischen 6.000 und 20.000 DM incl. Software. Wenn man die Anschaffungskosten für ein solches Gerät scheut, kann man das Brennen von CD's von Firmen durchführen lassen, die derartige Arbeiten im Kundenauftrag erledigen. Ein CD-Rohling kostet zwischen 15 und 30 DM. Inclusive der Arbeitskosten und der Gebühren für die Benutzung der Geräte kostet eine derartige CD etwa 100 bis 200 DM.

Diese Art der Herstellung eignet sich für die Produktion von CD's bis zu einer Auflage von 10 Stück. Darüber sollte man sich überlegen, ob man die CD nicht lieber pressen läßt.

Pressen von CD's in hoher Auflage

Will man eine große Auflage von CD's herstellen, dann sollte man Sie von einer Firma, die einen derartigen Service anbietet, pressen lassen. Bei dieser Methode wird zuerst eine Art Schablone hergestellt, die alle Informationseinheiten der CD in seitenverkehrter Form enthält (Erhebungen statt Vertiefungen). Mit dieser Schablone können dann beliebig viele CD's gepresst werden. Die gepressten CD's bestehen aus einem anderen Material als die CD's, die mit einem CD-Recorder gebrannt werden und sind wesentlich billiger.

Das Herstellen der Schablone (Mastering) kostet ca. 1000 bis 1500 DM. Für das Pressen einer einzelnen CD muß man je nach Auflage mit Kosten von 2 bis 5 DM rechnen. In diesem Betrag sind die Kosten für die CD-Box und den Labelaufdruck meistens enthalten. Das Herstellen einer Auflage von 1000 Stück kommt incl. Mastering erfahrungsgemäß auf etwa 3000 bis 4000 DM. Dazu kommen noch die Kosten für das Cover und evtl. ein Inlet (Begleitheftchen). Grob gerechnet muß man pro CD mit Kosten zwischen 4 und 6 DM pro Stück rechnen. Je größer die Auflage, desto billiger wird natürlich die Herstellung einer einzelnen CD. Viele Firmen bieten das Mastering bei größeren Auflagen (3000 - 5000 Stück) kostenlos an.

Fazit

Für welche der beiden Möglichkeiten man sich entscheidet, hängt natürlich von der Art der Anwendung ab. Will man nur die eigenen Urlaubsbilder und -videos zu einer Multimedia-Show verarbeiten, wird man sicher eine Einzel-CD brennen lassen. Möchte man dagegen eine Multimedia-Anwendung im großen Stil vermarkten, dann wird man das Pressen einer größeren Auflage in Auftrag geben.

Installation

Eine gute Installationsroutine ist die Visitenkarte eines Programms. Ein Anwender, der sich durch eine komplizierte und vielleicht sogar fehlerhafte Installationsprozedur durcharbeiten muß, wird Ihrem Programm mit Vorbehalten begegnen.

Sie sollten deshalb darauf achten, daß Ihre Applikation mit einer professionell wirkenden Installation versehen ist. Selbst, wenn Ihr Produkt direkt von einer CD laufen soll, sollte durch ein Installationsprogramm wenigstens eine Programmgruppe mit Icons angelegt werden, damit ein Anwender Ihr Programm bequem starten kann.

Ein gutes Installationsprogramm sollte den folgenden Anforderungen genügen:

1. Das Installationsprogramm sollte auf der gleichen Betriebssystemplattform laufen, wie das Produkt, das es installieren soll. Es macht einen schlechten Eindruck, wenn ein Anwender zuerst auf DOS umschalten muß, um ein Windows- oder OS/2-Programm zu installieren.
2. Das Installationsprogramm sollte einfach zu bedienen sein. Am besten sollte von der Diskette oder CD einfach nur ein Programm namens 'install.exe' oder 'setup.exe' gestartet werden, das den Anwender dann menügesteuert durch die Installation führt.
3. Das Installationsprogramm sollte es ermöglichen, Komponenten Ihres Produkts getrennt zu installieren, falls dies gewünscht wird.
4. Die Installationsroutine sollte es erlauben, Programmgruppen anzulegen und Einträge in INI-Dateien vorzunehmen.
5. Es sollte auch ein Deinstallationsprogramm enthalten sein, das eine Applikation wieder von der Festplatte des Anwenders entfernen kann.
6. Einen besonders guten Eindruck macht die Installation, wenn Hintergrundbitmaps eingebunden und während der Installation, weitere Bitmaps mit Erklärungen und Hinweisen ausgegeben werden können.
7. Das Installationsprogramm sollte frei von Lizenzgebühren sein, d.h. wenn Sie das Programm einmal erworben haben, sollte die Weitergabe des Programms an Ihre Kunden erlaubt sein. Es gibt Programme, bei denen für die Weitergabe noch zusätzliche Lizenzgebühren anfallen. So etwas kann teuer werden.

Es gibt mittlerweile eine ganze Reihe von konfigurierbaren Installationsprogrammen auf dem Markt. Im folgenden stellen wir Ihnen für Windows und OS/2 zwei Lösungen vor, die wir selbst verwenden.

Ein Programm, mit dem wir sehr gute Erfahrungen gemacht haben, ist 'EDI Install Pro' der kanadischen Firma 'Eschalon Development Inc'. Es handelt sich hierbei um ein Windows-Installationsprogramm, das allen gestellten Anforderungen gerecht wird und dabei auch noch sehr preiswert ist.

Eschalon Development Inc.
24-2979 Panorama Drive
Coquitlam, BC
V3E 2W8 Canada

Tel: : 604-945-3198

Fax: : 604-945-7602
BBS: : 604-945-7602

CompuServe: 76625,1320
Internet : 76625,1320@compuserve.com

Für OS/2 wird eine Installationsroutine namens 'Software-Installer' direkt von IBM angeboten.
Eine Demo-Version (sidemo.zip) dieses Produkts kann in der IBM-Mailbox (07034/63250)
bezogen werden.

Befehlsreferenz

Das Autorensystem RSE-Author verfügt über ca. 100 Befehle. In der unten stehenden Liste finden Sie alle Befehle nach Objekten und Elementen geordnet aufgelistet.

Allgemeine Befehle

Delay
Delete
ExecuteProgram
InfoBox
LoadBitmap
LoadWindow
Mci
MountProgram
OpenPrinter
PlayAudio
SetFieldPointer
SetGraphicsLibrary
SetHypertextPointer
SetMarkStyle
Sound
Quit
VideoPlayer
Wait

Befehle zum Objekt 'Fenster'

Alias
BackPage
Bookmark
Clear
Close
Disable
DrawBitmap
DrawBorder
DrawCircle
DrawEllipse
DrawIcon
DrawLine
DrawPoint
DrawRectangle
DrawText
Enable
FirstPage
GetBitmap
GrabFocus
Hide
LastPage
LoadPage
Mark
Maximize
Minimize
NextPage
Paint

PlayVideo
PrevPage
Restore
Scroll
SelectPage
SetBackground
SetBorder
SetBrush
SetFont
SetOrigin
SetOutputSize
SetPen
SetPosition
SetRasterOp
SetScript
SetScrollParameter
SetSize
SetTitle
SetUpdateMode
Show
StartTimer
StopTimer
StopVideo
ToTop
Unmark

Befehle zum Element 'Schalter'

Alias
Disable
Enable
GrabFocus
Hide
Paint
SetPosition
SetScript
SetSize
Show

Befehle zum Element 'Feld'

Alias
Disable
Enable
Mark
SetPosition
SetScript
SetSize
Unmark

Befehle zum Element 'Hypertext'

Alias
Disable
Enable

Mark
SetScript
Unmark

Befehle zum Element 'ListBox'

Alias
Disable
Enable
GrabFocus
Hide
Paint
SelectPos
SelectString
SetEntryScript
SetPosition
SetSize
Show

Befehle zum Objekt 'Drucker'

Ein Objekt vom Typ 'Drucker' muß mit dem allgemeinen Befehl OpenPrinter eröffnet werden. Danach können mit dem Befehl LoadDocument beliebig viele Druckseiten aus der Applikation geladen und ausgedruckt werden.

Alias
Close
LoadDocument
PrintPage

Kommentare

Sie können an jeder beliebigen Stelle innerhalb eines Befehls-Scripts Kommentare einfügen. Ein Kommentar wird immer durch ein Ausrufungszeichen (!) eingeleitet. Der Text zwischen dem Ausrufungszeichen und dem Zeilenende wird dann bei der Ausführung des Scripts durch das Laufzeitsystem ignoriert.

Beispiel:

! In diesem Script wird eine neue Hintergrundbitmap eingefügt.

Clear /all ! Fenster löschen.
LoadBitmap back2.bmp /record:window ! Bitmap laden und in den Window-Record eintragen.

Delay

Syntax

Delay msec

Beschreibung

Der Befehl Delay verzögert die Programmausführung um n Millisekunden

Fixe Parameter

msec - Verzögerung in Millisekunden

Optionale Parameter

keine

Beispiel

Delay 5000 ! Programm für 5 Sekunden anhalten

Delete

Syntax

```
Delete object  
Delete object.element  
Delete object /sub
```

Beschreibung

Der Befehl Delete entfernt ein Objekt oder Element vom Bildschirm. Auf diese Weise können z.B. gezielt Schalter oder maussensitive Felder entfernt werden. Wird dieser Befehl auf ein Fensterobjekt angewandt, wird dieses geschlossen. Das Löschen des Objekts 'MainWindow' beendet das Programm.

Fixe Parameter

object
object.element

Als Parameter wird der Name des zu löschenden Objekts/Elements übergeben. Es ist auch möglich, einen verknüpften Namen anzugeben. Soll z.B. der Schalter 'button01' im Fenster 'win1' gelöscht werden, kann dies durch Angabe von 'Delete win1.button01' geschehen. In diesem Fall wird nur der Schalter, nicht aber das Fenster gelöscht.

Optionale Parameter

/sub
/sub:all
/sub:button
/sub:field
/sub:hypertext
/sub:listbox
/sub>window

Dieser Parameter gibt an, daß nicht das durch 'object' angegebene Objekt gelöscht werden soll, sondern die darin enthaltenen Elemente. Das Objekt muß in dem Fall ein Fensterobjekt sein, sonst wird der Parameter '/sub' ignoriert. Wird nur '/sub' angegeben, werden alle Elemente im angegebenen Fenster gelöscht. Ansonsten kann zusätzlich der Typ der zu löschenden Elemente definiert werden. Sollen z.B. alle Schalter im Fenster gelöscht werden, kann dies durch Angabe von '/sub:button' erreicht werden.

Beispiel

```
Delete button01      ! Schalter 'button01' löschen  
Delete win           ! Fenster 'win' löschen  
Delete win.button01 ! Schalter 'button01' im Fenster 'win' löschen  
Delete win /sub      ! Alle Elemente im Fenster 'win' löschen  
Delete win /sub:field ! Alle Felder im Fenster 'win' löschen
```

ExecuteProgram

Syntax

```
ExecuteProgram prog /param:xxxxx
```

Beschreibung

Dieser Befehl startet ein externes Programm, das danach parallel zu der laufenden RSE-Author-Applikation ausgeführt wird.

Fixe Parameter

prog - Name des auszuführenden Programms

Optionale Parameter

```
/p:xxxxx  
/param:xxxxx
```

Diesem optionalen Parameter kann hinter dem Doppelpunkt ein Aufrufparameter für das zu startende Programm mitgegeben werden. Sollen dem Programm mehrere Parameter mitgegeben werden, muß der optionale Parameter mehrmals angegeben werden. In der Zeichenkette hinter dem Doppelpunkt dürfen keine Leerzeichen stehen.

Beispiel

```
ExecuteProgram calc.exe  
ExecuteProgram pbrush.exe /param:earth.bmp  
ExecuteProgram myprog.exe /p:val1 /p:val2 /p:val3
```

InfoBox

Syntax

InfoBox "....."

Beschreibung

Dieser Befehl erzeugt ein kleines Informationsfenster mit einem Text und einem OK-Schalter. Das ist z.B. sinnvoll, um Copyright-Informationen oder kurze Statusmeldungen auszugeben.

Fixe Parameter

"....." - Text, der in der InfoBox ausgegeben werden soll.

Optionale Parameter

keine

Beispiel

InfoBox "Australien-Reisebericht\n\n(C) 1994 Reitz Software-Entwicklung"

LoadBitmap

Syntax

LoadBitmap filename /width:xxx /height:xxx /alias:xxx

Beschreibung

Dieser Befehl lädt eine Grafik aus einer externen Datei in den Arbeitsspeicher. Diese Grafik kann dann mittels des Befehls DrawBitmap in ein beliebiges Fenster kopiert werden.

Es können die folgenden Grafikformate eingelesen werden: BMP, TIF, PCX, GIF und JPG. Die Ausgabegröße der Grafik kann durch die optionalen Parameter '/width' und '/height' verändert werden.

Geschwindigkeit

Die Ladegeschwindigkeiten für die unterschiedlichen Grafikformate sind sehr unterschiedlich. Generell gilt, daß ungepackte Formate (BMP, ungepacktes TIF) schneller geladen werden können als gepackte. Bei Lernprogrammen bei denen es darauf ankommt, daß die Grafiken schnell geladen werden, sollte wenn möglich, das BMP-Format verwendet werden. In der folgenden Liste werden die verschiedenen Grafikformate nach der Geschwindigkeit aufgelistet, mit der sie geladen werden können (die schnellsten zuerst, die langsamsten zuletzt):

BMP
TIF (ungepackt)
TIF (gepackt)
PCX
GIF
JPG

Grafikbibliotheken

Bitmap-Grafiken vom Typ BMP können auch aus einer sogenannten Grafikbibliothek geladen werden. In einer Grafikbibliothek können mehrere dutzend oder auch hunderte von Einzelgrafiken zusammengefaßt werden. Eine solche Bibliothek ist leichter zu handhaben als viele Einzelgrafiken. Außerdem können Grafiken, die sich in einer solchen Bibliothek befinden, nicht von Dritten verändert werden.

Erstellt wird eine Grafikbibliothek mit dem beiliegenden Programm 'grflib.exe'. Um aus einem Lernprogramm auf eine Bibliothek zuzugreifen, muß der Name der Bibliothek mit dem Befehl SetGraphicsLibrary eingestellt werden. Die Syntax des Befehls LoadBitmap verändert sich nicht. Wurde mit SetGraphicsLibrary eine gültige Bibliothek eingestellt, dann versucht das Autorensystem zuerst die Grafik aus der Bibliothek zu laden. Wird sie dort nicht gefunden, dann wird die Grafik als externe Datei auf der Festplatte gesucht.

JPEG-Kompressionsverfahren

Das JPEG-Verfahren (JPG) sollte nur in Verbindung mit Echtfarb-Bildern (16 Millionen Farben) verwendet werden, da nur bei Bildern mit hoher Farbtiefe wirklich gute Kompressionsraten erzielt werden (Faktor 10 - 20). Für Grafiken mit 256 Farben eignet sich JPEG nur bedingt, da für solche Grafiken vor dem Laden der Bilddaten eine Farbpalette erzeugt werden muß, was sehr zeitintensiv ist. Für Grafiken mit 16 oder nur 2

Farben ist das JPEG-Verfahren völlig ungeeignet.

Fixe Parameter

filename

Dateiname der zu ladenden Bitmap. Der Dateiname wird (ohne die Extension) als Objektname verwendet. Wird also z.B. ein Bitmap namens 'earth.bmp' geladen, dann kann die Bitmap unter dem Namen 'earth' angesprochen werden.

Optionale Parameter

/width:xxx

Gibt eine Breite an, auf die das Bitmap beim Laden skaliert werden soll. Wird der Parameter '/height' nicht angegeben, dann errechnet das Autorensystem die Höhe automatisch aus dem ursprünglichen Breiten/Höhen-Verhältnis der Grafik.

/height:xxx

Gibt eine Höhe an, auf die das Bitmap beim Laden skaliert werden soll. Wird der Parameter '/width' nicht angegeben, dann errechnet das Autorensystem die Breite automatisch aus dem ursprünglichen Breiten/Höhen-Verhältnis der Grafik.

/alias:xxx

Vergibt einen Alias-Namen für die Grafik.

Beispiel

```
LoadBitmap earth.bmp /alias:bmp01
SubWin1.DrawBitmap earth 0 0
SubWin2.DrawBitmap bmp01 0 0
```

```
LoadBitmap earth.bmp /width:320 ! Skalieren. Höhe wird angepaßt
LoadBitmap earth.bmp /height:240 ! Skalieren. Breite wird angepaßt
LoadBitmap earth.bmp /width:320 /height:240
```

Querverweis

[DrawBitmap](#)
[SetGraphicsLibrary](#)

LoadWindow

Syntax

LoadWindow object /center /border:xx /parent:xxx

Beschreibung

Dieser Befehl bringt ein Fensterobjekt, das im Editor interaktiv erstellt wurde, auf den Bildschirm.

Fixe Parameter

object - Name des Objekts

Optionale Parameter

/center

Fenster mittig auf dem Bildschirm zentrieren. Dieser Parameter überlagert den Wert, der im Eigenschaftendialog des Fensterobjekts eingegeben wurde.

/border:xx

Fenster mit einem 3D-Rahmen umgeben (gilt nur für Fenster vom Typ 0 und 1). Folgende Rahmentypen sind zulässig:

- 0** - Rechteck
- 1** - abgesenkte Fläche, 1 Pixel tief
- 2** - abgesenkte Fläche, 2 Pixel tief
- 3** - abgesenkte Fläche, 3 Pixel tief
- 4** - angehobene Fläche, 1 Pixel hoch
- 5** - angehobene Fläche, 2 Pixel hoch
- 6** - angehobene Fläche, 3 Pixel hoch
- 7** - abgesenkte Fuge, 1 Pixel tief
- 8** - abgesenkte Fuge, 2 Pixel tief
- 9** - abgesenkte Fuge, 3 Pixel tief
- 10** - angehobene Fuge, 1 Pixel hoch
- 11** - angehobene Fuge, 2 Pixel hoch
- 12** - angehobene Fuge, 3 Pixel hoch
- 13** - abgesenkte Rasterung, 2 Einheiten
- 14** - abgesenkte Rasterung, 3 Einheiten
- 15** - abgesenkte Rasterung, 4 Einheiten
- 16** - angehobene Rasterung, 2 Einheiten
- 17** - angehobene Rasterung, 3 Einheiten
- 18** - angehobene Rasterung, 4 Einheiten
- 19** - Rechteck mit Schatten, 1 Pixel
- 20** - Rechteck mit Schatten, 2 Pixel
- 21** - Rechteck mit Schatten, 3 Pixel
- 22** - Rechteck mit Schatten, 4 Pixel
- 23** - Rechteck mit Schatten, 5 Pixel
- 24** - abgesenkte Fläche mit eingebettetem Rechteck

/parent:xxx

Standardmäßig bekommen Fenster das Applikationsfenster (MainWindow) als Vaterfenster zugeordnet. Durch diesen Parameter kann ein beliebiges Fenster als Vater definiert werden. Dem Doppelpunkt folgt einfach der Objekt- oder Aliasname des gewünschten Fensters.

Beispiel

```
LoadWindow CopyrightInfo  
LoadWindow WorkWindow /center /border:24  
LoadWindow SubWindow2 /parent:WorkWindow
```


Mci

Syntax

Mci "....."

Beschreibung

Mit diesem Befehl können Kommandos direkt an die MCI-Schnittstelle (Media-Control-Interface) des Betriebssystems übergeben werden. Der Aufbau und die Syntax der MCI-Befehle muß der Dokumentation Ihres Betriebssystems entnommen werden.

Beachten Sie, daß RSE-Author wesentlich komfortablere Möglichkeiten zum Abspielen von Video und Audio anbietet. Dieser Befehl wurde für den Fall implementiert, daß die Standardfunktionen des Autorensystems für spezielle Anwendungsbereiche nicht ausreichen.

Fixe Parameter

"....."

Dem Befehl werden innerhalb von Anführungszeichen die MCI-Kommandos, die ausgeführt werden sollen, übergeben. Innerhalb des Befehlsstrings kann das Systemhandle eines Fensters durch die Funktion `hwnd(.....)` eingefügt werden. Das Autorensystem ersetzt diese Zeichenkette vor Ausführung des MCI-Kommandos durch das Systemhandle des übergebenen Fensterobjekts.

Optionale Parameter

keine

Beispiel

```
Mci "open test.avi alias xxx"  
Mci "window xxx handle hwnd(VideoWindow)"  
Mci "play xxx"
```

Querverweis

[PlayAudio](#)

[PlayVideo](#)

[VideoPlayer](#)

MountProgram

Syntax

MountProgram filename

Beschreibung

Ein RSE-Author-Programm kann maximal 500 Objekte enthalten. Bei sehr umfangreichen Projekten, wie elektronischen Büchern, Lexika usw. kann das zu wenig sein. Deshalb erlaubt dieser Befehl, der aktuell geladenen Programmdatei weitere Programmdateien hinzuzufügen. Auf diese Weise können Applikationen mit mehreren tausend Objekten erstellt werden. Die maximale Größe einer Applikation wird also nur durch den Arbeitsspeicher des Rechners begrenzt.

Bei der Bearbeitung der verschiedenen Programme sollte darauf geachtet werden, daß Objektnamen innerhalb der verschiedenen Programmdateien nicht doppelt vorkommen, da diese nach dem Einlagern wie ein Programm behandelt werden. Eine Ausnahme macht hier 'MainWindow', das automatisch angelegt wird. Beim Laden einer zusätzlichen Programmdatei mit 'MountProgram' wird dieses Objekt überlesen.

Dieser Befehl kann auch dazu verwendet werden, die Arbeit an einer Applikation aufzuteilen. Verschiedene Entwickler können gleichzeitig an verschiedenen Kapiteln eines Programms arbeiten. Innerhalb eines Hauptprogramms werden diese Kapitel dann einfach eingelagert.

Fixe Parameter

filename - Dateiname des Programms, das hinzugefügt werden soll.

Optionale Parameter

keine

Beispiel

```
MountProgram kap1.prg    ! könnte z.B. im Open-Handler von 'MainWindow' stehen.  
MountProgram kap2.prg  
MountProgram kap3.prg
```

OpenPrinter

Syntax

```
OpenPrinter name /option /break
```

Beschreibung

Dieser Befehl eröffnet einen angeschlossenen Drucker für die Ausgabe von Druckseiten, die in der Entwicklungsumgebung als Objekte vom Typ 'Dokument' erstellt wurden. Der Drucker wird als eigenständiges Objekt behandelt, das eigene Befehle ausführen kann.

Fixe Parameter

name - Name des Druckerobjekts

Optionale Parameter

/option

Vor dem Drucken wird eine Dialogbox aufgeblendet, in der der Anwender den Drucker konfigurieren kann. Des weiteren wird hier dem Anwender die Möglichkeit geboten, den Druckvorgang abzuberechnen.

/break

Während des Druckens wird eine Dialogbox aufgeblendet, die dem Anwender die Möglichkeit gibt, den Druckvorgang abzuberechnen.

Beispiel

```
OpenPrinter prn /option  
prn.LoadDocument doc001  
prn.PrintPage  
prn.LoadDocument doc002  
prn.PrintPage  
prn.LoadDocument doc003  
prn.PrintPage  
prn.Close
```

PlayAudio

Syntax

```
PlayAudio filename /loop /from:xxx /to:xxx
```

Beschreibung

Dieser Befehl spielt eine Audio-Datei über eine angeschlossene Soundkarte ab.

Fixe Parameter

filename - Dateiname des Audiofiles, das abgespielt werden soll.

Optionale Parameter

/loop

Das Abspielen der Audio-Datei wird bei Erreichen des Dateiendes neu angestoßen. Die Audio-Datei wird solange abgespielt, bis das Abspielen durch den Befehl [StopAudio](#) explizit angehalten oder das Programm beendet wird.

/from:xxx

Startposition in Millisekunden, ab der das Audiofile abgespielt werden soll.

/to:xxx

Endposition in Millisekunden, bis zu der das Audiofile abgespielt werden soll.

Beispiel

```
PlayAudio intro.wav  
PlayAudio intro.wav /loop  
PlayAudio intro.wav /from:10000 /to:20000
```

Querverweis

[StopAudio](#)
[PlayVideo](#)
[VideoPlayer](#)
[Mci](#)

SetFieldPointer

Syntax

SetFieldPointer par

Beschreibung

Dieser Befehl stellt den Standard-Mauszeiger für Hyperlinkfelder ein. Dieser Mauszeiger erscheint, wenn die Maus in ein Hyperlinkfeld hineinbewegt wird. Bei der Bearbeitung eines Hyperlinkfeldes in der Entwicklungsumgebung kann dem Feld ein individueller Mauszeiger zugeordnet werden. Geschieht dies nicht, wird der durch diesen Befehl eingestellte Zeiger verwendet.

Fixe Parameter

par

Nummer des Mauszeigers, der als Standardzeiger dienen soll. Folgende Zeiger stehen zur Verfügung:

- 0** - kein Mauszeiger
- 1** - Pfeil (Standardeinstellung)
- 2** - Sanduhr
- 3** - Fadenkreuz
- 4** - Linie mit aufgespaltenen Enden
- 5** - zeigende Hand
- 6** - hinweisende Hand
- 7** - Siegeszeichen
- 8** - OK-Zeichen
- 9** - Daumen hoch
- 10** - Daumen runter
- 11** - schreibende Hand
- 12** - grinsendes Gesicht
- 13** - neutrales Gesicht
- 14** - trauriges Gesicht
- 15** - Maus
- 16** - Diskette
- 17** - CD
- 18** - Tastatur
- 19** - Computer
- 20** - Telefon
- 21** - Buch
- 22** - Briefkasten
- 23** - Brief1
- 24** - Brief2
- 25** - Fahne

Optionale Parameter

keine

Beispiel

SetFieldPointer 5 ! zeigende Hand als Standardzeiger

Querverweis

SetHypertextPointer

SetGraphicsLibrary

Syntax

```
SetGraphicsLibrary filename
```

Beschreibung

Dieser Befehl stellt den Namen einer Grafikbibliothek ein, aus der Grafiken mit dem Befehl LoadBitmap geladen werden können. Eine solche Grafikbibliothek stellt eine Riesendatei dar, in der beliebig viele Grafiken vom Typ BMP zusammengefaßt werden können. Eine solche Bibliothek ist leichter zu handhaben als viele Einzelgrafiken. Außerdem können Grafiken, die sich in einer solchen Bibliothek befinden, nicht von Dritten verändert werden.

Die Syntax des Befehls LoadBitmap verändert sich nicht. Wurde mit SetGraphicsLibrary eine gültige Bibliothek eingestellt, dann versucht das Autorensystem zuerst die Grafik aus der Bibliothek zu laden. Wird sie dort nicht gefunden, dann wird die Grafik als externe Datei auf der Festplatte gesucht.

Erstellt wird eine Grafikbibliothek mit dem beiliegenden Programm 'grflib.exe'.

Fixe Parameter

filename - Dateiname der Grafikbibliothek

Optionale Parameter

keine

Beispiel

```
SetGraphicsLibrary graphics.lib
```

Querverweis

LoadBitmap

SetHypertextPointer

Syntax

SetHypertextPointer par

Beschreibung

Dieser Befehl stellt den Standard-Mauszeiger für Hypertextfelder ein. Dieser Mauszeiger erscheint, wenn die Maus in ein Hypertextfeld hineinbewegt wird. Bei der Bearbeitung eines Hypertextfeldes in der Entwicklungsumgebung kann dem Feld ein individueller Mauszeiger zugeordnet werden. Geschieht dies nicht, wird der durch diesen Befehl eingestellte Zeiger verwendet.

Fixe Parameter

par

Nummer des Mauszeigers, der als Standardzeiger dienen soll. Folgende Zeiger stehen zur Verfügung:

- 0** - kein Mauszeiger
- 1** - Pfeil
- 2** - Sanduhr
- 3** - Fadenkreuz
- 4** - Linie mit aufgespaltenen Enden
- 5** - zeigende Hand (StandardEinstellung)
- 6** - hinweisende Hand
- 7** - Siegeszeichen
- 8** - OK-Zeichen
- 9** - Daumen hoch
- 10** - Daumen runter
- 11** - schreibende Hand
- 12** - grinsendes Gesicht
- 13** - neutrales Gesicht
- 14** - trauriges Gesicht
- 15** - Maus
- 16** - Diskette
- 17** - CD
- 18** - Tastatur
- 19** - Computer
- 20** - Telefon
- 21** - Buch
- 22** - Briefkasten
- 23** - Brief1
- 24** - Brief2
- 25** - Fahne

Optionale Parameter

keine

Beispiel

SetHypertextPointer 5 ! zeigende Hand als Standardzeiger

Querverweis

SetFieldPointer

SetMarkStyle

Syntax

SetMarkStyle style width color

Beschreibung

Felder und Hypertextfelder können mittels des Befehls Mark mit einem Rahmen umgeben werden, der mit dem Befehl Unmark wieder entfernt werden kann. Das ist z.B. sinnvoll, wenn ein Feld beim Hineinbewegen des Mauszeigers optisch hervorgehoben werden soll.

Das Aussehen des Rahmens, den der Befehl Mark zeichnet, kann dabei durch einen optionalen Parameter ('/style:xx:xx:xx) individuell eingestellt werden. Wird dieser optionale Parameter nicht angegeben, werden die Standardwerte verwendet, die durch den Befehl 'SetMarkStyle' definiert werden können.

Fixe Parameter

style

Linienart. Folgende fünf Werte sind zulässig:

- 0 - null
- 1 - solid
- 2 - dot
- 3 - dash
- 4 - dashdot

Es kann sowohl die Nummer, als auch die dahinterstehende Textkonstante verwendet werden.

width

Breite der Linie in Pixeln.

color

Farbe der Umrandung. Folgende 16 Werte sind zulässig:

- 0 - black
- 1 - blue
- 2 - green
- 3 - cyan
- 4 - red
- 5 - magenta
- 6 - brown
- 7 - gray
- 8 - lightgray
- 9 - lightblue
- 10 - lightgreen
- 11 - lightcyan
- 12 - lightred
- 13 - lightmagenta
- 14 - yellow

15 - white

Es kann sowohl die Nummer, als auch die dahinterstehende Textkonstante verwendet werden.

Optionale Parameter

keine

Beispiel

```
SetMarkStyle 2 0 2  
SetMarkStyle dot 0 green
```

Querverweis

Mark
Unmark

Sound

Syntax

Sound par

Beschreibung

Dieser Befehl gibt einen Ton aus. Damit können z.B. Aussagen unterstrichen werden oder auf fehlerhafte Eingaben reagiert werden. Voraussetzung ist, daß ein passender Audio-Treiber (für Soundkarte oder Speaker) installiert ist.

Fixe Parameter

par

Nummer des Tonsignals, das ausgegeben werden soll. Folgende Werte sind zulässig:

- 0 - default
- 1 - info
- 2 - warning
- 3 - error
- 4 - query

Es kann sowohl die Nummer, als auch die dahinterstehende Textkonstante verwendet werden.

Optionale Parameter

keine

Beispiel

Sound 1
Sound info

Quit

Syntax

Quit

Beschreibung

Dieser Befehl beendet die Ausführung des laufenden Programmes.

VideoPlayer

Syntax

```
VideoPlayer filename width height /showpos /start
```

Beschreibung

Dieser Befehl eröffnet einen sogenannten Video-Player auf dem Bildschirm. Dieser Player besteht aus einem Fenster, in dem das Video angezeigt wird und einer Schalterleiste, die Funktionen wie Start, Stop, Pause, schnellen Vor- und Rücklauf usw. enthält. Der Anwender hat die Möglichkeit, sich das Video beliebig oft anzusehen und nach Belieben vor- und zurückzuspulen.

Die Parameter 'width' und 'height' geben die Breite und Höhe für das Ausgabefenster der Video-Datei an. Der Video-Player wird dann in seiner Größe automatisch an die Größe des Ausgabefensters angepaßt. Wenn Sie also z.B. ein AVI-Video namens test.avi abspielen wollen, das eine Auflösung von 320 mal 240 Punkten hat, dann geben Sie folgenden Aufruf an: 'VideoPlayer test.avi 320 240'.

Sie können auch größere oder kleinere Auflösungen angeben. Das Video wird dann beim Abspielen automatisch skaliert. Eine Skalierung des Videos führt allerdings i.d.R. zu Qualitätsverlusten in der Darstellung. Ein Video sollte deshalb, wenn möglich, immer mit der Auflösung abgespielt werden, mit der es aufgenommen wurde.

Diese Funktion ist für das Abspielen von Autodesk-Animator-Dateien (FLI, FLC) nur sehr bedingt geeignet, da die Struktur dieses Dateiformats das Vor- und Rückspulen der Datei nicht erlaubt. Außerdem kann die aktuelle Position innerhalb der Datei nicht angezeigt werden, da der MCI-Treiber keine entsprechenden Informationen liefert.

Fixe Parameter

filename

Dateiname der Video-Datei, die angezeigt werden soll.

width

Breite des Video-Ausgabefensters in Pixeln.

height

Höhe des Video-Ausgabefensters in Pixeln.

Optionale Parameter

/showpos

Wird dieser Parameter angegeben, dann zeigt der VideoPlayer beim Abspielen die verstrichene Zeit, sowie die Gesamtspieldauer des Videos an.

/start

Wird dieser Parameter angegeben, dann wird das Abspielen des Videos nach dem Eröffnen des VideoPlayer automatisch gestartet.

Beispiel

VideoPlayer test.avi 320 240

Querverweis

PlayAudio

PlayVideo

Mci

Wait

Syntax

Wait /time:xxxx

Beschreibung

Dieser Befehl hält die Ausführung des laufenden Programms an und wartet auf eine Eingabe (Tastendruck oder Mausklick). Das ist z.B. sinnvoll, wenn eine Grafik bildschirmfüllend angezeigt werden soll und durch einen Tastendruck wieder ins Programm zurückgeschaltet werden soll.

Fixe Parameter

keine

Optionale Parameter

/time:xxxx

Über diesen Parameter kann eine Maximalzeit in Millisekunden definiert werden, die das Autorensystem auf einen Tastendruck warten soll. Nach Ablauf dieser Zeit setzt das Laufzeitsystem die Programmausführung automatisch fort.

Beispiel

Wait ! Auf Tastendruck warten
Wait /time:30000! Maximal 30 Sekunden auf Tastendruck warten

Alias

Syntax

Alias name

Beschreibung

Beim Anlegen eines Objekts in der Entwicklungsumgebung muß ein Name für das Objekt vergeben werden. In der Regel wird dies ein aussagekräftiger Name sein wie 'PageWindow' oder 'CopyrightInfo', um auch später noch anhand des Namens die Funktion des Objekts erkennen zu können. Bei der Erfassung eines Befehls-Scripts ist es aber sehr umständlich, jedesmal einen derart langen Objektnamen auszusprechen. Deshalb besteht die Möglichkeit jedem Objekt zusätzlich noch einen kurzen Namen zu geben, um sich Tipparbeit zu ersparen. Dieser Kurzname kann mit dem Befehl 'Alias' definiert werden. Das Objekt kann dann sowohl mit seinem vollen Objektnamen als auch unter diesem Kurznamen angesprochen werden.

Fixe Parameter

name - Kurzname, unter dem das Objekt angesprochen werden kann.

Optionale Parameter

keine

Beispiel

```
MainWindow.Alias mw  
mw.DrawText "Die Vergabe eines Alias spart Arbeit."
```

BackPage

Syntax

BackPage

Beschreibung

Werden in einem Fenster Seitenobjekte durchgeblättert, speichert das Laufzeitsystem die zuletzt aufgerufenen 20 Seiten. Mit dem Befehl BackPage kann die zuvor geladene Seite wieder ins Fenster geladen werden. Das ist sinnvoll, wenn, unabhängig davon, wie eine Seite aufgerufen wurde, eine Rücksprungfunktion in eine Applikation eingebaut werden soll.

Beachten Sie bitte, daß der Befehl Clear die Aufzeichnung der Seiten löscht. Danach ist kein Rücksprung mehr möglich.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

PageWindow.BackPage ! Letzte Seite aufrufen.

Querverweis

LoadPage

FirstPage

LastPage

NextPage

PrevPage

SelectPage

Bookmark

Syntax

Bookmark /set /call /delete

Beschreibung

Dieser Befehl erlaubt es, Lesezeichen zu setzen und aufzurufen. Ein Lesezeichen ist ein Verweis auf ein Seitenobjekt, das jederzeit zugänglich sein soll. Ein Anwender kann für Bildschirmseiten, die von besonderem Interesse sind, Lesezeichen setzen und später durch Auswahl dieses Lesezeichens, die entsprechende Bildschirmseite wieder aufrufen.

Je nachdem, welche optionalen Parameter Sie dem Befehl mitgeben, wird eine Dialogbox zum Setzen ('/set') oder zum Aufrufen ('/call') eines bereits gesetzten Lesezeichens aufgeblendet. Wird kein optionaler Parameter angegeben, dann wird die Dialogbox zum Aufruf eines vorhandenen Lesezeichens aufgerufen.

Beachten Sie, daß jedes Fensterobjekt eine eigene Liste von Lesezeichen besitzt. Es können also in verschiedenen Fenstern Informationen verwaltet werden, die über getrennte Lesezeichen verfügen.

Wenn Sie den Befehl 'Bookmark' in Ihrem Programm verwenden, dann sollten Sie die Funktionsweise der Lesezeichen in der Dokumentation Ihres Programms erläutern.

Fixe Parameter

keine

Optionale Parameter

/set

Dieser Parameter blendet die Dialogbox zum Setzen eines Lesezeichens auf. In dieser Dialogbox können Lesezeichen eingefügt, geändert und gelöscht werden.

Um ein neues Lesezeichen anzulegen, muß der Einfügen-Schalter angeklickt werden. Daraufhin wird in einer kleinen Editbox ein erklärender Text zu dem Lesezeichen eingegeben. Nach Bestätigung dieses Textes wird der Text an der aktuellen Position in die Liste der Lesezeichen eingefügt. Diesem Text ist in der Folge das Seitenobjekt zugeordnet, das zum Zeitpunkt des Aufrufs der Dialogbox im Fenster geladen war. Wird später die Dialogbox zum Aufruf von Lesezeichen aufgerufen, wird durch Auswahl dieses Textes die zugeordnete Seite geladen.

Des weiteren kann der Text eines bestehenden Lesezeichens durch Betätigung des Ändern-Schalters editiert werden. Die Zuordnung der hinterlegten Seite ändert sich dadurch nicht.

Außerdem können nicht mehr benötigte Lesezeichen gelöscht werden, indem das entsprechende Lesezeichen selektiert und anschließend der Löschen-Schalter angeklickt wird.

/call

Dieser Parameter blendet die Dialogbox zum Aufruf eines Lesezeichens auf. In einer Liste

werden alle Lesezeichen aufgelistet. Durch Doppelklick auf einen Eintrag oder Bestätigung mit der OK-Taste wird die zum Lesezeichen gehörige Seite geladen.

/delete

Durch Angabe dieses Parameters werden alle Lesezeichen, die im angegebenen Fenster gesetzt wurden, gelöscht.

Beispiel

```
win.Bookmark /set    ! Lesezeichen setzen  
win.Bookmark /call   ! Lesezeichen aufrufen  
win.Bookmark /delete ! Lesezeichen löschen
```

Clear

Syntax

Clear /window /page /all /record

Beschreibung

Dieser Befehl löscht den Inhalt eines Fenster. Wenn Sie ausschließlich mit Seiten arbeiten, benötigen Sie diesen Befehl nicht, da beim Laden einer Seite das Fenster automatisch gelöscht wird.

Wurde zuvor eine Seite in das Fenster geladen, kann nach Ausführung des Befehls 'Clear' das PageExit-Script der Seite nicht mehr ausgeführt werden.

Fixe Parameter

keine

Optionale Parameter

/window

Der Fensterhintergrund wird gelöscht. Dabei handelt es sich um alle grafischen Elemente, die bei der Bearbeitung des Fensterobjekts direkt in das Fenster gesetzt wurden.

/page

Die aktuell geladene Seite wird gelöscht. Es werden alle grafischen Elemente gelöscht, die durch das Laden einer Seite oder durch einen Draw.....-Befehl in das Fenster eingetragen wurden.

/all

Hintergrund, aktuelle Seite, sowie alle Elemente (Schalter, Feld, Hypertext usw.) im Fenster werden gelöscht.

/record

Nur die Aufzeichnungen der Grafikausgaben, die für die Wiederherstellung des Fensterinhalts benötigt werden, werden gelöscht. Der Fensterinhalt bleibt erhalten, kann aber bei einer Repaint-Anforderung nicht wiederhergestellt werden.

Beispiel

```
win.Clear      ! Hintergrund und Seite werden gelöscht.  
win.Clear /all ! Hintergrund, Seite und Elemente werden gelöscht.  
win.Clear /page ! Seite wird gelöscht.  
win.Clear /page /record ! Aufzeichnung der Seite wird gelöscht. Der Fensterinhalt  
bleibt erhalten
```

Close

Syntax

Close

Beschreibung

Dieser Befehl schließt ein Fenster. Wird das Objekt 'MainWindow' mit diesem Befehl geschlossen, wird das Programm automatisch beendet. Beim Schliessen des Fensters über diesen Befehl wird das Close-Script des Fensters nicht aufgerufen.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

CopyrightInfo.Close ! Copyright-Informations-Fenster schliessen

Disable

Syntax

Disable

Beschreibung

Nach Aufruf dieses Befehl ist das Fenster nicht mehr aktivierbar. Es erhält keine Eingaben mehr, d.h. es kann nicht mehr mit der Maus angeklickt werden. Das gleiche gilt für alle Elemente (Schalter, Hyperlink, Hypertext usw.).

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

MainWindow.Disable ! Applikationsfenster ist nicht aktivierbar

Querverweis

[Enable](#)

DrawBitmap

Syntax

```
DrawBitmap name x y /fade:xx:xx:xx /transparent:xx /record:xxx /pal /nopal
```

Beschreibung

Dieser Befehl zeichnet ein Bitmap-Objekt in ein Fenster. Die Bitmap muß zuvor mit dem Befehl LoadBitmap als eigenständiges Objekt geladen worden sein.

Fixe Parameter

name

Name des Bitmap-Objekts

x

X-Koordinate, an der die Bitmap ausgegeben werden soll.

y

Y-Koordinate, an der die Bitmap ausgegeben werden soll.

Optionale Parameter

/fade:effekt

/fade:effekt:groesse:pause

Durch Angabe dieses Parameters kann die Bitmap mittels eines Überblendeffekts in das Fenster kopiert werden. Dem optionalen Parameter wird dazu die Nummer des gewünschten Überblendeffekts übergeben. Folgende Effekte sind möglich:

- 0** - absolutes Kopieren
- 1** - überblenden
- 2** - von oben nach unten
- 3** - von unten nach oben
- 4** - von links nach rechts
- 5** - von rechts nach links
- 6** - ausgehend von der Mitte nach oben und unten
- 7** - ausgehend von der Mitte nach links und rechts
- 8** - von oben und unten zur Mitte hin
- 9** - von links und rechts zur Mitte hin
- 10** - von innen nach aussen zoomen
- 11** - von aussen nach innen zoomen
- 12** - von oben nach unten einrollen
- 13** - von unten nach oben einrollen
- 14** - von links nach rechts einrollen
- 15** - von rechts nach links einrollen

Zusätzlich zu der Art des Überblendeffekts können noch zwei weitere Parameter angegeben werden: 'groesse' und 'pause'. Je nach Effekt kann die Bedeutung dieser Parameter variieren.

Für den Effekt 1 (überblenden) gibt der Parameter 'groesse' die Kantenlänge in Pixeln für einen kopierten Block an. Für alle anderen Effekte gibt dieser Parameter an, wieviele Linien zwischen zwei Pausen kopiert werden sollen.

Der Parameter 'pause' gibt die Verzögerung zwischen zwei Kopiervorgängen an. Je größer dieser Wert ist, desto langsamer wird eine Grafik aufgeblendet.

/trans:color

/transparent:color

Mit diesem Parameter wird eine transparente Farbe für die Verknüpfung der Bitmap mit dem Hintergrund eingestellt. Als Transparenzfarben sind die Farben Schwarz und Weiß zulässig. Der Wert der Farbe wird dem Parameter hinter dem Doppelpunkt angegeben. Es können sowohl die numerischen Werte der Farbe 0 oder 15, als auch Textkonstanten (black, white) angegeben werden.

```
/trans:0  
/trans:15  
/trans:black  
/trans:white
```

```
/transparent:0  
/transparent:15  
/transparent:black  
/transparent:white
```

/record:xxxxx

Innherhalb eines Fensters werden alle Ausgaben von Text und Grafik in zwei sogenannten Metafiles aufgezeichnet: im Window-Metafile und im Page-Metafile.

Mit diesem Parameter kann festgelegt werden, in welchem Metafile die Ausgabe der Bitmap aufgezeichnet werden soll. Dem Namensteil des Parameter folgt nach dem Doppelpunkt eine Angabe, in welchem Metafile die Bitmap aufgezeichnet werden soll.

```
/record:none - Die Ausgabe der Bitmap wird nicht aufgezeichnet.  
/record>window - Die Bitmap wird im Window-Metafile, also im Hintergrund  
aufgezeichnet.  
/record:page - Die Bitmap wird im Page-Metafile, also im Vordergrund  
aufgezeichnet (Standardeinstellung).
```

/pal

Gibt an, daß die Palette der Bitmap im Fenster eingestellt werden soll. Bei palettenbehafteten Bitmaps mit 256 Farben ist das notwendig. Das ist die Standardeinstellung.

/nopal

Gibt an, daß die Palette der Bitmap nicht eingestellt werden soll. Das kann sinnvoll sein, wenn eine andere Bitmap mit der gleichen Palette bereits in das Fenster geladen wurde, oder wenn das Programm grundsätzlich in einem HighColor- oder TrueColor-Modus arbeitet.

Beispiel

```
LoadBitmap earth.bmp /alias:bmp01
```

! Bitmap wird ohne alles einfach nur gezeichnet
win.DrawBitmap bmp01 50 50
Wait

! Bitmap wird überblendet
win.Clear
win.DrawBitmap bmp01 50 50 /fade:01
Wait

! Bitmap wird überblendet, Die Blockgröße ist 2
win.Clear
win.DrawBitmap bmp01 50 50 /fade:01:02:00
Wait

! Beim Zeichnen wird die Farbe schwarz transparent
win.Clear
win.DrawBitmap bmp01 50 50 /transparent:black
Wait

! Bitmap wird beim Zeichnen in das Window-Metafile eingetragen
win.Clear /all
win.DrawBitmap bmp01 50 50 /record:window
Wait

! Die Ausgabe der Bitmap wird gar nicht aufgezeichnet
win.Clear /all
win.DrawBitmap bmp01 50 50 /record:none
Wait

Querverweis

[LoadBitmap](#)

DrawBorder

Syntax

DrawBorder x y width height type /record:xxxx

Beschreibung

Dieser Befehl zeichnet einen 3D-Rahmen. Die meisten Rahmentypen setzen voraus, daß die Hintergrundfarbe des Fensters hellgrau ist, da nur in dieser Farbkombination ein dreidimensionales Aussehen der Rahmen erreicht wird.

Fixe Parameter

x

X-Koordinate des Rahmens.

y

Y-Koordinate des Rahmens.

width

Breite des Rahmens.

height

Höhe des Rahmens.

type

Typ des Rahmens. Folgende Werte sind zulässig:

- 0** - Rechteck
- 1** - abgesenkte Fläche, 1 Pixel tief
- 2** - abgesenkte Fläche, 2 Pixel tief
- 3** - abgesenkte Fläche, 3 Pixel tief
- 4** - angehobene Fläche, 1 Pixel hoch
- 5** - angehobene Fläche, 2 Pixel hoch
- 6** - angehobene Fläche, 3 Pixel hoch
- 7** - abgesenkte Fuge, 1 Pixel tief
- 8** - abgesenkte Fuge, 2 Pixel tief
- 9** - abgesenkte Fuge, 3 Pixel tief
- 10** - angehobene Fuge, 1 Pixel hoch
- 11** - angehobene Fuge, 2 Pixel hoch
- 12** - angehobene Fuge, 3 Pixel hoch
- 13** - abgesenkte Rasterung, 2 Einheiten
- 14** - abgesenkte Rasterung, 3 Einheiten
- 15** - abgesenkte Rasterung, 4 Einheiten
- 16** - angehobene Rasterung, 2 Einheiten
- 17** - angehobene Rasterung, 3 Einheiten
- 18** - angehobene Rasterung, 4 Einheiten
- 19** - Rechteck mit Schatten, 1 Pixel
- 20** - Rechteck mit Schatten, 2 Pixel
- 21** - Rechteck mit Schatten, 3 Pixel
- 22** - Rechteck mit Schatten, 4 Pixel
- 23** - Rechteck mit Schatten, 5 Pixel
- 24** - abgesenkte Fläche mit eingebettetem Rechteck

Optionale Parameter

/record:xxxx

Innherhalb eines Fensters werden alle Ausgaben von Text und Grafik in zwei sogenannten Metafiles aufgezeichnet: im Window-Metafile und im Page-Metafile.

Mit diesem Parameter kann festgelegt werden, in welchem Metafile die Ausgabe des Rahmens aufgezeichnet werden soll. Dem Namensteil des Parameter folgt nach dem Doppelpunkt eine Angabe, in welchem Metafile die Bitmap aufgezeichnet werden soll.

`/record:none` - Die Ausgabe des Rahmens wird nicht aufgezeichnet.

`/record>window` - Der Rahmen wird im Window-Metafile, also im Hintergrund aufgezeichnet.

`/record:page` - Der Rahmen wird im Page-Metafile, also im Vordergrund aufgezeichnet (Standardeinstellung).

Beispiel

```
win.DrawBorder 50 50 400 300 24  
Wait
```

```
win.Clear /all  
win.DrawBorder 50 50 400 300 22 /record>window
```

DrawCircle

Syntax

```
DrawCircle x y radius /record:xxxx
```

Beschreibung

Dieser Befehl zeichnet einen Kreis. Linienart und Füllmuster können zuvor mit den Befehlen SetPen und SetBrush definiert werden.

Fixe Parameter

x
X-Koordinate des Kreismittelpunkts.

y
Y-Koordinate des Kreismittelpunkts.

radius
Radius des Kreises in Pixeln

Optionale Parameter

/record:xxxx

Innherhalb eines Fensters werden alle Ausgaben von Text und Grafik in zwei sogenannten Metafiles aufgezeichnet: im Window-Metafile und im Page-Metafile.

Mit diesem Parameter kann festgelegt werden, in welchem Metafile die Ausgabe des Kreises aufgezeichnet werden soll. Dem Namensteil des Parameter folgt nach dem Doppelpunkt eine Angabe, in welchem Metafile die Bitmap aufgezeichnet werden soll.

/record:none - Die Ausgabe des Kreises wird nicht aufgezeichnet.

/record>window - Der Kreis wird im Window-Metafile, also im Hintergrund aufgezeichnet.

/record:page - Der Kreis wird im Page-Metafile, also im Vordergrund aufgezeichnet (Standardeinstellung).

Beispiel

```
win.SetPen solid 0 lightblue
win.SetBrush diagcross blue yellow
win.DrawCircle 200 200 150
Wait

win.Clear /all

win.SetPen dashdot 2 green
win.SetBrush b50 green lightcyan
win.DrawCircle 200 200 150 /record>window
```

Querverweis

SetPen

SetBrush

DrawEllipse

Syntax

```
DrawEllipse x y xrad yrad /record:xxxx
```

Beschreibung

Dieser Befehl zeichnet eine Ellipse. Linienart und Füllmuster können zuvor mit den Befehlen SetPen und SetBrush definiert werden.

Fixe Parameter

x
X-Koordinate des Ellipsenmittelpunkts.

y
Y-Koordinate des Ellipsenmittelpunkts.

xrad
Radius der Ellipse in horizontaler Richtung.

yrad
Radius der Ellipse in vertikaler Richtung.

Optionale Parameter

/record:xxxx

Innherhalb eines Fensters werden alle Ausgaben von Text und Grafik in zwei sogenannten Metafiles aufgezeichnet: im Window-Metafile und im Page-Metafile.

Mit diesem Parameter kann festgelegt werden, in welchem Metafile die Ausgabe der Ellipse aufgezeichnet werden soll. Dem Namensteil des Parameter folgt nach dem Doppelpunkt eine Angabe, in welchem Metafile die Bitmap aufgezeichnet werden soll.

`/record:none` - Die Ausgabe der Ellipse wird nicht aufgezeichnet.

`/record>window` - Die Ellipse wird im Window-Metafile, also im Hintergrund aufgezeichnet.

`/record:page` - Die Ellipse wird im Page-Metafile, also im Vordergrund aufgezeichnet (Standardeinstellung).

Beispiel

```
win.SetPen solid 0 lightblue  
win.SetBrush diagcross blue yellow  
win.DrawEllipse 200 200 100 80  
Wait
```

```
win.Clear /all
```

```
win.SetPen dashdot 2 green  
win.SetBrush b50 green lightcyan  
win.DrawEllipse 200 200 100 80 /record>window
```

Querverweis

SetPen

SetBrush

DrawIcon

Syntax

```
DrawIcon x y type /record:xxxx
```

Beschreibung

Dieser Befehl zeichnet ein Standard-Icon.

Fixe Parameter

x

X-Koordinate, an der das Icon gezeichnet werden soll.

y

Y-Koordinate, an der das Icon gezeichnet werden soll.

type

Typ des Icons. Folgende Werte sind zulässig:

- 0** - default
- 1** - info
- 2** - warning
- 3** - error
- 4** - query

Es kann sowohl die Nummer, als auch die dahinterstehende Textkonstante verwendet werden.

Optionale Parameter

/record:xxxx

Innherhalb eines Fensters werden alle Ausgaben von Text und Grafik in zwei sogenannten Metafiles aufgezeichnet: im Window-Metafile und im Page-Metafile.

Mit diesem Parameter kann festgelegt werden, in welchem Metafile die Ausgabe des Icons aufgezeichnet werden soll. Dem Namensteil des Parameter folgt nach dem Doppelpunkt eine Angabe, in welchem Metafile die Bitmap aufgezeichnet werden soll.

- `/record:none` - Die Ausgabe des Icons wird nicht aufgezeichnet.
- `/record>window` - Das Icon wird im Window-Metafile, also im Hintergrund aufgezeichnet.
- `/record:page` - Das Icon wird im Page-Metafile, also im Vordergrund aufgezeichnet (Standardeinstellung).

Beispiel

```
win.DrawIcon 50 50 default  
Wait
```

```
win.Clear /all  
win.DrawIcon 50 50 warning /record>window
```

DrawLine

Syntax

```
DrawLine x1 y1 x2 y2 /record:xxxx
```

Beschreibung

Dieser Befehl zeichnet eine Linie zwischen zwei Punkten. Die Linienart kann zuvor mit dem Befehl SetPen definiert werden.

Fixe Parameter

x1

X-Koordinate des Startpunktes.

y1

Y-Koordinate des Startpunktes.

x2

X-Koordinate des Endpunktes.

y2

Y-Koordinate des Endpunktes.

Optionale Parameter

/record:xxxx

Innherhalb eines Fensters werden alle Ausgaben von Text und Grafik in zwei sogenannten Metafiles aufgezeichnet: im Window-Metafile und im Page-Metafile.

Mit diesem Parameter kann festgelegt werden, in welchem Metafile die Ausgabe der Linie aufgezeichnet werden soll. Dem Namensteil des Parameter folgt nach dem Doppelpunkt eine Angabe, in welchem Metafile die Bitmap aufgezeichnet werden soll.

`/record:none` - Die Ausgabe der Linie wird nicht aufgezeichnet.

`/record>window` - Die Linie wird im Window-Metafile, also im Hintergrund aufgezeichnet.

`/record:page` - Die Linie wird im Page-Metafile, also im Vordergrund aufgezeichnet (Standardeinstellung).

Beispiel

```
win.SetPen solid 0 lightblue  
win.DrawLine 50 50 450 350  
Wait
```

```
win.Clear /all  
win.SetPen dot 2 green  
win.DrawLine 50 50 450 350 /record>window
```

Querverweis

SetPen

DrawPoint

Syntax

```
DrawPoint x y /record:xxxx
```

Beschreibung

Dieser Befehl zeichnet einen Punkt. Farbe und Größe des Punktes werden durch den Befehl SetPen definiert

Fixe Parameter

x
X-Koordinate des Punktes.

y
Y-Koordinate des Punktes.

Optionale Parameter

/record:xxxx

Innherhalb eines Fensters werden alle Ausgaben von Text und Grafik in zwei sogenannten Metafiles aufgezeichnet: im Window-Metafile und im Page-Metafile.

Mit diesem Parameter kann festgelegt werden, in welchem Metafile die Ausgabe des Punktes aufgezeichnet werden soll. Dem Namensteil des Parameter folgt nach dem Doppelpunkt eine Angabe, in welchem Metafile die Bitmap aufgezeichnet werden soll.

`/record:none` - Die Ausgabe des Punktes wird nicht aufgezeichnet.
`/record>window` - Der Punkt wird im Window-Metafile, also im Hintergrund aufgezeichnet.
`/record:page` - Der Punkt wird im Page-Metafile, also im Vordergrund aufgezeichnet (Standardeinstellung).

Beispiel

```
win.SetPen solid 2 lightblue
win.DrawPoint 200 200
Wait

win.Clear /all
win.SetPen solid 2 green
win.DrawPoint 200 200 /record>window
```

Querverweis

[SetPen](#)

DrawRectangle

Syntax

```
DrawRectangle x y width height /xround:xx /yround:xx /record:xxxx
```

Beschreibung

Dieser Befehl zeichnet ein Rechteck. Die Linienart und das Füllmuster können zuvor mit den Befehlen SetPen und SetBrush definiert werden.

Fixe Parameter

x

X-Koordinate des linken oberen Punktes des Rechtecks.

y

Y-Koordinate des linken oberen Punktes des Rechtecks.

width

Breite des Rechtecks.

height

Höhe des Rechtecks.

Optionale Parameter

/xr:xx

/xround:xx

Durch diesen optionalen Parameter wird die Eckenverrundung des Rechtecks in horizontaler Richtung angegeben. Dem Namensteil folgt durch einen Doppelpunkt getrennt, der Wert für die Verrundung.

/yr:xx

/yround:xx

Durch diesen optionalen Parameter wird die Eckenverrundung des Rechtecks in vertikaler Richtung angegeben. Dem Namensteil folgt durch einen Doppelpunkt getrennt, der Wert für die Verrundung.

/record:xxxx

Innherhalb eines Fensters werden alle Ausgaben von Text und Grafik in zwei sogenannten Metafiles aufgezeichnet: im Window-Metafile und im Page-Metafile.

Mit diesem Parameter kann festgelegt werden, in welchem Metafile die Ausgabe des Rechtecks aufgezeichnet werden soll. Dem Namensteil des Parameter folgt nach dem Doppelpunkt eine Angabe, in welchem Metafile die Bitmap aufgezeichnet werden soll.

/record:none - Die Ausgabe des Rechtecks wird nicht aufgezeichnet.

/record>window - Das Rechteck wird im Window-Metafile, also im Hintergrund aufgezeichnet.

/record:page - Das Rechteck wird im Page-Metafile, also im Vordergrund aufgezeichnet (Standardeinstellung).

Beispiel

```
win.SetPen solid 0 lightblue  
win.SetBrush diagcross blue yellow  
win.DrawRectangle 50 50 400 300  
Wait
```

```
win.Clear  
win.DrawRectangle 50 50 400 300 /xround:40 /yround:40  
Wait
```

```
win.Clear /all
```

```
win.SetPen dashdot 2 green  
win.SetBrush b50 green lightcyan  
win.DrawRectangle 50 50 400 300 /record>window
```

Querverweis

[SetPen](#)
[SetBrush](#)

DrawText

Syntax

```
DrawText "....." /x:xx /y:xx /record:xxxx
```

Beschreibung

Mit diesem Befehl kann ein Text in ein Fenster ausgegeben werden. Werden durch die optionalen Parameter '/x' und '/y' keine Koordinaten für die Textausgabe gesetzt, wird der Text in der linken oberen Ecke des Fensters ausgegeben. Die Schriftart, mit welcher der Text ausgegeben werden soll, kann zuvor mit dem Befehl SetFont definiert werden.

Fixe Parameter

"....." - Auszugebender Text.

Optionale Parameter

/x:xx
/col:xx

Dem Namensteil folgt die X-Koordinate, an welcher der Text ausgegeben werden soll.

/y:xx
/row:xx

Dem Namensteil folgt die Y-Koordinate, an welcher der Text ausgegeben werden soll.

/record:xxxx

Innherhalb eines Fensters werden alle Ausgaben von Text und Grafik in zwei sogenannten Metafiles aufgezeichnet: im Window-Metafile und im Page-Metafile.

Mit diesem Parameter kann festgelegt werden, in welchem Metafile die Ausgabe des Textes aufgezeichnet werden soll. Dem Namensteil des Parameter folgt nach dem Doppelpunkt eine Angabe, in welchem Metafile die Bitmap aufgezeichnet werden soll.

/record:none - Die Ausgabe des Textes wird nicht aufgezeichnet.
/record>window - Der Text wird im Window-Metafile, also im Hintergrund aufgezeichnet.
/record:page - Der Text wird im Page-Metafile, also im Vordergrund aufgezeichnet (Standardeinstellung).

Beispiel

```
win.SetFont 6 36 lightred white /w2 /s /t  
win.DrawText "RSE-Author ist ein flexibles Autorensystem" /x:5 /y:2  
Wait  
  
win.Clear /all  
win.DrawText "RSE-Author ist ein flexibles Autorensystem" /x:5 /y:2 /record>window
```

Querverweis

SetFont

Enable

Syntax

win.Enable

Beschreibung

Nach Aufruf dieses Befehl ist das Fenster aktivierbar. Es erhält Eingaben, d.h. es kann mit der Maus angeklickt werden. Das gleiche gilt für alle Elemente (Schalter, Hyperlink, Hypertext usw.).

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

MainWindow.Enable ! Applikationsfenster ist aktivierbar

Querverweis

[Disable](#)

FirstPage

Syntax

FirstPage object /clear /noclear /paint /back /fade:xx:xx:xx /x:xx /y:xx

Beschreibung

Dieser Befehl lädt die erste Seite, deren Name einem bestimmten Zeichenmuster entspricht. Dieses Zeichenmuster besteht aus einem festen Namensteil, der bei allen zu einer Gruppe gehörenden Seiten gleich sein muß, sowie einem oder mehreren Fragezeichen.

Nehmen wir an, Sie haben in Ihrem Programm 150 Seiten angelegt, deren Bezeichnungen 'Page001', 'Page002' bis 'Page150' lauten.

Um die erste Seite (d.h. die Seite, deren Namen den alphanumerisch kleinsten Wert darstellt) zu laden, müssen Sie den Befehl 'FirstPage Page???' angeben. Die Fragezeichen geben den Teil des Seitenamens an, der die Numerierung enthält. Der Befehl lädt die Seite, deren Name im Bereich der Fragezeichen den kleinsten Wert enthält (also 001).

Sie können Ihre Seiten natürlich auch in unterschiedliche Gruppen aufteilen, die völlig unabhängig voneinander in verschiedenen Fenstern verwaltet werden können.

Wenn Sie in einem Programm z.B. 50 Seiten zum Thema Auto und 50 Seiten zum Thema Flugzeug haben, dann benennen Sie die Seiten einfach von 'Auto01' bis 'Auto50', sowie 'Flugzeug01' bis 'Flugzeug50'. Der Befehl 'FirstPage Auto??' lädt die erste Seite zum Thema Auto. Der Befehl 'FirstPage Flugzeug??' lädt die erste Seite zum Thema Flugzeug.

Fixe Parameter

object - Zeichenmuster der Seitengruppe

Optionale Parameter

/clear

Das Fenster wird vor dem Laden der Seite gelöscht (StandardEinstellung). Das Page-Metafile wird gelöscht.

/noclear

Das Fenster wird vor dem Laden der neuen Seite nicht gelöscht. Das Page-Metafile wird gelöscht.

/paint

Vor dem Laden der neuen Seite wird ein Repaint des Fensters ausgeführt. Das ist in Verbindung mit dem Parameter '/noclear' sinnvoll, wenn das Fenster eine Hintergrundbitmap (angelegt als Element des Fensterobjekts) enthält. Auf diese Art können Seiten ohne Flackereffekt vor einer Hintergrundbitmap durchgeblättert werden.

/back

Die Seite wird im Arbeitsspeicher aufgebaut und anschließend in das Fenster kopiert.

Während des Aufbaus bleibt die alte Seite erhalten. Dadurch verkürzt sich die Zeit, in der nach dem Löschen das leere Fenster zu sehen ist.

Wird eine Seite überblendet (optionaler Parameter '/fade'), dann erübrigt sich dieser Parameter, da eine Seite zwecks Überblendung grundsätzlich im Speicher aufgebaut werden muß.

/fade:effekt

/fade:effekt:groesse:pause

Durch Angabe dieses Parameters kann die Seite mittels eines Überblendeffekts in das Fenster kopiert werden. Dem optionalen Parameter wird dazu die Nummer des gewünschten Überblendeffekts übergeben. Folgende Effekte sind möglich:

- 0** - absolutes Kopieren
- 1** - überblenden
- 2** - von oben nach unten
- 3** - von unten nach oben
- 4** - von links nach rechts
- 5** - von rechts nach links
- 6** - ausgehend von der Mitte nach oben und unten
- 7** - ausgehend von der Mitte nach links und rechts
- 8** - von oben und unten zur Mitte hin
- 9** - von links und rechts zur Mitte hin
- 10** - von innen nach aussen zoomen
- 11** - von aussen nach innen zoomen
- 12** - von oben nach unten einrollen
- 13** - von unten nach oben einrollen
- 14** - von links nach rechts einrollen
- 15** - von rechts nach links einrollen

Zusätzlich zu der Art des Überblendeffekts können noch zwei weitere Parameter angegeben werden: 'groesse' und 'pause'. Je nach Effekt kann die Bedeutung dieser Parameter variieren.

Für den Effekt 1 (überblenden) gibt der Parameter 'groesse' die Kantenlänge in Pixeln für einen kopierten Block an. Für alle anderen Effekte gibt dieser Parameter an, wieviele Linien zwischen zwei Pausen kopiert werden sollen.

Der Parameter 'pause' gibt die Verzögerung zwischen zwei Kopiervorgängen an. Je größer dieser Wert ist, desto langsamer wird eine Grafik aufgeblendet.

/x:xx /y:xx

Standardmäßig wird eine Seite beim Laden in ein Fenster an den Koordinaten ausgegeben, die im Eigenschaftendialog des Seitenobjekts eingegeben wurden (meistens 0/0). Falls die Anforderung besteht, ein und dieselbe Seite an verschiedene Koordinaten zu laden, dann können die Standardkoordinaten durch die optionalen Parameter '/x' und '/y' verändert werden. Dem Namensteil des Parameters folgt durch den Doppelpunkt getrennt, der Wert für die jeweilige Koordinate.

Beispiel

```
win.FirstPage Auto??    ! Erste Seite zum Thema Auto laden  
win.FirstPage Flugzeug??    ! Erste Seite zum Thema Flugzeug laden
```

win.FirstPage Page??? /back ! Seite im Hintergrund aufbauen
win.FirstPage Page??? /noclear /paint ! Fenster nicht löschen, Neuzeichnen
anstoßen
win.FirstPage Page??? /fade:01:02:00 ! Fenster überblenden

Querverweis

LastPage
NextPage
PrevPage
LoadPage
BackPage
SelectPage

GetBitmap

Syntax

```
GetBitmap name /x:xx /y:xx /width:xx /height:xx
```

Beschreibung

Dieser Befehl erzeugt aus dem Inhalt des Fensters, auf das er angewendet wird, ein Bitmap-Objekt. Werden keine optionalen Parameter angegeben, dann wird die gesamte Ausgabefläche des Fensters in ein Bitmap umgewandelt. Über die optionalen Parameter '/x', '/y', '/width' und '/height' kann der Bereich, der in eine Bitmap umgewandelt werden soll, genauer definiert werden.

Fixe Parameter

name - Name des Bitmap-Objekts, das erzeugt werden soll.

Optionale Parameter

/x:xx
/col:xx

X-Koordinate des Bereiches, der in eine Bitmap umgewandelt werden soll.

/y:xx
/row:xx

Y-Koordinate des Bereiches, der in eine Bitmap umgewandelt werden soll.

/width:xx

Breite des Bereiches, der in eine Bitmap umgewandelt werden soll.

/height:xx

Höhe des Bereiches, der in eine Bitmap umgewandelt werden soll.

Beispiel

```
win.GetBitmap bmp01  
Wait  
win.Clear /all  
win.DrawBitmap bmp01 0 0  
  
win.GetBitmap bmp01 /x:50 /y:50 /width:300 /height:200
```

GrabFocus

Syntax

GrabFocus

Beschreibung

Durch Aufruf dieses Befehls erhält das Fenster den Fokus, d.h. das Fenster erhält in der Folge alle Eingaben (Tastendrucke usw.)

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

win.GrabFocus

Hide

Syntax

Hide

Beschreibung

Das Fenster ist nach Aufruf dieses Befehls nicht mehr sichtbar, es existiert aber sowohl auf der Ebene des Laufzeitsystems, als auch auf der Ebene des zugrundeliegenden Fenstersystems weiter.

Alle Elemente des Fensters werden ebenfalls unsichtbar. Bei Aufruf des Befehls Show werden diese wieder angezeigt.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

win.Hide

Querverweis

Show

LastPage

Syntax

LastPage object /clear /noclear /paint /back /fade:xx:xx:xx /x:xx /y:xx

Beschreibung

Dieser Befehl lädt die letzte Seite, deren Name einem bestimmten Zeichenmuster entspricht. Dieses Zeichenmuster besteht aus einem festen Namensteil, der bei allen zu einer Gruppe gehörenden Seiten gleich sein muß, sowie einem oder mehreren Fragezeichen.

Nehmen wir an, Sie haben in Ihrem Programm 150 Seiten angelegt, deren Bezeichnungen 'Page001', 'Page002' bis 'Page150' lauten.

Um die letzte Seite (d.h. die Seite, deren Namen den alphanumerisch größten Wert darstellt) zu laden, müssen Sie den Befehl 'LastPage Page???' angeben. Die Fragezeichen geben den Teil des Seitenamens an, der die Numerierung enthält. Der Befehl lädt die Seite, deren Name im Bereich der Fragezeichen den größten Wert enthält (also 150).

Sie können Ihre Seiten natürlich auch in unterschiedliche Gruppen aufteilen, die völlig unabhängig voneinander in verschiedenen Fenstern verwaltet werden können.

Wenn Sie in einem Programm z.B. 50 Seiten zum Thema Auto und 50 Seiten zum Thema Flugzeug haben, dann benennen Sie die Seiten einfach von 'Auto01' bis 'Auto50', sowie 'Flugzeug01' bis 'Flugzeug50'. Der Befehl 'LastPage Auto??' lädt die letzte Seite zum Thema Auto. Der Befehl 'LastPage Flugzeug??' lädt die letzte Seite zum Thema Flugzeug.

Fixe Parameter

object - Zeichenmuster der Seitengruppe

Optionale Parameter

/clear

Das Fenster wird vor dem Laden der Seite gelöscht (Standardeinstellung). Das Page-Metafile wird gelöscht.

/noclear

Das Fenster wird vor dem Laden der neuen Seite nicht gelöscht. Das Page-Metafile wird gelöscht.

/paint

Vor dem Laden der neuen Seite wird ein Repaint des Fensters ausgeführt. Das ist in Verbindung mit dem Parameter '/noclear' sinnvoll, wenn das Fenster eine Hintergrundbitmap (angelegt als Element des Fensterobjekts) enthält. Auf diese Art können Seiten ohne Flackereffekt vor einer Hintergrundbitmap durchgeblättert werden.

/back

Die Seite wird im Arbeitsspeicher aufgebaut und anschließend in das Fenster kopiert.

Während des Aufbaus bleibt die alte Seite erhalten. Dadurch verkürzt sich die Zeit, in der nach dem Löschen das leere Fenster zu sehen ist.

Wird eine Seite überblendet (optionaler Parameter '/fade'), dann erübrigt sich dieser Parameter, da eine Seite zwecks Überblendung grundsätzlich im Speicher aufgebaut werden muß.

/fade:effekt

/fade:effekt:groesse:pause

Durch Angabe dieses Parameters kann die Seite mittels eines Überblendeffekts in das Fenster kopiert werden. Dem optionalen Parameter wird dazu die Nummer des gewünschten Überblendeffekts übergeben. Folgende Effekte sind möglich:

- 0** - absolutes Kopieren
- 1** - überblenden
- 2** - von oben nach unten
- 3** - von unten nach oben
- 4** - von links nach rechts
- 5** - von rechts nach links
- 6** - ausgehend von der Mitte nach oben und unten
- 7** - ausgehend von der Mitte nach links und rechts
- 8** - von oben und unten zur Mitte hin
- 9** - von links und rechts zur Mitte hin
- 10** - von innen nach aussen zoomen
- 11** - von aussen nach innen zoomen
- 12** - von oben nach unten einrollen
- 13** - von unten nach oben einrollen
- 14** - von links nach rechts einrollen
- 15** - von rechts nach links einrollen

Zusätzlich zu der Art des Überblendeffekts können noch zwei weitere Parameter angegeben werden: 'groesse' und 'pause'. Je nach Effekt kann die Bedeutung dieser Parameter variieren.

Für den Effekt 1 (überblenden) gibt der Parameter 'groesse' die Kantenlänge in Pixeln für einen kopierten Block an. Für alle anderen Effekte gibt dieser Parameter an, wieviele Linien zwischen zwei Pausen kopiert werden sollen.

Der Parameter 'pause' gibt die Verzögerung zwischen zwei Kopiervorgängen an. Je größer dieser Wert ist, desto langsamer wird eine Grafik aufgeblendet.

/x:xx /y:xx

Standardmäßig wird eine Seite beim Laden in ein Fenster an den Koordinaten ausgegeben, die im Eigenschaftendialog des Seitenobjekts eingegeben wurden (meistens 0/0). Falls die Anforderung besteht, ein und dieselbe Seite an verschiedene Koordinaten zu laden, dann können die Standardkoordinaten durch die optionalen Parameter '/x' und '/y' verändert werden. Dem Namensteil des Parameters folgt durch den Doppelpunkt getrennt, der Wert für die jeweilige Koordinate.

Beispiel

```
win.LastPage Auto??    ! Letzte Seite zum Thema Auto laden  
win.LastPage Flugzeug??    ! Letzte Seite zum Thema Flugzeug laden
```

win.LastPage Page??? /back ! Seite im Hintergrund aufbauen
win.LastPage Page??? /noclear /paint ! Fenster nicht löschen, Neuzeichnen
anstoßen
win.LastPage Page??? /fade:01:02:00 ! Fenster überblenden

Querverweis

[FirstPage](#)
[NextPage](#)
[PrevPage](#)
[LoadPage](#)
[BackPage](#)
[SelectPage](#)

LoadPage

Syntax

LoadPage object /add /clear /noclear /paint /back /fade:xx:xx:xx /x:xx /y:xx

Beschreibung

Dieser Befehl lädt ein Seitenobjekt in ein Fenster.

Fixe Parameter

object - Name des Seitenobjekts

Optionale Parameter

/add

Dieser Parameter gibt an, daß die Seite der aktuell geladenen Seite hinzugefügt werden soll. Das Fenster wird in dem Fall nicht gelöscht (die Parameter '/clear' und '/noclear' werden ignoriert) und die neue Seite wird nicht als aktuelle Seite definiert. Das ist z.B. sinnvoll, wenn nach dem Laden einer Seite mit einer kurzen Zeitverzögerung weitere Bildelemente (z.B. Titelzeilen) mittels Überblendeffekten eingefügt werden sollen. Seiten, die mittels dieses Parameters einer Hauptseite hinzugefügt werden, sollten möglichst im PageEntry-Handler der Hauptseite geladen werden.

/clear

Das Fenster wird vor dem Laden der Seite gelöscht (Standardeinstellung). Das Page-Metafile wird gelöscht.

/noclear

Das Fenster wird vor dem Laden der neuen Seite nicht gelöscht. Das Page-Metafile wird gelöscht.

/paint

Vor dem Laden der neuen Seite wird ein Repaint des Fensters ausgeführt. Das ist in Verbindung mit dem Parameter '/noclear' sinnvoll, wenn das Fenster eine Hintergrundbitmap (angelegt als Element des Fensterobjekts) enthält. Auf diese Art können Seiten ohne Flackereffekt vor einer Hintergrundbitmap durchgeblättert werden.

/back

Die Seite wird im Arbeitsspeicher aufgebaut und anschließend in das Fenster kopiert. Während des Aufbaus bleibt die alte Seite erhalten. Dadurch verkürzt sich die Zeit, in der nach dem Löschen das leere Fenster zu sehen ist.

Wird eine Seite überblendet (optionaler Parameter '/fade'), dann erübrigt sich dieser Parameter, da eine Seite zwecks Überblendung grundsätzlich im Speicher aufgebaut werden muß.

/fade:effekt

/fade:effekt:groesse:pause

Durch Angabe dieses Parameters kann die Seite mittels eines Überblendeffekts in das Fenster kopiert werden. Dem optionalen Parameter wird dazu die Nummer des gewünschten Überblendeffekts übergeben. Folgende Effekte sind möglich:

- 0** - absolutes Kopieren
- 1** - überblenden
- 2** - von oben nach unten
- 3** - von unten nach oben
- 4** - von links nach rechts
- 5** - von rechts nach links
- 6** - ausgehend von der Mitte nach oben und unten
- 7** - ausgehend von der Mitte nach links und rechts
- 8** - von oben und unten zur Mitte hin
- 9** - von links und rechts zur Mitte hin
- 10** - von innen nach aussen zoomen
- 11** - von aussen nach innen zoomen
- 12** - von oben nach unten einrollen
- 13** - von unten nach oben einrollen
- 14** - von links nach rechts einrollen
- 15** - von rechts nach links einrollen

Zusätzlich zu der Art des Überblendeffekts können noch zwei weitere Parameter angegeben werden: 'groesse' und 'pause'. Je nach Effekt kann die Bedeutung dieser Parameter variieren.

Für den Effekt 1 (überblenden) gibt der Parameter 'groesse' die Kantenlänge in Pixeln für einen kopierten Block an. Für alle anderen Effekte gibt dieser Parameter an, wieviele Linien zwischen zwei Pausen kopiert werden sollen.

Der Parameter 'pause' gibt die Verzögerung zwischen zwei Kopiervorgängen an. Je größer dieser Wert ist, desto langsamer wird eine Grafik aufgeblendet.

/x:xx /y:xx

Standardmäßig wird eine Seite beim Laden in ein Fenster an den Koordinaten ausgegeben, die im Eigenschaftendialog des Seitenobjekts eingegeben wurden (meistens 0/0). Falls die Anforderung besteht, ein und dieselbe Seite an verschiedene Koordinaten zu laden, dann können die Standardkoordinaten durch die optionalen Parameter '/x' und '/y' verändert werden. Dem Namensteil des Parameters folgt durch den Doppelpunkt getrennt, der Wert für die jeweilige Koordinate.

Beispiel

```
win1.LoadPage Auto15
win2.LoadPage Flugzeug36
```

```
win.LoadPage Page532 /back ! Seite im Hintergrund aufbauen
win.LoadPage Page532 /noclear /paint ! Fenster nicht löschen, Neuzeichen
anstoßen
win.LoadPage Page532 /fade:01:02:00 ! Fenster überblenden
```

```
win.LoadPage Page532
Delay 1000 ! 1 Sekunde warten
win.LoadPage Page532Title01 /add /fade:01:02:00 ! Titelzeile-1 einblenden
```

Delay 1000 ! 1 Sekunde warten
win.LoadPage Page532Title02 /add /fade:01:02:00 ! Titelzeile-2 einblenden

Querverweis

[FirstPage](#)
[LastPage](#)
[NextPage](#)
[PrevPage](#)
[BackPage](#)
[SelectPage](#)

Mark

Syntax

Mark /field /hypertext /style:xx:xx:xx /invert

Beschreibung

Dieser Befehl markiert alle Hyperlink- und/oder Hypertextfelder im angegebenen Fenster.

Standardmäßig werden alle Hyperlink- und Hypertextfelder im Fenster markiert. Durch die optionalen Parameter '/field' und '/hypertext' kann genauer definiert werden, auf welchen Feldtyp der Befehl wirken soll.

Wird die Art der Markierung nicht über die optionalen Parameter '/style' oder '/invert' angegeben, dann werden die Felder durch ein Rechteck mit der über den Befehl SetMarkStyle eingestellten Linienart markiert.

Fixe Parameter

keine

Optionale Parameter

/field

Nur die Hyperlinkfelder im Fenster werden markiert.

/hypertext

Nur die Hypertextfelder im Fenster werden markiert.

/style:xx:xx:xx

Als Markierung wird ein Rechteck verwendet. Durch Doppelpunkt getrennt, folgen die Werte für die Linienart, die Liniendicke und die Farbe der Linie (äquivalent zu den Einstellungen durch den Befehl SetPen). Beachten Sie bitte, daß die Markierung im XOR-Modus in den Hintergrund eingezeichnet wird. Die eingestellte Farbe erscheint deshalb nur auf schwarzem Hintergrund. Auf andersfarbigem Hintergrund kann es zu Abweichungen kommen.

/invert

Die Felder und/oder Hypertextfelder werden durch Invertierung markiert.

Beispiel

```
win.Mark  
win.Mark /field  
win.Mark /hypertext  
win.Mark /field /style:solid:0:green  
win.Mark /hypertext /invert
```

Querverweis

SetMarkStyle
Unmark

Maximize

Syntax

Maximize

Beschreibung

Mit diesem Befehl kann ein Standardfenster maximiert werden. Bei allen anderen Fenstertypen bewirkt dieser Befehl nichts. Ist das Fenster nicht sichtbar, so erscheint das Fenster erst nach dem nächsten Aufruf des Befehls Show als maximiertes Fenster. Ist das Fenster bereits maximiert, hat diesr Befehl keine Auswirkung.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

win.Maximize

Querverweis

Minimize
Restore

Minimize

Syntax

Minimize

Beschreibung

Mit diesem Befehl kann ein Standardfenster minimiert werden. Bei allen anderen Fenstertypen bewirkt dieser Befehl nichts. Ist das Fenster nicht sichtbar, so erscheint das Fenster erst nach dem nächsten Aufruf des Befehls Show als minimiertes Fenster. Ist das Fenster bereits minimiert, hat diesr Befehl keine Auswirkung.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

win.Minimize

Querverweis

Maximize
Restore

NextPage

Syntax

NextPage object /loop /clear /noclear /paint /back /fade:xx:xx:xx /x:xx /y:xx

Beschreibung

Beim Laden einer Seite mit den Befehlen LoadPage, FirstPage und LastPage speichert das Laufzeitsystem den Namen dieser Seite. Mit dem Befehl 'NextPage' kann dann die nächste Seite, deren Name dem gleichen Zeichenmuster entspricht, geladen werden. Dieses Zeichenmuster besteht aus einem festen Namensteil, der bei allen zu einer Gruppe gehörenden Seiten gleich sein muß, sowie einem oder mehreren Fragezeichen.

Nehmen wir an, Sie haben in Ihrem Programm 150 Seiten angelegt, deren Bezeichnungen 'Page001', 'Page002' bis 'Page150' lauten. Sie haben dann mit dem Befehl 'LoadPage' die Seite mit der Bezeichnung 'Page032' geladen. Durch Angabe des Befehls 'NextPage Page???' wird dann die Seite mit der Bezeichnung 'Page033' geladen. Ein weiterer Aufruf von 'NextPage' lädt die Seite mit der Bezeichnung 'Page034' usw. Die Fragezeichen geben den Teil des Seitenamens an, der die Numerierung enthält. Der Befehl lädt die Seite, deren Name im Bereich der Fragezeichen den nächstgrößeren Wert enthält (also 033, 034,).

Sie können Ihre Seiten natürlich auch in unterschiedliche Gruppen aufteilen, die völlig unabhängig voneinander in verschiedenen Fenstern verwaltet werden können.

Wenn Sie in einem Programm z.B. 50 Seiten zum Thema Auto und 50 Seiten zum Thema Flugzeug haben, dann benennen Sie die Seiten einfach von 'Auto01' bis 'Auto50', sowie 'Flugzeug01' bis 'Flugzeug50'. Der Befehl 'NextPage Auto??' erlaubt das Durchblättern aller Seiten zum Thema Auto. Der Befehl 'NextPage Flugzeug??' blättert alle Seiten zum Thema Flugzeug durch.

Falls die Seite mit der größten Bezeichnung bereits in ein Fenster geladen ist, dann wird jeder weitere Aufruf von 'NextPage' ignoriert, es sei denn, der optionale Parameter '/loop' wurde angegeben.

Fixe Parameter

object - Zeichenmuster der Seitengruppe

Optionale Parameter

/loop

Ist dieser Parameter angegeben, dann wird beim Erreichen der letzten Seite automatisch wieder auf die erste Seite verzweigt. Die Seiten können also in einer Endlosschleife durchgeblättert werden.

/clear

Das Fenster wird vor dem Laden der Seite gelöscht (Standardeinstellung). Das Page-Metafile wird gelöscht.

/noclear

Das Fenster wird vor dem Laden der neuen Seite nicht gelöscht. Das Page-Metafile wird gelöscht.

/paint

Vor dem Laden der neuen Seite wird ein Repaint des Fensters ausgeführt. Das ist in Verbindung mit dem Parameter '/noclear' sinnvoll, wenn das Fenster eine Hintergrundbitmap (angelegt als Element des Fensterobjekts) enthält. Auf diese Art können Seiten ohne Flackereffekt vor einer Hintergrundbitmap durchgeblättert werden.

/back

Die Seite wird im Arbeitsspeicher aufgebaut und anschließend in das Fenster kopiert. Während des Aufbaus bleibt die alte Seite erhalten. Dadurch verkürzt sich die Zeit, in der nach dem Löschen das leere Fenster zu sehen ist.

Wird eine Seite überblendet (optionaler Parameter '/fade'), dann erübrigt sich dieser Parameter, da eine Seite zwecks Überblendung grundsätzlich im Speicher aufgebaut werden muß.

/fade:effekt

/fade:effekt:groesse:pause

Durch Angabe dieses Parameters kann die Seite mittels eines Überblendeffekts in das Fenster kopiert werden. Dem optionalen Parameter wird dazu die Nummer des gewünschten Überblendeffekts übergeben. Folgende Effekte sind möglich:

- 0** - absolutes Kopieren
- 1** - überblenden
- 2** - von oben nach unten
- 3** - von unten nach oben
- 4** - von links nach rechts
- 5** - von rechts nach links
- 6** - ausgehend von der Mitte nach oben und unten
- 7** - ausgehend von der Mitte nach links und rechts
- 8** - von oben und unten zur Mitte hin
- 9** - von links und rechts zur Mitte hin
- 10** - von innen nach aussen zoomen
- 11** - von aussen nach innen zoomen
- 12** - von oben nach unten einrollen
- 13** - von unten nach oben einrollen
- 14** - von links nach rechts einrollen
- 15** - von rechts nach links einrollen

Zusätzlich zu der Art des Überblendeffekts können noch zwei weitere Parameter angegeben werden: 'groesse' und 'pause'. Je nach Effekt kann die Bedeutung dieser Parameter variieren.

Für den Effekt 1 (überblenden) gibt der Parameter 'groesse' die Kantenlänge in Pixeln für einen kopierten Block an. Für alle anderen Effekte gibt dieser Parameter an, wieviele Linien zwischen zwei Pausen kopiert werden sollen.

Der Parameter 'pause' gibt die Verzögerung zwischen zwei Kopiervorgängen an. Je größer dieser Wert ist, desto langsamer wird eine Grafik aufgeblendet.

/x:xx **/y:xx**

Standardmäßig wird eine Seite beim Laden in ein Fenster an den Koordinaten ausgegeben, die im Eigenschaftendialog des Seitenobjekts eingegeben wurden (meistens 0/0). Falls die Anforderung besteht, ein und dieselbe Seite an verschiedene Koordinaten zu laden, dann können die Standardkoordinaten durch die optionalen Parameter '/x' und '/y' verändert werden. Dem Namensteil des Parameters folgt durch den Doppelpunkt getrennt, der Wert für die jeweilige Koordinate.

Beispiel

```
win.NextPage Auto?? /loop    ! Nächste Seite zum Thema Auto laden
win.NextPage Flugzeug?? /loop    ! Nächste Seite zum Thema Flugzeug laden

win.NextPage Page??? /back    ! Seite im Hintergrund aufbauen
win.NextPage Page??? /noclear /paint    ! Fenster nicht löschen, Neuzeichnen
anstoßen
win.NextPage Page??? /fade:01:02:00    ! Fenster überblenden
```

Querverweis

[FirstPage](#)
[LastPage](#)
[PrevPage](#)
[LoadPage](#)
[BackPage](#)
[SelectPage](#)

Paint

Syntax

Paint

Beschreibung

Dieser Befehl bewirkt das Neuzeichnen eines Fensters.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

win.Paint

PlayVideo

Syntax

```
PlayVideo filename /loop /from:xxx /to:xxx /paint:xxxx
```

Beschreibung

Dieser Befehl spielt eine Video-Datei im angegebenen Fenster ab. Das Video wird dabei automatisch an die Größe des Fensters angepaßt

Fixe Parameter

filename - Dateiname der Video-Datei, die abgespielt werden soll.

Optionale Parameter

/loop

Das Abspielen der Video-Datei wird bei Erreichen des Dateiendes neu angestoßen. Die Video-Datei wird solange abgespielt, bis das Abspielen durch den Befehl StopVideo explizit angehalten oder das Programm beendet wird.

/from:xxx

Startposition in Millisekunden, ab der die Video-Datei abgespielt werden soll.

/to:xxx

Endposition in Millisekunden, bis zu der die Video-Datei abgespielt werden soll.

/paint

/paint:xxxx

Die Angabe dieses Parameters bewirkt, daß nach dem kompletten Abspielen der Video-Datei ein Neuzeichnen des Fensters, in dem das Video abgespielt wurde, angestoßen wird. Dadurch werden die Ursprungsinhalte des Fensters (z.B. ein Bitmap, welches das erste Bild des Videos enthält) wieder hergestellt. Dem Parameter kann als Wert außerdem noch eine Verzögerung in Millisekunden mitgegeben werden, die zwischen dem Beenden des Abspielens und dem Neuzeichnen des Fensters liegen soll. Dadurch können abrupte Übergänge vermieden werden. Es sieht i.d.R. nicht sehr schön aus, wenn sofort nach dem letzten Videobild der Fensterinhalt umspringt.

Beispiel

```
win.PlayVideo test.avi
win.PlayVideo test.avi /loop
win.PlayVideo test.avi /from:10000 /to:20000
win.PlayVideo test.avi /paint
win.PlayVideo test.avi /paint:1000
```

Querverweis

StopVideo

PlayAudio

VideoPlayer
Mci

PrevPage

Syntax

PrevPage object /loop /clear /noclear /paint /back /fade:xx:xx:xx /x:xx /y:xx

Beschreibung

Beim Laden einer Seite mit den Befehlen LoadPage, FirstPage und LastPage speichert das Laufzeitsystem den Namen dieser Seite. Mit dem Befehl 'PrevPage' kann dann die vorige Seite, deren Name dem gleichen Zeichenmuster entspricht, geladen werden. Dieses Zeichenmuster besteht aus einem festen Namensteil, der bei allen zu einer Gruppe gehörenden Seiten gleich sein muß, sowie einem oder mehreren Fragezeichen.

Nehmen wir an, Sie haben in Ihrem Programm 150 Seiten angelegt, deren Bezeichnungen 'Page001', 'Page002' bis 'Page150' lauten. Sie haben dann mit dem Befehl 'LoadPage' die Seite mit der Bezeichnung 'Page032' geladen. Durch Angabe des Befehls 'PrevPage Page???' wird dann die Seite mit der Bezeichnung 'Page031' geladen. Ein weiterer Aufruf von 'PrevPage' lädt die Seite mit der Bezeichnung 'Page030' usw. Die Fragezeichen geben den Teil des Seitenamens an, der die Numerierung enthält. Der Befehl lädt die Seite, deren Name im Bereich der Fragezeichen den nächstkleineren Wert enthält (also 031, 030,).

Sie können Ihre Seiten natürlich auch in unterschiedliche Gruppen aufteilen, die völlig unabhängig voneinander in verschiedenen Fenstern verwaltet werden können.

Wenn Sie in einem Programm z.B. 50 Seiten zum Thema Auto und 50 Seiten zum Thema Flugzeug haben, dann benennen Sie die Seiten einfach von 'Auto01' bis 'Auto50', sowie 'Flugzeug01' bis 'Flugzeug50'. Der Befehl 'NextPage Auto??' erlaubt das Durchblättern aller Seiten zum Thema Auto. Der Befehl 'NextPage Flugzeug??' blättert alle Seiten zum Thema Flugzeug durch.

Falls die Seite mit der kleinsten Bezeichnung bereits in ein Fenster geladen ist, dann wird jeder weitere Aufruf von 'PrevPage' ignoriert, es sei denn, der optionale Parameter '/loop' wurde angegeben.

Fixe Parameter

object - Zeichenmuster der Seitengruppe

Optionale Parameter

/loop

Ist dieser Parameter angegeben, dann wird beim Erreichen der ersten Seite automatisch wieder auf die letzte Seite verzweigt. Die Seiten können also in einer Endlosschleife durchgeblättert werden.

/clear

Das Fenster wird vor dem Laden der Seite gelöscht (Standardeinstellung). Das Page-Metafile wird gelöscht.

/noclear

Das Fenster wird vor dem Laden der neuen Seite nicht gelöscht. Das Page-Metafile wird gelöscht.

/paint

Vor dem Laden der neuen Seite wird ein Repaint des Fensters ausgeführt. Das ist in Verbindung mit dem Parameter '/noclear' sinnvoll, wenn das Fenster eine Hintergrundbitmap (angelegt als Element des Fensterobjekts) enthält. Auf diese Art können Seiten ohne Flackereffekt vor einer Hintergrundbitmap durchgeblättert werden.

/back

Die Seite wird im Arbeitsspeicher aufgebaut und anschließend in das Fenster kopiert. Während des Aufbaus bleibt die alte Seite erhalten. Dadurch verkürzt sich die Zeit, in der nach dem Löschen das leere Fenster zu sehen ist.

Wird eine Seite überblendet (optionaler Parameter '/fade'), dann erübrigt sich dieser Parameter, da eine Seite zwecks Überblendung grundsätzlich im Speicher aufgebaut werden muß.

/fade:effekt

/fade:effekt:groesse:pause

Durch Angabe dieses Parameters kann die Seite mittels eines Überblendeffekts in das Fenster kopiert werden. Dem optionalen Parameter wird dazu die Nummer des gewünschten Überblendeffekts übergeben. Folgende Effekte sind möglich:

- 0** - absolutes Kopieren
- 1** - überblenden
- 2** - von oben nach unten
- 3** - von unten nach oben
- 4** - von links nach rechts
- 5** - von rechts nach links
- 6** - ausgehend von der Mitte nach oben und unten
- 7** - ausgehend von der Mitte nach links und rechts
- 8** - von oben und unten zur Mitte hin
- 9** - von links und rechts zur Mitte hin
- 10** - von innen nach aussen zoomen
- 11** - von aussen nach innen zoomen
- 12** - von oben nach unten einrollen
- 13** - von unten nach oben einrollen
- 14** - von links nach rechts einrollen
- 15** - von rechts nach links einrollen

Zusätzlich zu der Art des Überblendeffekts können noch zwei weitere Parameter angegeben werden: 'groesse' und 'pause'. Je nach Effekt kann die Bedeutung dieser Parameter variieren.

Für den Effekt 1 (überblenden) gibt der Parameter 'groesse' die Kantenlänge in Pixeln für einen kopierten Block an. Für alle anderen Effekte gibt dieser Parameter an, wieviele Linien zwischen zwei Pausen kopiert werden sollen.

Der Parameter 'pause' gibt die Verzögerung zwischen zwei Kopiervorgängen an. Je größer dieser Wert ist, desto langsamer wird eine Grafik aufgeblendet.

/x:xx **/y:xx**

Standardmäßig wird eine Seite beim Laden in ein Fenster an den Koordinaten ausgegeben, die im Eigenschaftendialog des Seitenobjekts eingegeben wurden (meistens 0/0). Falls die Anforderung besteht, ein und dieselbe Seite an verschiedene Koordinaten zu laden, dann können die Standardkoordinaten durch die optionalen Parameter '/x' und '/y' verändert werden. Dem Namensteil des Parameters folgt durch den Doppelpunkt getrennt, der Wert für die jeweilige Koordinate.

Beispiel

```
win.PrevPage Auto?? /loop    ! Vorige Seite zum Thema Auto laden
win.PrevPage Flugzeug?? /loop    ! Vorige Seite zum Thema Flugzeug laden

win.PrevPage Page??? /back    ! Seite im Hintergrund aufbauen
win.PrevPage Page??? /noclear /paint    ! Fenster nicht löschen, Neuzeichnen
anstoßen
win.PrevPage Page??? /fade:01:02:00    ! Fenster überblenden
```

Querverweis

[FirstPage](#)
[LastPage](#)
[NextPage](#)
[LoadPage](#)
[BackPage](#)
[SelectPage](#)

Restore

Syntax

Restore

Beschreibung

Mit diesem Befehl kann ein Standardfenster vom minimierten bzw. maximierten Zustand auf die vorherige Größe eingestellt werden. Ist das Fenster nicht sichtbar, so erscheint das Fenster erst nach dem nächsten Aufruf des Befehls Show in seiner ursprünglichen Größe. Ist das Fenster weder maximiert noch minimiert, so hat der Aufruf dieses Befehls keine Wirkung.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

win.Restore

Querverweis

Maximize
Minimize

Scroll

Syntax

```
Scroll dx dy /xstep:xx /ystep:xx
```

Beschreibung

Dieser Befehl verschiebt den Inhalt des angegebenen Fensters. Durch Angabe der fixen Parameter wird um einen festen Betrag verschoben. Zusätzlich kann über optionale Parameter eingestellt werden, daß innerhalb einer Schleife der Fensterinhalt mehrmals verschoben werden soll (Laufbild-Effekt).

Fixe Parameter

dx

Betrag, um den in horizontaler Richtung gescrollt werden soll. Ein positiver Wert schiebt den Fensterinhalt nach links. Ein negativer Wert schiebt den Fensterinhalt nach rechts.

dy

Betrag, um den in vertikaler Richtung gescrollt werden soll. Ein positiver Wert schiebt den Fensterinhalt nach oben. Ein negativer Wert schiebt den Fensterinhalt nach unten.

Optionale Parameter

/xs
/xstep

Anzahl der Wiederholungen, um die in horizontaler Richtung gescrollt werden soll. Angenommen, Sie möchten den Fensterinhalt um 160 Pixel nach links verschieben. Das soll in 20 Stufen zu je 8 Pixel geschehen, um einen Laufbild-Effekt zu erreichen. Der erforderliche Aufruf lautet dann: 'Scroll 8 0 /xstep:20

/ys
/ystep

Anzahl der Wiederholungen, um die in vertikaler Richtung gescrollt werden soll. Angenommen, Sie möchten den Fensterinhalt um 160 Pixel nach oben verschieben. Das soll in 20 Stufen zu je 8 Pixel geschehen, um einen Laufbild-Effekt zu erreichen. Der erforderliche Aufruf lautet dann: 'Scroll 0 8 /ystep:20

Beispiel

```
win.Scroll 32 0 ! Fensterinhalt um 32 Pixel nach links verschieben
win.Scroll -32 0 ! Fensterinhalt um 32 Pixel nach rechts verschieben
win.Scroll 0 32 ! Fensterinhalt um 32 Pixel nach oben verschieben
win.Scroll 0 -32 ! Fensterinhalt um 32 Pixel nach unten verschieben
win.Scroll 8 0 /xstep:20 ! Fensterinhalt in 20 Schritten zu 8 Pixel nach links
verschieben
win.Scroll -8 0 /xstep:20 ! Fensterinhalt in 20 Schritten zu 8 Pixel nach rechts
verschieben
win.Scroll 0 8 /ystep:20 ! Fensterinhalt in 20 Schritten zu 8 Pixel nach oben
verschieben
```

win.Scroll 0 -8 /ystep:20 ! Fensterinhalt in 20 Schritten zu 8 Pixel nach unten verschieben

Querverweis

SetOrigin

SelectPage

Syntax

SelectPage object width height /clear /noclear /paint /back /fade:xx:xx:xx /x:xx /y:xx

Beschreibung

Jeder Seite kann während der Erstellung in der Entwicklungsumgebung im Eigenschaften-Dialog ein Suchbegriff zugeordnet werden. Der Befehl 'SelectPage' blendet ein Auswahlfenster auf, in dem die Suchbegriffe aller Seiten aufgelistet werden, deren Name dem angegebenen Zeichenmuster entspricht. Dieses Zeichenmuster besteht aus einem festen Namensteil, der bei allen zu einer Gruppe gehörenden Seiten gleich sein muß, sowie einem oder mehreren Fragezeichen.

Nehmen wir an, Sie haben in Ihrem Programm 150 Seiten angelegt, deren Bezeichnungen 'Page001', 'Page002' bis 'Page150' lauten. Jeder dieser Seiten haben Sie einen Suchbegriff zugeordnet. Durch den Befehl 'SelectPage Page???' werden alle 150 Suchbegriffe in einem Dialogfenster aufgeblendet. Der Anwender kann sich wahlfrei die Seite aussuchen, die er sich ansehen möchte. Durch Bestätigung der ausgewählten Seite wird diese dann geladen.

Fixe Parameter

object - Zeichenmuster der Seitengruppe
width - Breite des Auswahlfensters
height - Höhe des Auswahlfensters

Optionale Parameter

/clear

Das Fenster wird vor dem Laden der Seite gelöscht (Standardeinstellung). Das Page-Metafile wird gelöscht.

/noclear

Das Fenster wird vor dem Laden der neuen Seite nicht gelöscht. Das Page-Metafile wird gelöscht.

/paint

Vor dem Laden der neuen Seite wird ein Repaint des Fensters ausgeführt. Das ist in Verbindung mit dem Parameter '/noclear' sinnvoll, wenn das Fenster eine Hintergrundbitmap (angelegt als Element des Fensterobjekts) enthält. Auf diese Art können Seiten ohne Flackereffekt vor einer Hintergrundbitmap durchgeblättert werden.

/back

Die Seite wird im Arbeitsspeicher aufgebaut und anschließend in das Fenster kopiert. Während des Aufbaus bleibt die alte Seite erhalten. Dadurch verkürzt sich die Zeit, in der nach dem Löschen das leere Fenster zu sehen ist.

Wird eine Seite überblendet (optionaler Parameter '/fade'), dann erübrigt sich dieser

Parameter, da eine Seite zwecks Überblendung grundsätzlich im Speicher aufgebaut werden muß.

/fade:effekt

/fade:effekt:groesse:pause

Durch Angabe dieses Parameters kann die Seite mittels eines Überblendeffekts in das Fenster kopiert werden. Dem optionalen Parameter wird dazu die Nummer des gewünschten Überblendeffekts übergeben. Folgende Effekte sind möglich:

- 0** - absolutes Kopieren
- 1** - überblenden
- 2** - von oben nach unten
- 3** - von unten nach oben
- 4** - von links nach rechts
- 5** - von rechts nach links
- 6** - ausgehend von der Mitte nach oben und unten
- 7** - ausgehend von der Mitte nach links und rechts
- 8** - von oben und unten zur Mitte hin
- 9** - von links und rechts zur Mitte hin
- 10** - von innen nach aussen zoomen
- 11** - von aussen nach innen zoomen
- 12** - von oben nach unten einrollen
- 13** - von unten nach oben einrollen
- 14** - von links nach rechts einrollen
- 15** - von rechts nach links einrollen

Zusätzlich zu der Art des Überblendeffekts können noch zwei weitere Parameter angegeben werden: 'groesse' und 'pause'. Je nach Effekt kann die Bedeutung dieser Parameter variieren.

Für den Effekt 1 (überblenden) gibt der Parameter 'groesse' die Kantenlänge in Pixeln für einen kopierten Block an. Für alle anderen Effekte gibt dieser Parameter an, wieviele Linien zwischen zwei Pausen kopiert werden sollen.

Der Parameter 'pause' gibt die Verzögerung zwischen zwei Kopiervorgängen an. Je größer dieser Wert ist, desto langsamer wird eine Grafik aufgeblendet.

/x:xx /y:xx

Standardmäßig wird eine Seite beim Laden in ein Fenster an den Koordinaten ausgegeben, die im Eigenschaftendialog des Seitenobjekts eingegeben wurden (meistens 0/0). Falls die Anforderung besteht, ein und dieselbe Seite an verschiedene Koordinaten zu laden, dann können die Standardkoordinaten durch die optionalen Parameter '/x' und '/y' verändert werden. Dem Namensteil des Parameters folgt durch den Doppelpunkt getrennt, der Wert für die jeweilige Koordinate.

Beispiel

```
win.SelectPage Auto??    ! Alle Seiten zum Thema Auto auflisten  
win.SelectPage Flugzeug??    ! Alle Seiten zum Thema Flugzeug auflisten
```

```
win.SelectPage Page??? /back    ! Seite im Hintergrund aufbauen  
win.SelectPage Page??? /noclear /paint    ! Fenster nicht löschen, Neuzeichnen  
anstoßen
```

win.SelectPage Page??? /fade:01:02:00 ! Fenster überblenden

Querverweis

[FirstPage](#)

[LastPage](#)

[NextPage](#)

[PprevtPage](#)

[LoadPage](#)

[BackPage](#)

SetBackground

Syntax

```
SetBackground style col1 col2 /transparent /paint
```

Beschreibung

Mit diesem Befehl wird das Füllmuster für den Fensterhintergrund eingestellt. Soll der neue Fensterhintergrund sofort sichtbar werden, muß ein Neuzeichnen des Fensters durch Angabe des optionalen Parameters '/paint' angestoßen werden.

Fixe Parameter

style

Art des Füllmusters. Folgende Werte sind zulässig:

- 0 - null
- 1 - solid
- 2 - horz
- 3 - vert
- 4 - cross
- 5 - diagcross
- 6 - updiag
- 7 - downdiag
- 8 - b25
- 9 - b50
- 10 - b75

Es kann sowohl die Nummer, als auch die dahinterstehende Textkonstante verwendet werden.

col1

Zeichenfarbe des Musters. Es sind die weiter unten angegebenen Farbwerte zulässig.

col2

Füllfarbe des Musters.. Folgende 16 Werte sind zulässig:

- 0 - black
- 1 - blue
- 2 - green
- 3 - cyan
- 4 - red
- 5 - magenta
- 6 - brown
- 7 - gray
- 8 - lightgray
- 9 - lightblue
- 10 - lightgreen
- 11 - lightcyan
- 12 - lightred
- 13 - lightmagenta

14 - yellow
15 - white

Es kann sowohl die Nummer, als auch die dahinterstehende Textkonstante verwendet werden.

Optionale Parameter

/trans
/transparent

Dieser Parameter gibt an, daß die Zwischenräume des Musters nicht ausgefüllt werden sollen. Die Füllfarbe, die durch den Parameter 'col2' definiert wird, ist also nicht zu sehen.

/paint

Dieser Parameter bewirkt ein Neuzeichnen des Fensters, um den neuen Hintergrund sofort zur Anzeige zu bringen.

Beispiel

```
win.SetBackground diagcross lightred yellow /paint
```

SetBorder

Syntax

SetBorder border

Beschreibung

Dieser Befehl ordnet einem Fenster einen 3D-Rahmen zu. Das Fenster muß ein Child-Window vom Typ 0 oder 1 sein. Wird dieser Befehl auf ein Standardfenster (Typ 2) oder ein Dialogfenster (Typ 3, 4) angewendet, wird der Befehlsaufruf ignoriert.

Fixe Parameter

border - Gibt den Rahmentyp an. Folgende Werte sind zulässig:

- 0** - Rechteck
- 1** - abgesenkte Fläche, 1 Pixel tief
- 2** - abgesenkte Fläche, 2 Pixel tief
- 3** - abgesenkte Fläche, 3 Pixel tief
- 4** - angehobene Fläche, 1 Pixel hoch
- 5** - angehobene Fläche, 2 Pixel hoch
- 6** - angehobene Fläche, 3 Pixel hoch
- 7** - abgesenkte Fuge, 1 Pixel tief
- 8** - abgesenkte Fuge, 2 Pixel tief
- 9** - abgesenkte Fuge, 3 Pixel tief
- 10** - angehobene Fuge, 1 Pixel hoch
- 11** - angehobene Fuge, 2 Pixel hoch
- 12** - angehobene Fuge, 3 Pixel hoch
- 13** - abgesenkte Rasterung, 2 Einheiten
- 14** - abgesenkte Rasterung, 3 Einheiten
- 15** - abgesenkte Rasterung, 4 Einheiten
- 16** - angehobene Rasterung, 2 Einheiten
- 17** - angehobene Rasterung, 3 Einheiten
- 18** - angehobene Rasterung, 4 Einheiten
- 19** - Rechteck mit Schatten, 1 Pixel
- 20** - Rechteck mit Schatten, 2 Pixel
- 21** - Rechteck mit Schatten, 3 Pixel
- 22** - Rechteck mit Schatten, 4 Pixel
- 23** - Rechteck mit Schatten, 5 Pixel
- 24** - abgesenkte Fläche mit eingebettetem Rechteck

Optionale Parameter

keine

Beispiel

```
SubWindow.SetBorder 20 ! 3D-Rahmen setzen
```

SetBrush

Syntax

```
SetBrush style col1 col2 /transparent
```

Beschreibung

Mit diesem Befehl wird das Muster für das Füllen von grafischen Elementen wie Kreisen, Rechtecken usw. eingestellt.

Fixe Parameter

style

Art des Füllmusters. Folgende Werte sind zulässig:

- 0 - null
- 1 - solid
- 2 - horz
- 3 - vert
- 4 - cross
- 5 - diagcross
- 6 - updiag
- 7 - downdiag
- 8 - b25
- 9 - b50
- 10 - b75

Es kann sowohl die Nummer, als auch die dahinterstehende Textkonstante verwendet werden.

col1

Zeichenfarbe des Musters. Es sind die weiter unten angegebenen Farbwerte zulässig.

col2

Füllfarbe des Musters.. Folgende 16 Werte sind zulässig:

- 0 - black
- 1 - blue
- 2 - green
- 3 - cyan
- 4 - red
- 5 - magenta
- 6 - brown
- 7 - gray
- 8 - lightgray
- 9 - lightblue
- 10 - lightgreen
- 11 - lightcyan
- 12 - lightred
- 13 - lightmagenta
- 14 - yellow

15 - white

Es kann sowohl die Nummer, als auch die dahinterstehende Textkonstante verwendet werden.

Optionale Parameter

/transparent

Dieser Parameter gibt an, daß die Zwischenräume des Musters nicht ausgefüllt werden sollen. Die Füllfarbe, die durch den Parameter 'col2' definiert wird, ist also nicht zu sehen.

Beispiel

```
win.SetPen solid 0 black  
win.SetBrush diagcross lightred yellow  
win.Paint
```

Querverweis

[SetPen](#)

SetFont

Syntax

```
win.SetFont type size col1 col2 /w0 /w1 /w2 /i /c /u /k /o /s /t
```

Beschreibung

Dieser Befehl definiert die Schriftart, die bei der nächsten Textausgabe durch den Befehl DrawText innerhalb des Fensters verwendet werden soll.

Fixe Parameter

type

Typ der Schriftart. Die Art und Reihenfolge der Schriftarten, die das Autorensystem verwenden soll, wird vom Anwender in der Dialogbox 'Schriften zuordnen' eingestellt. Diese Dialogbox ist z.B. während der Bearbeitung eines Text-Elements zugänglich.

Dieser Parameter gibt an, der wievielte Eintrag in dieser Liste verwendet werden soll. Steht z.B. die Schriftart 'Times New Roman' an der sechsten Stelle der Liste, dann muß für diesen Parameter der Wert 5 angegeben werden (die Numerierung beginnt mit der 0).

size

Höhe der Schrift. Beachten Sie bitte, daß nicht alle Schriften frei skalierbar sind. Für manche Schriften sind hier nur bestimmte Werte für die Größe erlaubt.

col1

Zeichenfarbe der Schrift. Es sind die weiter unten angegebenen Farbwerte zulässig.

col2

Füllfarbe der Schrift. Folgende 16 Werte sind zulässig:

- 0 - black
- 1 - blue
- 2 - green
- 3 - cyan
- 4 - red
- 5 - magenta
- 6 - brown
- 7 - gray
- 8 - lightgray
- 9 - lightblue
- 10 - lightgreen
- 11 - lightcyan
- 12 - lightred
- 13 - lightmagenta
- 14 - yellow
- 15 - white

Es kann sowohl die Nummer, als auch die dahinterstehende Textkonstante verwendet werden.

Optionale Parameter

/w0

Strichstärke der Schrift ist 'dünn' (wird unter Windows und OS/2 nicht unterstützt).

/w1

Strichstärke der Schrift ist 'normal'.

/w2

Strichstärke der Schrift ist 'fett'.

/i

kursive Darstellung (italic).

/c

kursive Darstellung (italic).

/u

unterstrichene Darstellung (underline).

/k

durchgestrichene Darstellung (strikeout).

/o

umrandete Darstellung (outline).

/s

schattierte Darstellung (shadow).

/t

transparente Darstellung (transparent).

Beispiel

```
win.SetFont 5 36 lightred white /w2 /s /t  
win.DrawText "RSE-Author ist ein flexibles Autorensystem."
```

Querverweis

[DrawText](#)

SetOrigin

Syntax

SetOrigin x y

Beschreibung

Dieser Befehl setzt die Ursprungskordinaten innerhalb eines Fensters. Dadurch kann der Inhalt eines Fensters versetzt werden. Der Aufruf von 'SetOrigin 100 0' versetzt z.B. alle Fensterinhalte um 100 Pixel nach links. Dieser Befehl ist vor allem sinnvoll, wenn durch Betätigung der Rollbalken oder durch Aufruf des Befehls Scroll Fensterinhalte verschoben wurden und die Ursprungskordinaten ('SetOrigin 0 0') wiederhergestellt werden sollen.

Fixe Parameter

x - X-Koordinate des Fensterursprungs

y - Y-Koordinate des Fensterursprungs

Optionale Parameter

keine

Beispiel

```
win.Scroll 8 0 /xstep:20 ! Um 160 Pixel nach links verschieben  
Wait  
win.SetOrigin 0 0 ! Ursprungskordinaten wiederherstellen
```

Querverweis

[Scroll](#)

SetOutputSize

Syntax

```
SetOutputSize width height
```

Beschreibung

Dieser Befehl verändert die Größe des Ausgabebereiches eines Fensters. Die Größe von Rahmen und Titelleiste werden automatisch angepaßt. Dieser Befehl wird benötigt, wenn sichergestellt werden soll, daß unabhängig von der Größe des Rahmens und der Titelleiste immer ein definierter Ausgabebereich vorhanden ist.

Fixe Parameter

width - Breite des Ausgabebereichs

height - Höhe des Ausgabebereichs

Optionale Parameter

keine

Beispiel

```
win.SetOutputSize 320 240
```

Querverweis

[SetPosition](#)
[SetSize](#)

SetPen

Syntax

SetPen style width color

Beschreibung

Mit diesem Befehl wird der Linienstil für die Umrandung von grafischen Elementen wie Kreisen, Rechtecken usw. eingestellt.

Fixe Parameter

style

Art des Linienstils. Folgende Werte sind zulässig:

- 0 - null
- 1 - solid
- 2 - dot
- 3 - dash
- 4 - dashdot

Es kann sowohl die Nummer, als auch die dahinterstehende Textkonstante verwendet werden.

width

Dicke der Linie in Pixeln. Wird hier eine 0 angegeben, dann wird automatisch die kleinste Liniendicke (1 Pixel) verwendet. Unter Windows wird bei Dicken größer 1 Pixel nur der Linienstil 'solid' unterstützt.

color

Zeichenfarbe der Linie. Folgende 16 Werte sind zulässig:

- 0 - black
- 1 - blue
- 2 - green
- 3 - cyan
- 4 - red
- 5 - magenta
- 6 - brown
- 7 - gray
- 8 - lightgray
- 9 - lightblue
- 10 - lightgreen
- 11 - lightcyan
- 12 - lightred
- 13 - lightmagenta
- 14 - yellow
- 15 - white

Es kann sowohl die Nummer, als auch die dahinterstehende Textkonstante verwendet werden.

Optionale Parameter

keine

Beispiel

```
win.SetPen solid 0 black  
win.SetBrush diagcross lightred yellow  
win.Paint
```

Querverweis

[SetBrush](#)

SetPosition

Syntax

```
SetPosition x y
```

Beschreibung

Mit diesem Befehl wird die Position des Fensters verändert. Bei Standardfenstern (Typ 2) und Dialogfenster (Typ 3 und 4) beziehen sich die Koordinaten auf die linke obere Ecke des Bildschirms. Bei Child-Windows (Typ 0 und 1) beziehen sich die Koordinaten auf die linke obere Ecke der Ausgabefläche des Vaterfensters.

Fixe Parameter

x - X-Koordinate der neuen Position.
y - Y-Koordinate der neuen Position.

Optionale Parameter

keine

Beispiel

```
win.SetPosition 50 50
```

Querverweis

[SetSize](#)
[SetOutputSize](#)

SetRasterOp

Syntax

```
SetRasterOp rop
```

Beschreibung

Dieser Befehl definiert den Verknüpfungsmodus, mit dem grafische Elemente beim Zeichnen mit dem Hintergrund verknüpft werden.

Fixe Parameter

rop

Art des Verknüpfungsmodus. Folgende Werte sind zulässig:

- 0 - overpaint
- 1 - invert
- 2 - and
- 3 - or
- 4 - xor
- 5 - nxor

Es kann sowohl die Nummer, als auch die dahinterstehende Textkonstante verwendet werden.

Optionale Parameter

keine

Beispiel

```
win.SetRasterOp invert
```

SetScript

Syntax

SetScript {.....} /open /close /move /resize /click /dclick

Beschreibung

Dieser Befehl erlaubt es, ein Kommando-Script für das Fenster neu zu definieren.

Fixe Parameter

{.....}

Das neue Kommando-Script wird in geschweifte Klammern eingeschlossen. Die Definition des neuen Scripts kann sich auch über mehrere Zeilen erstrecken. Wird kein optionaler Parameter angegeben, dann wird das Open-Script des Fensters neu besetzt.

Optionale Parameter

/open

Das Open-Script wird neu definiert.

/close

Das Close-Script wird neu definiert.

/move

Das Move-Script wird neu definiert.

/resize

Das Resize-Script wird neu definiert.

/click

Das Click-Script wird neu definiert.

/dclick

Das Dclick-Script wird neu definiert.

Beispiel

win.SetScript {Sound warning} /click ! Click-Script wird neu definiert

SetScrollParameter

Syntax

```
SetScrollParameter /release /immediate /key
```

Beschreibung

Dieser Befehl definiert das Verhalten der Rollbalken in einem Fenster.

Fixe Parameter

keine

Optionale Parameter

/release

Wird einer der Pfeile in einem Rollbalken angeklickt, erfolgt der Scrollvorgang erst, wenn die Maustaste losgelassen wird (Standardeinstellung).

/immediate

Wird einer der Pfeile in einem Rollbalken angeklickt, wird der Fensterinhalt während des Festhaltens der Maus zeilenweise weitergescrollt.

/key

Wenn das angegebene Fenster über Rollbalken verfügt, kann das Fenster auch über die Pfeiltasten gescrollt werden. Da immer nur ein Fenster über die Pfeiltasten gescrollt werden kann, werden vorherige Zuordnungen dieser Art ungültig.

Beispiel

```
win.SetScrollParameter /immediate /key
```

SetSize

Syntax

```
SetSize width height
```

Beschreibung

Dieser Befehl verändert die Größe des Fensters. Die Angaben für Breite und Höhe beziehen sich auf die gesamte Fensterfläche incl. Titelleiste und Rahmen.

Fixe Parameter

width - Breite des Fensters

height - Höhe des Fensters

Optionale Parameter

keine

Beispiel

```
win.SetSize 320 240
```

Querverweis

[SetPosition](#)
[SetOutputSize](#)

SetTitle

Syntax

```
SetTitle "....."
```

Beschreibung

Dieser Befehl stellt den Text für die Titelleiste eines Standardfensters (Typ 2) oder Dialogfensters (Typ 3 und 4) ein. Ist das Fenster ein Child-Window (Typ 0 und 1) wird der Aufruf dieses Befehls ignoriert.

Fixe Parameter

"....." - Neuer Text für die Titelzeile

Optionale Parameter

keine

Beispiel

```
win.SetTitle "Das ist der Titel des Fensters"
```

SetUpdateMode

Syntax

```
SetUpdateMode par
```

Beschreibung

Mit diesem Befehl ist es möglich, das automatische Update eines Fensters ein- und auszuschalten. Wird das automatische Update abgeschaltet, wird das Neuzeichnen des Fensters bei Veränderungen der Größe, oder wenn ein Fenster vor diesem Fenster geschlossen wird, nicht mehr durchgeführt.

Fixe Parameter

par

Dieser Parameter gibt an, ob das automatische Update eingeschaltet ist oder nicht. Folgende Werte können hier angegeben werden:

```
0 / off / false  
1 / on / true
```

Optionale Parameter

keine

Beispiel

```
win.SetUpdateMode 0  
win.SetUpdateMode 1  
win.SetUpdateMode off  
win.SetUpdateMode on  
win.SetUpdateMode false  
win.SetUpdateMode true
```

Show

Syntax

Show

Beschreibung

Ein unsichtbares Fenster (Einstellung im Eigenschaften-Dialog oder Aufruf des Befehls Hide) kann mit diesem Befehl wieder sichtbar geschaltet werden. Alle Elemente des Fensters werden ebenfalls wieder sichtbar.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

win.Show

Querverweis

Hide

StartTimer

Syntax

```
StartTimer msec {.....} /reset /restart
```

Beschreibung

Dieser Befehl startet einen sogenannten Timer. Dieser Timer führt nach der angegebenen Anzahl von Millisekunden ein Script aus, das dem Befehl in geschweifte Klammern eingeschlossen, übergeben wird.

Das ist z.B. sinnvoll, wenn eine selbstablaufende Show programmiert werden soll, oder wenn nach Ablauf einer bestimmten Zeit automatisch zum Hauptindex zurückgesprungen werden soll.

Jedes Bildschirmfenster kann einen eigenen Timer enthalten, so daß auch mehrere Timer gleichzeitig aktiv sein können. Ein Timer sollte grundsätzlich auf das eigene Fenster wirken. Das hat den Vorteil, daß beim Schließen eines Fensters der Timer automatisch deaktiviert wird. Durch Aufruf des Befehl StopTimer kann ein Timer explizit angehalten werden.

Fixe Parameter

msec

Anzahl der Millisekunden, nach deren Ablauf das Timer-Script ausgeführt werden soll.

{.....}

Zwischen den eckigen Klammern steht das Script, das nach Ablauf des Timers ausgeführt werden soll.

Optionale Parameter

/reset

Beim Auftreten einer Benutzeraktion (Taste oder Maustaste gedrückt), soll der Timer auf den Startwert zurückgesetzt werden.

/restart

Nach Auslösung eines Timeouts (Ausführung des Timer-Scripts), soll der Timer neu gestartet werden.

Beispiel

! Nach 1 Minute ohne Eingabe den Hauptindex wieder laden.
win.StartTimer 60000 {win.LoadPage MainIndex} /reset

Querverweis

StopTimer

StopTimer

Syntax

StopTimer

Beschreibung

Dieser Befehl deaktiviert einen laufenden Timer, der zuvor mit dem Befehl StartTimer aktiviert wurde.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

win.StopTimer

Querverweis

StartTimer

StopVideo

Syntax

StopVideo

Beschreibung

Dieser Befehl stoppt das Abspielen eines Videos, das in dem angegebenen Fenster mit dem Befehl [PlayVideo](#) gestartet wurde.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

win.StopVideo

QuerVerweis

[PlayVideo](#)

ToTop

Syntax

ToTop

Beschreibung

Der Aufruf dieses Befehls bewirkt, daß das Fenster auf dem Bildschirm als oberstes Fenster dargestellt wird.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

win.ToTop

Unmark

Syntax

Unmark [/field /hypertext]

Beschreibung

Dieser Befehl hebt die Markierung aller Hyperlink- und/oder Hypertextfelder im angegebenen Fenster auf.

Standardmäßig wird die Markierung aller Hyperlink- und Hypertextfelder aufgehoben. Durch die optionalen Parameter '/field' und '/hypertext' kann genauer definiert werden, auf welchen Feldtyp der Befehl wirken soll.

Fixe Parameter

keine

Optionale Parameter

/field

Die Markierung der Hyperlinkfelder im Fenster wird aufgehoben.

/hypertext

Die Markierung der Hypertextfelder im Fenster wird aufgehoben.

Beispiel

```
win.Unmark  
win.Unmark /field  
win.Unmark /hypertext
```

Querverweis

Mark

Alias

Syntax

Alias name

Beschreibung

Dieser Befehl erlaubt es, zusätzlich zu dem Namen, der einem Schalter bei der Bearbeitung im Eigenschaften-Dialog gegeben werden kann, noch einen Kurznamen zu vergeben.

Fixe Parameter

name - Kurzname, unter dem der Schalter angesprochen werden kann.

Optionale Parameter

keine

Beispiel

```
Button01.Alias bt01  
bt01.Disable
```

Disable

Syntax

Disable

Beschreibung

Nach Aufruf dieses Befehl ist der Schalter nicht mehr aktivierbar.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

but.Disable ! Schalter ist nicht mehr aktivierbar

Querverweis

[Enable](#)

Enable

Syntax

Enable

Beschreibung

Nach Aufruf dieses Befehl ist ein Schalter aktivierbar.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

but.Enable ! Schalter ist aktivierbar

Querverweis

[Disable](#)

GrabFocus

Syntax

GrabFocus

Beschreibung

Durch Aufruf dieses Befehls erhält der Schalter den Fokus, d.h. der Schalter wird optisch hervorgehoben und kann durch Drücken der Eingabetaste aktiviert werden.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

but.GrabFocus

Hide

Syntax

Hide

Beschreibung

Der Schalter ist nach Aufruf dieses Befehls nicht mehr sichtbar, er existiert aber sowohl auf der Ebene des Laufzeitsystems, als auch auf der Ebene des zugrundeliegenden Fenstersystems weiter.

Bei Aufruf des Befehls Show wird der Schalter wieder angezeigt.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

but.Hide

Querverweis

Show

Paint

Syntax

Paint

Beschreibung

Dieser Befehl bewirkt das Neuzeichnen eines Schalters.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

but.Paint

SetPosition

Syntax

```
SetPosition x y
```

Beschreibung

Mit diesem Befehl wird die Position des Schalters im Fenster verändert. Die Koordinaten beziehen sich auf die linke obere Ecke der Ausgabefläche des Vaterfensters.

Fixe Parameter

- x** - X-Koordinate der neuen Position.
- y** - Y-Koordinate der neuen Position.

Optionale Parameter

keine

Beispiel

```
but.SetPosition 50 50
```

Querverweis

[SetSize](#)

SetScript

Syntax

```
SetScript {.....}
```

Beschreibung

Dieser Befehl erlaubt es, ein Kommando-Script für den Schalter neu zu definieren.

Fixe Parameter

```
{.....}
```

Das neue Kommando-Script wird in geschweifte Klammern eingeschlossen. Die Definition des neuen Scripts kann sich auch über mehrere Zeilen erstrecken.

Optionale Parameter

keine

Beispiel

```
but.SetScript {Sound warning} ! Script wird neu definiert
```


SetSize

Syntax

```
SetSize width height
```

Beschreibung

Dieser Befehl verändert die Größe des Schalters. Dieser Befehl wirkt nicht auf Bitmap-Schalter, da deren Größe durch die enthaltene Bitmap definiert wird.

Fixe Parameter

width - Breite des Schalters

height - Höhe des Schalters

Optionale Parameter

keine

Beispiel

```
but.SetSize 100 30
```

Querverweis

[SetPosition](#)

Show

Syntax

Show

Beschreibung

Ein unsichtbarer Schalter (Aufruf des Befehls Hide) kann mit diesem Befehl wieder sichtbar geschaltet werden.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

but.Show

Querverweis

Hide

Alias

Syntax

Alias name

Beschreibung

Dieser Befehl erlaubt es, zusätzlich zu dem Namen, der einem Hyperlinkfeld bei der Bearbeitung im Eigenschaften-Dialog gegeben werden kann, noch einen Kurznamen zu vergeben.

Fixe Parameter

name - Kurzname, unter dem das Feld angesprochen werden kann.

Optionale Parameter

keine

Beispiel

Feld01.Alias fd01
fd01.Disable

Disable

Syntax

Disable

Beschreibung

Nach Aufruf dieses Befehl ist das Hyperlinkfeld nicht mehr aktivierbar.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

fld.Disable ! Schalter ist nicht mehr aktivierbar

Querverweis

[Enable](#)

Enable

Syntax

Enable

Beschreibung

Nach Aufruf dieses Befehl ist ein Hyperlinkfeld aktivierbar.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

fld.Enable ! Schalter ist aktivierbar

Querverweis

[Disable](#)

Mark

Syntax

Mark /style:xx:xx:xx /invert

Beschreibung

Dieser Befehl markiert ein Hyperlinkfeld.

Wird die Art der Markierung nicht über die optionalen Parameter '/style' oder '/invert' angegeben, dann wird das Feld durch ein Rechteck mit der über den Befehl SetMarkStyle eingestellten Linienart markiert.

Fixe Parameter

keine

Optionale Parameter

/style:xx:xx:xx

Als Markierung wird ein Rechteck verwendet. Durch Doppelpunkt getrennt, folgen die Werte für die Linienart, die Liniendicke und die Farbe der Linie (äquivalent zu den Einstellungen durch den Befehl SetPen). Beachten Sie bitte, daß die Markierung im XOR-Modus in den Hintergrund eingezeichnet wird. Die eingestellte Farbe erscheint deshalb nur auf schwarzem Hintergrund. Auf andersfarbigem Hintergrund kann es zu Abweichungen kommen.

/invert

Das Feld wird durch Invertierung markiert.

Beispiel

fld.Mark
fld.Mark /style:solid:0:green
fld.Mark /invert

Querverweis

SetMarkStyle
Unmark

SetPosition

Syntax

```
SetPosition x y
```

Beschreibung

Mit diesem Befehl wird die Position des Hyperlinkfeldes im Fenster verändert. Die Koordinaten beziehen sich auf die linke obere Ecke der Ausgabefläche des Vaterfensters.

Fixe Parameter

- x** - X-Koordinate der neuen Position.
- y** - Y-Koordinate der neuen Position.

Optionale Parameter

keine

Beispiel

```
fld.SetPosition 50 50
```

Querverweis

[SetSize](#)

SetScript

Syntax

```
SetScript {.....} /click /dclick /movein /moveout
```

Beschreibung

Dieser Befehl erlaubt es, ein Kommando-Script für das Hyperlinkfeld neu zu definieren.

Fixe Parameter

{.....}

Das neue Kommando-Script wird in geschweifte Klammern eingeschlossen. Die Definition des neuen Scripts kann sich auch über mehrere Zeilen erstrecken. Wird kein optionaler Parameter angegeben, dann wird das Click-Script des Fensters neu besetzt.

Optionale Parameter

/click

Das Click-Script wird neu definiert.

/dclick

Das Dclick-Script wird neu definiert.

/movein

Das MoveIn-Script wird neu definiert.

/moveout

Das Moveout-Script wird neu definiert.

Beispiel

```
fld.SetScript {Sound warning} /click ! Click-Script wird neu definiert
```


SetSize

Syntax

```
SetSize width height
```

Beschreibung

Dieser Befehl verändert die Größe des Hyperlinkfeldes.

Fixe Parameter

width - Breite des Hyperlinkfeldes

height - Höhe des Hyperlinkfeldes

Optionale Parameter

keine

Beispiel

```
fld.SetSize 100 30
```

Querverweis

[SetPosition](#)

Unmark

Syntax

Unmark

Beschreibung

Dieser Befehl hebt die Markierung eines Hyperlinkfeldes wieder auf.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

fld.Unmark

Querverweis

Mark

Alias

Syntax

Alias name

Beschreibung

Dieser Befehl erlaubt es, zusätzlich zu dem Namen, der einem Hypertextfeld bei der Bearbeitung im Eigenschaften-Dialog gegeben werden kann, noch einen Kurznamen zu vergeben.

Fixe Parameter

name - Kurzname, unter dem das Feld angesprochen werden kann.

Optionale Parameter

keine

Beispiel

```
Hyper01.Alias ht01  
ht01.Disable
```

Disable

Syntax

Disable

Beschreibung

Nach Aufruf dieses Befehl ist das Hypertextfeld nicht mehr aktivierbar.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

hyp.Disable ! Schalter ist nicht mehr aktivierbar

Querverweis

[Enable](#)

Enable

Syntax

Enable

Beschreibung

Nach Aufruf dieses Befehl ist ein Hypertextfeld aktivierbar.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

hyp.Enable ! Schalter ist aktivierbar

Querverweis

[Disable](#)

Mark

Syntax

Mark /style:xx:xx:xx /invert

Beschreibung

Dieser Befehl markiert ein Hypertextfeld.

Wird die Art der Markierung nicht über die optionalen Parameter '/style' oder '/invert' angegeben, dann wird das Feld durch ein Rechteck mit der über den Befehl SetMarkStyle eingestellten Linienart markiert.

Fixe Parameter

keine

Optionale Parameter

/style:xx:xx:xx

Als Markierung wird ein Rechteck verwendet. Durch Doppelpunkt getrennt, folgen die Werte für die Linienart, die Liniendicke und die Farbe der Linie (äquivalent zu den Einstellungen durch den Befehl SetPen). Beachten Sie bitte, daß die Markierung im XOR-Modus in den Hintergrund eingezeichnet wird. Die eingestellte Farbe erscheint deshalb nur auf schwarzem Hintergrund. Auf andersfarbigem Hintergrund kann es zu Abweichungen kommen.

/invert

Das Feld wird durch Invertierung markiert.

Beispiel

```
hyp.Mark  
hyp.Mark /style:solid:0:green  
hyp.Mark /invert
```

Querverweis

SetMarkStyle
Unmark

SetScript

Syntax

```
SetScript {.....} /click /dclick /movein /moveout
```

Beschreibung

Dieser Befehl erlaubt es, ein Kommando-Script für das Hypertextfeld neu zu definieren.

Fixe Parameter

{.....}

Das neue Kommando-Script wird in geschweifte Klammern eingeschlossen. Die Definition des neuen Scripts kann sich auch über mehrere Zeilen erstrecken. Wird kein optionaler Parameter angegeben, dann wird das Click-Script des Fensters neu besetzt.

Optionale Parameter

/click

Das Click-Script wird neu definiert.

/dclick

Das Dclick-Script wird neu definiert.

/movein

Das MoveIn-Script wird neu definiert.

/moveout

Das Moveout-Script wird neu definiert.

Beispiel

```
hyp.SetScript {Sound warning} /click ! Click-Script wird neu definiert
```

Unmark

Syntax

Unmark

Beschreibung

Dieser Befehl hebt die Markierung eines Hypertextfeldes wieder auf.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

hyp.Unmark

Querverweis

Mark

Alias

Syntax

Alias name

Beschreibung

Dieser Befehl erlaubt es, zusätzlich zu dem Namen, der einer ListBox bei der Bearbeitung im Eigenschaften-Dialog gegeben werden kann, noch einen Kurznamen zu vergeben.

Fixe Parameter

name - Kurzname, unter dem die ListBox angesprochen werden kann.

Optionale Parameter

keine

Beispiel

```
ListBox01.Alias lb01  
lb01.Disable
```

Disable

Syntax

Disable

Beschreibung

Nach Aufruf dieses Befehl ist die ListBox nicht mehr aktivierbar.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

Ist.Disable ! ListBox ist nicht mehr aktivierbar

Querverweis

[Enable](#)

Enable

Syntax

Enable

Beschreibung

Nach Aufruf dieses Befehl ist eine ListBox aktivierbar.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

Ist.Enable ! ListBox ist aktivierbar

Querverweis

[Disable](#)

GrabFocus

Syntax

GrabFocus

Beschreibung

Durch Aufruf dieses Befehls erhält die ListBox den Fokus, d.h. die ListBox wird optisch hervorgehoben. Des weiteren kann der Auswahlbalken der ListBox mit den Pfeiltasten verschoben werden.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

Ist.GrabFocus

Hide

Syntax

Hide

Beschreibung

Die ListBox ist nach Aufruf dieses Befehls nicht mehr sichtbar, sie existiert aber sowohl auf der Ebene des Laufzeitsystems, als auch auf der Ebene des zugrundeliegenden Fenstersystems weiter.

Bei Aufruf des Befehls Show wird die ListBox wieder angezeigt.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

Ist.Hide

Querverweis

Show

Paint

Syntax

Paint

Beschreibung

Dieser Befehl bewirkt das Neuzeichnen einer ListBox.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

Ist.Paint

SelectPos

Syntax

SelectPos pos

Beschreibung

Dieser Befehl selektiert einen Eintrag in einer ListBox und hebt diesen Eintrag optisch hervor.

Fixe Parameter

pos - Nummer des Eintrags, der selektiert werden soll.

Optionale Parameter

keine

Beispiel

Ist.SelectPos 10

Querverweis

SelectString

SelectString

Syntax

```
SelectString "....."
```

Beschreibung

Dieser Befehl selektiert einen Eintrag in einer ListBox. Als Parameter wird der Text des Eintrags übergeben, der selektiert werden soll.

Fixe Parameter

"....." - Text des Eintrags, der selektiert werden soll.

Optionale Parameter

keine

Beispiel

```
Ist.SelectString "Eintrag05"
```

Querverweis

[SelectPos](#)

SetEntryScript

Syntax

```
SetEntryScript {.....} /pos:xx /select /dclick
```

Beschreibung

Dieser Befehl erlaubt es, ein Kommando-Script für einen Eintrag der ListBox neu zu definieren.

Fixe Parameter

{.....}

Das neue Kommando-Script wird in geschweifte Klammern eingeschlossen. Die Definition des neuen Scripts kann sich auch über mehrere Zeilen erstrecken. Wird kein optionaler Parameter angegeben, dann wird das Select-Script des Fensters neu besetzt.

Optionale Parameter

/pos:xx

Mit diesem optionalen Parameter wird angegeben, für welchen Eintrag der ListBox ein neues Kommandoscript definiert werden soll. Wird dieser Parameter nicht angegeben, dann wird der aktuelle Eintrag mit einem neuen Script belegt.

/select

Das Select-Script wird neu definiert.

/dclick

Das Dclick-Script wird neu definiert.

Beispiel

```
Ist.SetEntryScript {Sound warning} /pos:10 /select ! Script wird neu definiert
```

SetPosition

Syntax

```
SetPosition x y
```

Beschreibung

Mit diesem Befehl wird die Position der ListBox im Fenster verändert. Die Koordinaten beziehen sich auf die linke obere Ecke der Ausgabefläche des Vaterfensters.

Fixe Parameter

- x** - X-Koordinate der neuen Position.
- y** - Y-Koordinate der neuen Position.

Optionale Parameter

keine

Beispiel

```
Ist.SetPosition 50 50
```

Querverweis

[SetSize](#)

SetSize

Syntax

SetSize width height

Beschreibung

Dieser Befehl verändert die Größe der ListBox.

Fixe Parameter

width - Breite der ListBox

height - Höhe der ListBox

Optionale Parameter

keine

Beispiel

```
Ist.SetSize 100 30
```

Querverweis

[SetPosition](#)

Show

Syntax

Show

Beschreibung

Eine unsichtbare ListBox (Aufruf des Befehls Hide) kann mit diesem Befehl wieder sichtbar geschaltet werden.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

Ist.Show

Querverweis

Hide

Alias

Syntax

Alias name

Beschreibung

Dieser Befehl erlaubt es, zusätzlich zu dem Namen, der einem Drucker bei der Eröffnung mit dem Befehl OpenPrinter gegeben wurde, noch einen Kurznamen zu vergeben.

Fixe Parameter

name - Kurzname, unter dem der Drucker angesprochen werden kann.

Optionale Parameter

keine

Beispiel

```
OpenPrinter Printer /option  
Printer.Alias prn  
prn.LoadDocument Doku001
```

Close

Syntax

Close

Beschreibung

Schließt den Drucker wieder und beendet den aktuellen Druckauftrag.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

prn.Close

LoadDocument

Syntax

LoadDocument object

Beschreibung

Dieser Befehl lädt ein Druckobjekt, das in der Entwicklungsumgebung als Objekt vom Typ 'Dokument' angelegt wurde, in den Druckpuffer. Um dieses Dokument auf dem Drucker auszugeben, muß anschließend der Befehl PrintPage aufgerufen werden.

Fixe Parameter

object - Name des Druckobjekts.

Optionale Parameter

keine

Beispiel

```
prn.LoadDocument Doku001  
prn.PrintPage
```

PrintPage

Syntax

PrintPage

Beschreibung

Dieser Befehl bewirkt die Ausgabe einer Druckseite, die zuvor mit dem Befehl LoadDocument geladen wurde, auf dem Drucker.

Fixe Parameter

keine

Optionale Parameter

keine

Beispiel

```
prn.LoadDocument Doku001  
prn.PrintPage
```


Metafile

In einem sogenannten Metafile werden alle Text- und Grafikausgaben aufgezeichnet, die in ein Fenster gemacht werden. Wird ein Neuzeichnen eines Fensters nötig, weil z.B. ein davorliegendes Fenster geschlossen wurde, dann werden die im Metafile aufgezeichneten Ausgaben einfach neu in das Fenster gezeichnet.

Window-Metafile / Page-Metafile

RSE-Author zeichnet alle Ausgaben in zwei verschiedenen Metafiles auf: dem Window- und dem Page-Metafile.

Im Window-Metafile werden alle Ausgaben aufgezeichnet, die als Elemente bei der Bearbeitung des Fensterobjekts direkt in das Fenster eingetragen wurden. Diese Ausgaben stellen den Hintergrund für das Fensterobjekt dar.

Alle folgenden Ausgaben (Laden einer Seite, Draw....-Befehl) werden im Page-Metafile aufgezeichnet. Diese Ausgaben stellen den Vordergrund des Fensterobjekts dar.

Beim Laden einer neuen Seite wird nur das Page-Metafile gelöscht. Danach wird ein Neuzeichnen des Fensters angestoßen. Beim Neuzeichnen des Fensters werden die Ausgaben im Window-Metafile wiederhergestellt, der Hintergrund wird also restauriert. Die Ausgaben der vorigen Seite sind aber verschwunden. Vor dem restaurierten Hintergrund wird dann die neue Seite gezeichnet.

Datei

In diesem Menü können Sie eine Programm-Datei anlegen, zur Bearbeitung eröffnen, abspeichern oder ausdrucken. Des weiteren steht ein Menüpunkt zur Verfügung, um den Editor zu beenden.

Das Menü enthält die folgenden Punkte:

Neu

Öffnen

Speichern

Speichern unter

Drucken

Druckerinstallation

Beenden

Datei/Neu

Mit diesem Menüpunkt wird eine Programm-Datei namens 'namenlos.rsc' neu angelegt. Beim Abspeichern der Datei wird automatisch abgefragt, unter welchem Namen die Datei gespeichert werden soll.

Datei/Öffnen

Mit diesem Menüpunkt wird eine Programm-Datei zur Bearbeitung eröffnet. Zur Abfrage des Dateinamens wird eine Dateiauswahlbox aufgeblendet. Hier wird der Name der Datei eingegeben. Der Dateiname muß die Kennung .PRG haben. Nach der Auswahl der Programm-Datei werden die in der Datei enthaltenen Objekte im Listenfenster angezeigt und können dort zur Bearbeitung ausgewählt werden.

Datei/Speichern

Nach Aufruf dieses Menüpunkts wird die Programm-Datei abgespeichert. Falls es sich um eine neu eröffnete Datei namens 'namenlos.prg' handelt, wird zuvor noch der Dateiname abgefragt, unter dem die Programm-Datei gespeichert werden soll.

Datei/Speichern unter

Nach Aufruf dieses Menüpunkts wird die gerade bearbeitete Programm-Datei unter einem anderen Namen abgespeichert. Der neue Dateiname wird zuvor in einer Dateiauswahlbox abgefragt.

Datei/Drucken

Mit dieser Funktion wird das aktuell bearbeitete Objekt auf einen angeschlossenen Drucker ausgegeben. Bei der Ausgabe auf den Drucker werden zuerst allgemeine Informationen über das gedruckte Objekt in der Kopfzeile ausgegeben. Danach wird das Objekt als Bitmap gedruckt. Die Seite wird dann mit Linien aufgefüllt, auf denen Notizen eingetragen werden können.

Handelt es sich bei dem zu druckenden Objekt um ein Dokument, dann wird das Objekt genauso ausgedruckt wie im Testmodus.

Datei/Druckerinstallation

Diese Funktion ruft eine Dialogbox zur Konfiguration des angeschlossenen Druckers auf.

Datei/Beenden

Nach Aufruf dieses Menüpunktes wird der Editor beendet und aus dem Arbeitsspeicher entfernt.

Bearbeiten

In diesem Menü stehen Ihnen Funktionen zum Zugriff auf die Zwischenablage (Clipboard) Ihres Systems zur Verfügung.

Die Zwischenablage dient dazu, Daten zwischen verschiedenen Applikationen oder auch innerhalb einer Applikation zu kopieren oder zu verschieben.

Folgende Funktionen stehen zur Verfügung:

Rückgängig

Ausschneiden

Kopieren

Einfügen

Löschen

Elemente nach vorne

Elemente nach hinten

Bearbeiten/Rückgängig

Mit diesem Befehl wird der letzte Clipboard-Befehl rückgängig gemacht. Das aktuelle Arbeitsfenster wird wieder so hergestellt, wie es vor der letzten Clipboard-Operation war.

Bearbeiten/Ausschneiden

Dieser Befehl entfernt alle selektierten Elemente aus dem aktuell bearbeiteten Objekt und kopiert sie in die Zwischenablage.

Sie können dann mit Bearbeiten/Einfügen die Elemente in irgendein anderes Objekt einfügen (oder auch im selben Objekt an anderer Stelle).

Da die Elemente in der Zwischenablage verbleiben, können sie beliebig oft eingefügt werden.

Bearbeiten/Kopieren

Dieser Befehl kopiert alle selektierten Elemente in die Zwischenablage, ohne sie dabei aus dem Objekt zu entfernen.

Mit dem Befehl Bearbeiten/Einfügen können Sie die Elemente in ein anderes Objekt kopieren.

Da die Elemente in der Zwischenablage verbleiben, können sie beliebig oft eingefügt werden.

Bearbeiten/Einfügen

Der Befehl *Einfügen* fügt eine Liste von Elementen aus der Zwischenablage in das aktuell bearbeitete Objekt ein.

Bearbeiten/Löschen

Der Befehl *Löschen* löscht alle selektierten Elemente im aktuell bearbeiteten Objekt, ohne sie in die Zwischenablage zu kopieren.

Ein Einfügen der Elemente wie bei Bearbeiten/Ausschneiden oder Bearbeiten/Kopieren ist danach nicht mehr möglich.

Bearbeiten/Elemente nach vorne

Nach Aufruf dieser Funktion werden alle selektierten Elemente in den Vordergrund verschoben.

Bearbeiten/Elemente nach hinten

Nach Aufruf dieser Funktion werden alle selektierten Elemente in den Hintergrund verschoben.

Objekt

In diesem Menü stehen Ihnen Funktionen zum Einfügen von neuen Objekten in die aktuell bearbeitete Programm-Datei zur Verfügung.

Folgende Funktionen stehen zur Verfügung:

Seite
Fenster
Dokument
Einlagern

Ein Objekt kann alternativ auch über den Einfügen-Schalter im Listenfenster angelegt werden.

Objekt/Seite

Dieser Befehl legt ein Seitenobjekt an. Ein derartiges Objekt enthält eine Gruppe von Grafikanweisungen, die von einem Programm in ein bereits eröffnetes Fenster geladen werden können. Angenommen, Sie haben im Programm ein Fenster eröffnet, in dem ein Anwender in Texten und Grafiken blättern kann. Für jede Text- oder Grafikseite legen Sie ein Seitenobjekt an. Beim Blättern wird das Ausgabefenster gelöscht und nachfolgend die nächste Seite geladen.

Objekt/Fenster

Dieser Befehl legt ein Fensterobjekt an. Dieses Objekt stellt ein komplettes Fenster dar. Wird in einem Programm ein solches Objekt geladen, wird das Fenster mit allen Inhalten auf dem Bildschirm eröffnet.

Objekt/Dokument

Dieser Befehl legt ein Druckobjekt an. Ein solches Objekt enthält eine komplette Seite für die Ausgabe auf den Drucker. Wenn Ihr Programm Trainingsunterlagen ausgeben soll, dann können Sie diese Unterlagen als Objekte in der Programm-Datei anlegen und innerhalb des Programms aufrufen.

Objekt/Einlagern

Mit diesem Befehl können Objekte aus einer anderen Programm-Datei in die aktuell bearbeitete Datei eingelagert werden.

Zu diesem Zweck wird zuerst eine Dateiauswahlbox aufgeblendet, in welcher der Name der Programm-Datei eingegeben werden muß, aus der Objekte übernommen werden sollen.

Danach wird eine Dialogbox aufgeblendet, die zwei Listen enthält. In der linken Liste sind alle Objekte aufgelistet, die in der Datei enthalten sind, aus der Objekte übernommen werden sollen. In der rechten Liste werden die Objekte eingetragen, die kopiert werden sollen.

Element

In diesem Menü stehen Ihnen Funktionen zum Einfügen von neuen Elementen in das aktuell bearbeitete Programm-Objekt zur Verfügung.

Folgende Funktionen stehen zur Verfügung:

Linie
Rechteck
Kreis
Grafik
Icon
Text
Rahmen
Schalter
Hyperlink-Feld
Listbox
Fenster

Ein Element kann alternativ auch über die Schalterleiste des Barbeitungsfensters angelegt werden.

Element/Linie

Dieser Befehl fügt eine Linie in das aktuell bearbeitete Objekt ein. Um die Linie an eine andere Stelle im Bearbeitungsfenster zu verschieben, klicken Sie die Linie mit der linken Maustaste an und verschieben Sie die Linie bei gedrückter Maustaste. Um die Größe der Linie zu verändern, klicken Sie die Umrandung der Linie mit der linken Maustaste an und verschieben Sie die Umrandung bei weiterhin gedrückter Maustaste. Die Größe des Linienelements paßt sich der Bewegung der Maus an.

Um die Eigenschaften der Linie (Farbe, Stil, Orientierung) zu verändern, führen Sie mit der Maus einen Doppelklick auf die Linie aus. Daraufhin wird ein Dialogfenster aufgeblendet, in dem Sie alle Eigenschaften der Linie einstellen können.

Element/Rechteck

Dieser Befehl fügt ein Rechteck in das aktuell bearbeitete Objekt ein. Um das Rechteck an eine andere Stelle im Bearbeitungsfenster zu verschieben, klicken Sie das Rechteck mit der linken Maustaste an und verschieben Sie das Rechteck bei gedrückter Maustaste. Um die Größe des Rechtecks zu verändern, klicken Sie die Umrandung des Rechtecks mit der linken Maustaste an und verschieben Sie die Umrandung bei weiterhin gedrückter Maustaste. Die Größe des Rechtecks paßt sich der Bewegung der Maus an.

Um die Eigenschaften des Rechtecks (Farbe, Stil, Füllmuster) zu verändern, führen Sie mit der Maus einen Doppelklick auf das Rechteck aus. Daraufhin wird ein Dialogfenster aufgeblendet, in dem Sie alle Eigenschaften des Rechtecks einstellen können.

Element/Kreis

Dieser Befehl fügt eine Ellipse in das aktuell bearbeitete Objekt ein. Um die Ellipse an eine andere Stelle im Bearbeitungsfenster zu verschieben, klicken Sie die Ellipse mit der linken Maustaste an und verschieben Sie die Ellipse bei gedrückter Maustaste. Um die Größe der Ellipse zu verändern, klicken Sie die Umrandung der Ellipse mit der linken Maustaste an und verschieben Sie die Umrandung bei weiterhin gedrückter Maustaste. Die Größe des Ellipsenelements paßt sich der Bewegung der Maus an.

Um die Eigenschaften der Ellipse (Farbe, Stil, Füllmuster) zu verändern, führen Sie mit der Maus einen Doppelklick auf die Ellipse aus. Daraufhin wird ein Dialogfenster aufgeblendet, in dem Sie alle Eigenschaften der Ellipse einstellen können.

Element/Grafik

Dieser Befehl fügt eine Bitmap-Grafik in das aktuell bearbeitete Objekt ein. Beim Anlegen dieses Elements wird vorerst ein Platzhalter eingefügt. Durch einen Doppelklick mit der Maus wird der Bearbeitungsdialog aufgeblendet. Hier kann dann der Dateiname der Bitmap-Grafik angegeben werden.

Um die Grafik an eine andere Stelle im Bearbeitungsfenster zu verschieben, klicken Sie die Grafik mit der linken Maustaste an und verschieben Sie die Grafik bei gedrückter Maustaste.

Ist im Bearbeitungsdialog die Option 'Bitmap skalieren' aktiviert, kann die Grafik in der Größe verändert werden. Um die Größe der Grafik zu verändern, klicken Sie die Umrandung der Grafik mit der linken Maustaste an und verschieben Sie die Umrandung bei weiterhin gedrückter Maustaste. Die Größe des Grafikelements paßt sich der Bewegung der Maus an.

Um die Eigenschaften der Grafik (Dateiname, Palette, skalierbar usw.) zu verändern, führen Sie mit der Maus einen Doppelklick auf die Grafik aus. Daraufhin wird ein Dialogfenster aufgeblendet, in dem Sie alle Eigenschaften der Grafik einstellen können.

Element/Icon

Dieser Befehl fügt ein Standard-Icon in das aktuell bearbeitete Objekt ein. Um das Icon an eine andere Stelle im Bearbeitungsfenster zu verschieben, klicken Sie das Icon mit der linken Maustaste an und verschieben Sie die Ellipse bei gedrückter Maustaste. Icons können nicht in der Größe verändert werden.

Um die Eigenschaften des Icons (Typ) zu verändern, führen Sie mit der Maus einen Doppelklick auf das Icon aus. Daraufhin wird ein Dialogfenster aufgeblendet, in dem Sie alle Eigenschaften des Icons einstellen können.

Element/Text

Dieser Befehl fügt einen Textblock in das aktuell bearbeitete Objekt ein. Um den Textblock an eine andere Stelle im Bearbeitungsfenster zu verschieben, klicken Sie den Textblock mit der linken Maustaste an und verschieben Sie den Textblock bei gedrückter Maustaste.

Ist im Bearbeitungsdialog die Option 'automatischer Umbruch' aktiviert, kann der Textblock in der Größe verändert werden. Um die Größe des Textblocks zu verändern, klicken Sie die Umrandung des Blocks mit der linken Maustaste an und verschieben Sie die Umrandung bei weiterhin gedrückter Maustaste. Nach dem Loslassen der Maustaste wird ein neuer Umbruch für den Textblock berechnet und der Block wird der gewünschten Größe so gut wie möglich angepaßt.

Um die Eigenschaften des Textblocks (Textinhalte, Schriftart usw.) zu verändern, führen Sie mit der Maus einen Doppelklick auf den Textblock aus. Daraufhin wird ein Dialogfenster aufgeblendet, in dem Sie alle Eigenschaften des Textblocks einstellen können.

Element/Rahmen

Dieser Befehl fügt einen 3D-Rahmen in das aktuell bearbeitete Objekt ein. Um den Rahmen an eine andere Stelle im Bearbeitungsfenster zu verschieben, klicken Sie den Rahmen mit der linken Maustaste an und verschieben Sie den Rahmen bei gedrückter Maustaste. Um die Größe des Rahmens zu verändern, klicken Sie die Umrandung des Rahmens mit der linken Maustaste an und verschieben Sie die Umrandung bei weiterhin gedrückter Maustaste. Die Größe des Rahmens paßt sich der Bewegung der Maus an.

Um die Eigenschaften des Rahmens (Typ) zu verändern, führen Sie mit der Maus einen Doppelklick auf den Rahmen aus. Daraufhin wird ein Dialogfenster aufgeblendet, in dem Sie alle Eigenschaften des Rahmens einstellen können.

Element/Schalter

Dieser Befehl fügt einen Schalter in das aktuell bearbeitete Objekt ein. Um den Schalter an eine andere Stelle im Bearbeitungsfenster zu verschieben, klicken Sie den Schalter mit der linken Maustaste an und verschieben Sie den Schalter bei gedrückter Maustaste. Um die Größe des Schalters zu verändern, klicken Sie die Umrandung des Schalters mit der linken Maustaste an und verschieben Sie die Umrandung bei weiterhin gedrückter Maustaste. Die Größe des Schalters paßt sich der Bewegung der Maus an. Grafikschalter können nicht in der Größe verändert werden, da sich dieser Schaltertyp automatisch an die Größe der Bitmap anpaßt, die in die Schalterfläche eingebettet ist.

Um die Eigenschaften des Schalters (Typ, Text, Grafik usw.) zu verändern, führen Sie mit der Maus einen Doppelklick auf den Schalter aus. Daraufhin wird ein Dialogfenster aufgeblendet, in dem Sie alle Eigenschaften des Schalters einstellen können.

Element/Hyperlink-Feld

Dieser Befehl fügt ein maussensitives Feld in das aktuell bearbeitete Objekt ein. Um das Feld an eine andere Stelle im Bearbeitungsfenster zu verschieben, klicken Sie das Feld mit der linken Maustaste an und verschieben Sie das Feld bei gedrückter Maustaste. Um die Größe des Felds zu verändern, klicken Sie die Umrandung des Felds mit der linken Maustaste an und verschieben Sie die Umrandung bei weiterhin gedrückter Maustaste. Die Größe des Felds paßt sich der Bewegung der Maus an.

Um die Eigenschaften des Feldes (ID, Invertieren usw.) zu verändern, führen Sie mit der Maus einen Doppelklick auf das Feld aus. Daraufhin wird ein Dialogfenster aufgeblendet, in dem Sie alle Eigenschaften des Felds einstellen können.

Element/Listbox

Dieser Befehl fügt eine Liste in das aktuell bearbeitete Objekt ein. Um die Liste an eine andere Stelle im Bearbeitungsfenster zu verschieben, klicken Sie die Liste mit der linken Maustaste an und verschieben Sie die Liste bei gedrückter Maustaste. Um die Größe der Liste zu verändern, klicken Sie die Umrandung der Liste mit der linken Maustaste an und verschieben Sie die Umrandung bei weiterhin gedrückter Maustaste. Die Größe des Listenobjekts paßt sich der Bewegung der Maus an.

Um die Eigenschaften der Liste (Listeneinträge, Farbe, Rahmen usw.) zu verändern, führen Sie mit der Maus einen Doppelklick auf die Liste aus. Daraufhin wird ein Dialogfenster aufgeblendet, in dem Sie alle Eigenschaften der Liste einstellen können.

Element/Fenster

Dieser Befehl fügt ein Kindfenster in das aktuell bearbeitete Objekt ein. Um das Fenster an eine andere Stelle im Bearbeitungsfenster zu verschieben, klicken Sie das Fenster mit der linken Maustaste an und verschieben Sie das Fenster bei gedrückter Maustaste. Um die Größe des Fensters zu verändern, klicken Sie die Umrandung des Fensters mit der linken Maustaste an und verschieben Sie die Umrandung bei weiterhin gedrückter Maustaste. Die Größe des Fensters paßt sich der Bewegung der Maus an.

Um die Eigenschaften des Fensters (Name, ID-Nummer usw.) zu verändern, führen Sie mit der Maus einen Doppelklick auf das Fenster aus. Daraufhin wird ein Dialogfenster aufgeblendet, in dem Sie alle Eigenschaften des Fensters einstellen können.

Optionen

In diesem Menü finden sich Funktionen, die der Einstellung der diversen Parameter dienen, welche die Arbeitsweise des Editors direkt beeinflussen.

Folgende Funktionen stehen zur Verfügung:

Programm starten

Passwort bearbeiten

Gitter einstellen

Einstellungen speichern

Optionen/Programm starten

Diese Funktion erlaubt es, das aktuell bearbeitete Programm zu Testzwecken aus der Entwicklungsumgebung heraus zu starten. Falls die Datei neu angelegt wurde und noch keinen Namen hat, wird vor dem Programmstart noch ein Dateiname abgefragt. Beachten Sie bitte auch, daß vor jedem Programmstart die Programmdatei automatisch abgespeichert wird.

Optionen/Passwort bearbeiten

Diese Funktion erlaubt es, ein Passwort für das aktuell bearbeitete Programm zu vergeben. Wenn Sie ein Passwort vergeben haben, dann wird in Zukunft dieses Passwort beim Eröffnen der Programmdatei zur Bearbeitung abgefragt. Das dient dazu, Ihr Programm gegen die unbefugte Einsichtnahme durch Dritte zu schützen.

Achten Sie bitte sorgfältig darauf, daß Sie Ihr Passwort nicht vergessen. Falls das geschehen sollte, können Sie Ihr Programm nämlich nicht mehr bearbeiten. Machen Sie am besten eine Sicherheitskopie der Programmdatei, bevor Sie die Datei mit einem Passwort ausstatten.

Optionen/Gitter einstellen

Diese Funktion ruft ein Dialogfenster auf, in dem Einstellungen für ein Hilfsgitter vorgenommen werden können.

Optionen/Einstellungen speichern

Durch Anklicken dieses Menüpunktes wird die automatische Statussicherung des Ressource-Editors beim Beenden des Editors oder beim Wechseln einer Ressource-Datei, ein- bzw. ausgeschaltet.

Fenster

In diesem Menü werden Funktionen zur Verfügung gestellt, die es erlauben, die eröffneten Editorfenster und Symbole nach unterschiedlichen Kriterien zu ordnen.

Folgende Funktionen stehen zur Verfügung:

Horizontal ordnen

Vertikal ordnen

Überlappend ordnen

Symbole ordnen

Alle schließen

Fenster/Horizontal ordnen

Dieser Befehl ordnet alle offenen Editorfenster auf dem Bildschirm übereinander an, so daß sie sich nicht gegenseitig verdecken.

Fenster/Vertikal ordnen

Dieser Befehl ordnet alle offenen Editorfenster auf dem Bildschirm nebeneinander an, so daß sie sich nicht gegenseitig verdecken.

Fenster/Überlappend ordnen

Dieser Befehl stapelt alle offenen Editorfenster übereinander, so daß jedes die gleiche Größe wie die anderen hat und nur ein Teil des darunterliegenden Fensters sichtbar ist.

Fenster/Symbole ordnen

Der Befehl *Symbole anordnen* ordnet alle Symbole, die minimierte Editorfenster darstellen, auf dem Desktop neu an. Die Symbole werden dabei gleichmäßig, in der linken unteren Ecke des Desktops beginnend, angeordnet.

Fenster/Alle schließen

Der Befehl *Alle schließen* schließt alle geöffneten Editorfenster auf dem Desktop. Die in den Editorfenstern bearbeiteten Module werden vor dem Schließen des jeweiligen Fensters automatisch gespeichert.

Hilfe

Dieses Menü bietet verschiedene Funktionen an, die Zugang zum Windows-Hilfesystem bieten und die des weiteren verschiedene Informationen über das Betriebssystem, auf dem das Autorensystem läuft, liefern.

Folgende Funktionen stehen zur Verfügung:

[Index](#)

[Hilfe verwenden](#)

[Info](#)

[System](#)

Hilfe/Index

Die Online-Hilfe verfügt über einen umfassenden Index zu Aufbau und Inhalt des Hilfe-Menüs.

Sie gelangen aus dem Index in das Hilfesystem, indem Sie eines der hervorgehobenen Wörter oder Ausdrücke anklicken oder mit dem Tabulator anwählen und dann die Eingabetaste drücken.

Wählen Sie den Befehl *Hilfe verwenden* aus dem Hilfe-Menü, wenn Sie nicht wissen, wie ein Hilfesystem unter Windows zu bedienen ist.

Hilfe/Hilfe verwenden

Dieser Befehl beschreibt den Umgang mit dem RSE-Author-Hilfesystem (oder einem anderen Hilfesystem unter Windows).

Hilfe/Info

Dieser Befehl blendet eine kleine Dialogbox auf, die Ihnen Informationen über das Copyright und die Version des Autorensystems liefern.

Hilfe/System

Dieser Befehl blendet ein Fenster auf, in dem Informationen über die Betriebssystemplattform, auf der das Autorensystem läuft, zu finden sind.

Anlegen von Objekten

Nach dem Start der Entwicklungsumgebung befindet sich in der linken oberen Ecke des Programmfensters ein Listenfenster, in dem alle Objekte eines Programms aufgelistet werden. In diesem Fenster befinden sich eine Liste und vier Schalter. In der Liste werden die Objekte der aktuellen Datei angezeigt. Durch Anklicken der Schalter kann ein Objekt eingefügt, zur Bearbeitung eröffnet, gelöscht oder der Name des Objekts geändert werden.

Um ein neues Objekt einzufügen, klicken Sie den 'Einfügen'-Schalter an und halten Sie die Maustaste gedrückt. Nach der Betätigung des Schalters klappt ein kleines Pulldown-Menü herunter, in dem Sie auswählen können, welchen Objekttyp Sie einfügen möchten. Folgende Typen für Objekte gibt es:

Seite

Dieses Objekt stellt den Inhalt für ein Fenster dar. Angenommen, Sie haben im Programm ein Fenster eröffnet, in dem ein Anwender in Texten und Grafiken blättern kann. Für jede Text- oder Grafikseite legen Sie ein Seitenobjekt an. Beim Blättern wird das Ausgabefenster gelöscht und nachfolgend die nächste Seite aus dem Programm geladen.

Fenster

Dieses Objekt stellt ein komplettes Fenster dar. Wird in einem Programm ein solches Objekt geladen, wird das Fenster mit allen Inhalten auf dem Bildschirm eröffnet.

Dokument

Ein solches Objekt enthält eine komplette Seite für die Ausgabe auf den Drucker. Wenn Ihr Programm Trainingsunterlagen ausgeben soll, dann können Sie diese Unterlagen als Objekte in der Programm-Datei anlegen und innerhalb des laufenden Programms aufrufen.

Sie können ein Objekt auch über den Menüpunkt Objekt in der Menüleiste des Ressource-Editors einfügen.

Der RSE-Author-Editor vergibt beim Einfügen eines Objekts automatisch einen Namen. Um diesen Namen zu ändern, positionieren Sie den Auswahlbalken der Liste auf den Objektnamen und klicken Sie danach den Schalter 'Ändern' an. Sie können dann in einem kleinen Fenster den Namen des Objekts ändern.

Um ein Objekt an eine andere Stelle in der Liste zu verschieben, selektieren Sie einfach die Bezeichnung des Objekts mit der Maus. Danach können Sie die Objektbezeichnung mit den Pfeiltasten bei gleichzeitig gedrückter Shift-Taste an eine andere Stelle in der Liste verschieben.

Bearbeiten von Objekten

Um ein Objekt zu bearbeiten, positionieren Sie im Listenfenster den Auswahlbalken auf den Namen des Objekts und betätigen Sie danach den 'Bearbeiten'-Schalter. Alternativ können Sie das Objekt auch mit einem Doppelklick auf den Namen des Objekts in der Liste zur Bearbeitung eröffnen.

Nach der Auswahl des Objekts öffnet sich ein Fenster. Dieses Fenster ist in zwei Teile aufgeteilt. Auf der linken Seite befindet sich eine Schalterleiste. Durch Anklicken eines dieser Schalter wird ein neues Element in das Objekt eingefügt. Dieses Element können Sie dann mit der Maus an eine andere Stelle bewegen und in der Größe verändern. Um die Eigenschaften eines Elements zu ändern, führen Sie einen Doppelklick mit der Maus auf das Element aus. Daraufhin öffnet sich ein Dialogfenster, in dem Sie alle Eigenschaften des Elements (Farbe, Stil, Füllmuster, zugeordnete Bitmap usw.) einstellen oder verändern können.

Die folgenden 11 Elementtypen stehen zur Verfügung:



Fügt eine Linie in das Objekt ein.



Fügt ein Rechteck in das Objekt ein.



Fügt einen Kreis in das Objekt ein.



Fügt eine Bitmap in das Objekt ein. Beim Anlegen dieses Elements wird vorerst ein Platzhalter eingefügt. Durch einen Doppelklick mit der Maus wird der Bearbeitungsdialog aufgeblendet. Hier kann dann der Dateiname der Bitmap-Grafik angegeben werden.



Fügt ein Standard-Icon in das Objekt ein. Es stehen fünf verschiedene Icons (Standard, Information, Warnung, Fehler, Frage) zur Verfügung. Welches Icon dargestellt werden soll, wird im Bearbeitungsdialog festgelegt.



Fügt einen Textblock in das Objekt ein. Um den Text zu bearbeiten, muß der Bearbeitungsdialog aufgeblendet werden. In diesem Dialogfenster kann in einem Editor-Control der Text bearbeitet und verändert werden. Des weiteren kann in dem Dialogfenster auch die Schriftart für den Text eingestellt werden. Innerhalb des Textes können die Textattribute geändert und Textstellen als Hypertextfelder markiert werden.



Fügt einen 3D-Rahmen in das Objekt ein. Diese 3D-Rahmen dienen dazu, ein Fenster optisch zu gliedern. Es stehen insgesamt 24 verschiedene Rahmentypen zur Verfügung.



Fügt einen Schalter in das Objekt ein. Es können wahlweise Textschalter, Symbolschalter oder Schalter mit freidefinierbaren Grafiken eingefügt werden. Von welchem Typ der Schalter sein soll, wird im Bearbeitungsdialog eingestellt.



Fügt ein maussensitives Feld (Hyperlink) in das Objekt ein.



Fügt eine Listbox in das Objekt ein.



Fügt ein Unterfenster in das Objekt ein. Im Bearbeitungsdialog kann dabei eine Referenz zu einem anderen Ressource-Objekt angegeben werden.

Je nach Objekttyp unterscheidet sich die Schalterleiste. Beim Bearbeiten von Gruppen, virtuellen Fenstern und Druckdokumenten können keine Bedienungselemente für ein Programm (Schalter, Liste usw.) eingegeben werden.

Sie können ein Element auch über den Menüpunkt Element in der Menüleiste des Ressource-Editors einfügen.

Eigenschaften des Objekts ändern

Je nach Objekttyp stehen noch zwei weitere Textschalter zur Verfügung. Der Schalter 'Eigenschaften' blendet ein Dialogfenster auf, in dem die Grundeigenschaften des Objekts eingestellt werden können. Im Fall eines Fensterobjekts sind dies z.B. der Fenstertyp, die ID-Nummer des Fensters, die Eigenschaften des Fensters (schließbar, verschiebbar usw.), sowie der Hintergrund.

Objekt testen

Mit dem Schalter 'Test' wird eine Instanz des bearbeiteten Objekts erzeugt. Wird gerade ein Fensterobjekt bearbeitet, dann wird im Testmodus ein Fenster mit den Eigenschaften und Inhalten eröffnet, die während der Bearbeitung festgelegt wurden. Das Testfenster kann durch sein System-Menü oder durch Betätigung der ESC-Taste geschlossen werden. Bei einem Druckobjekt wird die bearbeitete Druckseite auf einen angeschlossenen Drucker ausgegeben.

Eigenschaften eines Elements ändern

Um die Eigenschaften eines Elements zu verändern, führen Sie bitte einen Doppelklick mit der Maus auf dieses Element aus. Daraufhin wird ein Dialogfenster eröffnet, in dem Sie alle Eigenschaften (Schrift, Füllmuster usw.) des Elements einstellen können. Texte, sowie die Zuordnung von Bitmaps werden ebenfalls im jeweiligen Dialog-Fenster eingegeben.

Elementliste aufblenden

Durch Betätigung der rechten Maustaste innerhalb des Bearbeitungsfensters können Sie eine Liste aller Elemente im gerade bearbeiteten Objekt aufblenden. Um ein Element an eine andere Stelle in der Liste zu verschieben, selektieren Sie einfach das Element. Danach

können Sie das Element mit den Pfeiltasten bei gleichzeitig gedrückter Shift-Taste an eine andere Stelle in der Liste verschieben.

Gruppe von Elementen verschieben

Um mehrere Elemente gleichzeitig zu verschieben, drücken Sie bitte die Shift-Taste. Halten Sie diese Taste gedrückt und klicken Sie mit der Maus alle Elemente an, die verschoben werden sollen. Beim Anklicken des letzten Elements halten Sie die Maustaste gedrückt und verschieben Sie die Elementgruppe an die Stelle, an der sie die Gruppe haben wollen.

Element löschen

Selektieren Sie das Element und drücken Sie die Lösch-Taste (Entf).

Tastaturbelegung

Während der Bearbeitung eines Objekts stehen Ihnen folgende Tastenkombinationen zur Verfügung:

Pfeil rechts

Selektiertes Element nach rechts verschieben.

Pfeil links

Selektiertes Element nach links verschieben.

Pfeil hoch

Selektiertes Element nach oben verschieben.

Pfeil runter

Selektiertes Element nach unten verschieben.

Shift-Pfeil rechts

Breite des selektierten Elements vergrößern.

Shift-Pfeil links

Breite des selektierten Elements verkleinern.

Shift-Pfeil hoch

Höhe des selektierten Elements verkleinern.

Shift-Pfeil runter

Höhe des selektierten Elements vergrößern.

Strg-Pfeil rechts

Voriges Element selektieren.

Strg-Pfeil links

Nächstes Element selektieren.

Strg-Pfeil hoch

Voriges Element selektieren.

Strg-Pfeil runter

Nächstes Element selektieren.

Leertaste

Schaltet das aktuelle Bearbeitungsfenster in den Vollbildmodus und zurück. Im Vollbildmodus steht die gesamte Bildschirmfläche zur Bearbeitung des Objekts zur Verfügung.

Element 'Linie' bearbeiten

In diesem Dialogfenster können Sie die Eigenschaften eines Linienelements einstellen. Folgende Einstellungen sind möglich:

Koordinaten

X/Y-Position des Elements.

Dimensionen

Breite/Höhe des Elements.

Orientierung

Gibt die Orientierung (von oben nach unten oder von unten nach oben) der Linie im umschließenden Rechteck an.

Linienstil

Legt das Aussehen der Linie fest.

Element 'Rechteck' bearbeiten

In diesem Dialogfenster können Sie die Eigenschaften eines Rechtecks einstellen. Folgende Einstellungen sind möglich:

Koordinaten

X/Y-Position des Elements.

Dimensionen

Breite/Höhe des Elements.

Eckenverrundung

Legt die Rundung der Ecken fest.

3D-Rahmen

Erlaubt die Definition eines optionalen 3D-Rahmens, der das Rechteck zusätzlich umgibt.

Linienstil

Legt das Aussehen der umgebenden Linie fest.

Füllmuster

Legt das Aussehen des Füllmusters fest.

Element 'Kreis' bearbeiten

In diesem Dialogfenster können Sie die Eigenschaften einer Ellipse einstellen. Folgende Einstellungen sind möglich:

Koordinaten

X/Y-Position des Elements.

Dimensionen

Breite/Höhe des Elements.

Linienstil

Legt das Aussehen der umgebenden Linie fest.

Füllmuster

Legt das Aussehen des Füllmusters fest.

Element 'Grafik' bearbeiten

In diesem Dialogfenster können Sie die Eigenschaften einer Bitmap-Grafik einstellen. Folgende Einstellungen sind möglich:

Koordinaten

X/Y-Position des Elements.

Dimensionen

Breite/Höhe des Elements.

3D-Rahmen

Erlaubt die Definition eines optionalen 3D-Rahmens, der die Grafik zusätzlich umgibt.

Dateiname der Grafikdatei

Hier wird der Dateiname der Grafik eingegeben. Mit dem kleinen Schalter rechts von diesem Eingabefeld kann eine Dateiauswahlbox aufgerufen werden.

Dateiname einer Grafikdatei, deren Palette übernommen werden soll

Dieses Eingabefeld ist optional. Hier kann der Dateiname einer Grafik eingegeben werden, an deren Palette die Bitmap angepaßt werden soll. Mit dem kleinen Schalter rechts von diesem Eingabefeld kann eine Dateiauswahlbox aufgerufen werden. Die Angabe einer externen Palette ist nur sinnvoll, wenn das Feld 'Palette einstellen' selektiert wurde.

Anzeige

Hier wird festgelegt, ob die Grafik bei der Anzeige mittels Colour-Keying (eine Farbe wird als transparent definiert) mit dem Hintergrund verknüpft werden soll. Dabei kann zwischen Schwarz und Weiß als transparenter Farbe gewählt werden.

Aus Performance-Gründen wird im Bearbeitungsfenster des Ressource-Editors auf die transparente Verknüpfung einer Grafik mit dem Hintergrund verzichtet. Grafiken werden während der Bearbeitung immer undurchsichtig angezeigt. Um sich das Ergebnis einer solchen Verknüpfung anzusehen, betätigen Sie bitte den 'Test'-Schalter im Bearbeitungsfenster.

Palette einstellen

Ist diese Checkbox aktiviert, dann wird die Palette der Grafik im bearbeiteten Objekt eingestellt.

Bitmap skalieren

Ist diese Checkbox aktiviert, dann ist es möglich, die Bitmap im Bearbeitungsfenster durch einfaches Vergrößern oder Verkleinern des Elements mit der Maus zu skalieren.

Definitionen für ein maussensitives Feld

Eine Bitmap kann hier zusätzlich als Hyperlink-Feld definiert werden.

Script

Wurde die Bitmap als Hyperlink-Feld definiert, dann kann durch Anklicken dieses Schalters ein Editorfenster aufgerufen werden, in dem die Befehls-Scripts des Hyperlink-Feldes bearbeitet werden können.

Element 'Icon' bearbeiten

In diesem Dialogfenster können Sie die Eigenschaften eines Standard-Icons einstellen.
Folgende Einstellungen sind möglich:

Koordinaten

X/Y-Position des Elements.

Dimensionen

Breite/Höhe des Elements.

Typ des Symbols

Folgende Typen sind möglich: Standard, Information, Warnung, Fehler, Frage.

Element 'Text' bearbeiten

In diesem Dialogfenster können Sie die Eigenschaften eines Textblocks einstellen. Folgende Einstellungen sind möglich:

Koordinaten

X/Y-Position des Elements.

Dimensionen

Breite/Höhe des Elements.

Automatischer Umbruch

Ist diese Checkbox aktiviert, dann wird beim Verändern der Größe eines Textblocks ein neuer Umbruch für den Text berechnet.

Schriftart

Durch Anklicken des Schalters im Schriftartenfeld wird ein Dialogfenster aufgeblendet, in dem die Schriftart für den Text eingegeben werden kann.

Schriftattribut für Hypertextfelder

Durch Anklicken des Schalters im Hypertextfeld wird ein Dialogfenster aufgeblendet, in dem zusätzliche Schriftattribute für das Aussehen von Hypertextfeldern definiert werden können.

Bearbeitung des Textes

In diesem Dialogfenster ist ein Editorfenster eingebettet, in dem der Text des Textblocks bearbeitet werden kann. Während der Bearbeitung des Textes können an beliebiger Stelle die Textattribute geändert werden. Des weiteren können Hypertextfelder definiert werden.

Hypertext-Liste

Rechts neben dem Editorfenster findet sich eine Listbox, in der alle Hypertext-Felder angezeigt werden, die im editierten Text definiert wurden. Hypertextfelder werden im Text dadurch definiert, daß der entsprechende Begriff in eckige Klammern ('[.....]') gesetzt wird. Um die Liste der Hypertext-Felder zu aktualisieren, muß der Schalter mit der Beschriftung 'Aktualisieren' angeklickt werden. Durch einen Doppelklick auf ein in der Liste eingetragenes Hypertextfeld wird ein Dialogfenster eröffnet, in dem die Eigenschaften des Feldes bearbeitet werden können.

Element Hypertext bearbeiten

In diesem Dialogfenster können Sie die Eigenschaften eines Hypertextfeldes einstellen. Folgende Einstellungen sind möglich:

Alias

Elementbezeichnung, unter der das Hypertextfeld innerhalb eines Programms angesprochen werden kann.

Maus

Art des Mauszeigers, der beim Bewegen des Mauszeigers über das Hypertextfeld angezeigt werden soll.

Individuelle Attribute

Standardmäßig werden für die Darstellung der Hypertextfelder die Attributeinstellungen verwendet, die innerhalb des Dialogfensters für den Textblock definiert wurden. Wird diese Checkbox aktiviert, können durch Anklicken des Schalters 'Schriftart' individuelle Schriftattribute für dieses Hypertextfeld definiert werden.

Schriftart

Durch Anklicken dieses Schalters wird ein Dialogfenster aufgeblendet, in dem zusätzliche Schriftattribute für das Aussehen von Hypertextfeldern definiert werden können.

Script

Durch Anklicken dieses Schalters wird ein Editorfenster aufgerufen, in dem die Befehls-Scripts des Hypertextfeldes bearbeitet werden können.

Element '3D-Rahmen' bearbeiten

In diesem Dialogfenster können Sie die Eigenschaften eines 3D-Rahmens einstellen. Folgende Einstellungen sind möglich:

Koordinaten

X/Y-Position des Elements.

Dimensionen

Breite/Höhe des Elements.

Rahmentyp

Folgende Rahmentypen sind möglich:

- Rechteck
- abgesenkte Fläche, 1 Pixel tief
- abgesenkte Fläche, 2 Pixel tief
- abgesenkte Fläche, 3 Pixel tief
- gehobene Fläche, 1 Pixel hoch
- gehobene Fläche, 2 Pixel hoch
- gehobene Fläche, 3 Pixel hoch
- abgesenkte Fuge, 1 Pixel tief
- abgesenkte Fuge, 2 Pixel tief
- abgesenkte Fuge, 3 Pixel tief
- gehobene Fuge, 1 Pixel hoch
- gehobene Fuge, 2 Pixel hoch
- gehobene Fuge, 3 Pixel hoch
- abgesenkte Rasterung, 2 Einheiten
- abgesenkte Rasterung, 3 Einheiten
- abgesenkte Rasterung, 4 Einheiten
- gehobene Rasterung, 2 Einheiten
- gehobene Rasterung, 3 Einheiten
- gehobene Rasterung, 4 Einheiten
- Rechteck mit Schatten, 1 Pixel
- Rechteck mit Schatten, 2 Pixel
- Rechteck mit Schatten, 3 Pixel
- Rechteck mit Schatten, 4 Pixel
- Rechteck mit Schatten, 5 Pixel
- abgesenkte Fläche mit eingebettetem Rechteck

Element 'Schalter' bearbeiten

In diesem Dialogfenster können Sie die Eigenschaften eines Schalters einstellen. Folgende Einstellungen sind möglich:

Koordinaten

X/Y-Position des Elements.

Dimensionen

Breite/Höhe des Elements.

Alias

Elementbezeichnung, unter der der Schalter innerhalb eines Programms angesprochen werden kann.

Schalter-Typ

Folgende Schaltertypen sind zulässig:

- Text-Schalter
- Grafik-Schalter
- Symbol-Schalter

Je nachdem, welcher Typ hier ausgewählt wird, werden die zugehörigen Bearbeitungsfelder auf der rechten Seite des Dialogfensters aktiviert.

Text für Text-Schalter

Hier wird der Text für einen Schalter eingegeben. Rechts von dem Eingabefeld befindet sich ein kleiner Schalter, der ein Dialogfenster aufruft, in dem die Schriftart für den Schalter definiert werden kann.

Dateinamen für Grafik-Schalter

Hier können in drei Eingabefeldern die Dateinamen der Schaltergrafiken eingegeben werden. Die erste Schaltergrafik stellt den Grundzustand des Schalters dar. Diese Grafik **muß** definiert werden. Die zweite Grafik wird beim Eindrücken eines Schalters dargestellt. Die dritte Grafik wird beim Sperren eines Schalters angezeigt. Die Angabe der zweiten und dritten Grafik ist optional. Neben den Eingabefeldern für die Dateinamen befindet sich je ein kleiner Schalter, über den eine Dateiauswahlbox aufgerufen werden kann.

Anzeige

Hier wird festgelegt, wie ein Grafikschafter angezeigt werden soll. Folgende Möglichkeiten bestehen:

3D-Rahmen

Die Schaltergrafiken werden mit einem 3D-Rahmen umgeben (Standardaussehen von Schaltern).

undurchsichtig

Die Schaltergrafiken werden ohne Rahmen dargestellt. Das Aussehen des Schalters wird vollständig durch die Grafiken definiert.

durchsichtig schwarz

Die Schaltergrafiken werden ohne Rahmen dargestellt. Das Aussehen des Schalters wird vollständig durch die Grafiken definiert. Die Farbe 'Schwarz' wird als transparent definiert, d.h. überall wo die Grafik die Farbe 'Schwarz' enthält, scheint der Hintergrund durch.

durchsichtig weiß

Die Schaltergrafiken werden ohne Rahmen dargestellt. Das Aussehen des Schalters wird vollständig durch die Grafiken definiert. Die Farbe 'Weiß' wird als transparent definiert, d.h. überall wo die Grafik die Farbe 'Weiß' enthält, scheint der Hintergrund durch.

Durch Verwendung einer Transparenzfarbe können beliebig geformte Schalter (rund, oval usw.) erzeugt werden.

Aus Performance-Gründen wird im Bearbeitungsfenster des Ressource-Editors auf die transparente Verknüpfung der Schaltergrafiken mit dem Hintergrund verzichtet. Grafiken werden während der Bearbeitung immer undurchsichtig angezeigt. Um sich das Ergebnis einer solchen Verknüpfung anzusehen, betätigen Sie bitte den 'Test'-Schalter im Bearbeitungsfenster.

Statusmodus

Unter den Eingabefeldern für die Dateinamen befindet sich noch eine Checkbox. Wird diese Checkbox aktiviert, dann wird der Grafikscharter in den sogenannten 'Statusmodus' versetzt. In diesem Modus bleibt der Schalter nach dem Anklicken mit der Maus versenkt und springt erst beim erneuten Anklicken wieder heraus.

Typangabe für Symbolschalter

Hier kann der Typ eines Schalters mit einem vordefinierten Symbol angegeben werden. Folgende Typen sind möglich:

- Pfeil hoch
- Pfeil runter
- Pfeil link
- Pfeil rechts

Script

Durch Anklicken dieses Schalters wird ein Editorfenster aufgerufen, in dem das Befehls-Script des Schalters bearbeitet werden kann.

Element 'Hyperlink-Feld' bearbeiten

In diesem Dialogfenster können Sie die Eigenschaften eines maussensitiven Feldes einstellen. Folgende Einstellungen sind möglich:

Koordinaten

X/Y-Position des Elements.

Dimensionen

Breite/Höhe des Elements.

Alias

Elementbezeichnung, unter der das Feld innerhalb eines Programms angesprochen werden kann.

Mauszeiger

Hier kann das Aussehen des Mauszeigers innerhalb des Feldes definiert werden.

Element 'Liste' bearbeiten

In diesem Dialogfenster können Sie die Eigenschaften eines Listenelements einstellen. Folgende Einstellungen sind möglich:

Koordinaten

X/Y-Position des Elements.

Dimensionen

Breite/Höhe des Elements.

Alias

Elementbezeichnung, unter der die Liste innerhalb eines Programms angesprochen werden kann.

Hintergrundfarbe

Farbe des Listenhintergrunds.

3D-Rahmen

Definition des Rahmens der Liste.

Liste sortieren

Wird diese Checkbox aktiviert, dann werden die Einträge, die in die Liste eingefügt werden, alphabetisch sortiert.

Texteingabe in Liste

Um einen Text in die Liste einzufügen, muß der Text in das Eingabefeld in der rechten oberen Ecke des Dialogfensters eingetragen werden. Durch Anklicken des Einfügen-Schalters wird der Text in die Liste auf der rechten Seite des Dialogfensters eingetragen. Mit dem Ändern-Schalter kann ein Eintrag nachträglich geändert werden. Mit dem Löschen-Schalter kann ein Eintrag wieder entfernt werden.

Schriftart

Durch Anklicken des Schriftart-Schalters wird ein Dialogfenster aufgeblendet, in dem die Schriftart für das Eingabefeld definiert werden kann.

Script

Durch Anklicken dieses Schalters wird ein Editorfenster aufgerufen, in dem die Befehls-Scripts für den aktuell selektierten Listeneintrag bearbeitet werden können.

Element 'Fenster' bearbeiten

In diesem Dialogfenster können Sie die Eigenschaften eines Kindfensters einstellen. Es besteht dabei die Möglichkeit, ein neues Fenster zu definieren oder einen Querverweis auf ein bestehendes Fenster herzustellen. Folgende Einstellungen sind möglich:

Koordinaten

X/Y-Position des Elements.

Dimensionen

Breite/Höhe des Elements.

Alias

Bezeichnung, unter der das Fenster innerhalb eines Programms angesprochen werden kann. Falls ein Querverweis auf ein bestehendes Fensterobjekt angegeben wird (siehe unten), dann muß dieses Feld leer bleiben, da in dem Fall die Bezeichnung des Objektes verwendet wird.

Objekt

Hier kann der Name eines bereits bestehenden Programm-Objekts eingegeben werden. Dabei muß es sich um ein Fensterobjekt vom Typ 0 oder 1 (Kindfenster) handeln. Dieses Programm-Objekt muß in einem eigenen Bearbeitungsfenster editiert werden.

Wurde in diesem Feld ein Verweis auf ein anderes Programm-Objekt definiert, dann werden die Einstellungen für das Alias, die Rollbalken und den Fensterhintergrund ignoriert, da in diesem Fall die Einstellungen des definierten Fensterobjekts verwendet werden.

3D-Rahmen

Definition eines optionalen 3D-Rahmens für das Kindfenster.

Rollbalken

Durch Aktivieren dieser Checkboxen werden dem Fenster horizontale und vertikale Rollbalken zugeordnet. Neben den Checkboxen befinden sich je zwei Eingabefelder. In dem ersten Eingabefeld wird die Anzahl der Pixel eingetragen, um die mit dem Rollbalken verschoben werden kann. Im zweiten Eingabefeld wird die minimale Anzahl der Pixel angegeben, um die verschoben wird.

Beispiel: Sie tragen im ersten Feld den Wert 320 ein, im zweiten den Wert 16, d.h. der Fensterinhalt kann um insgesamt 320 Pixel verschoben werden. Bei jeder Scrollaktion wird um mindestens 16 Pixel verschoben.

Die hier vorgenommenen Einstellungen werden nur verwendet, wenn kein Querverweis auf ein bestehendes Fensterobjekt definiert wurde. Besteht ein solcher Querverweis, werden die Einstellungen des Objekts verwendet.

Hintergrund

Hier kann das Füllmuster für den Fensterhintergrund eingestellt werden.

Die hier vorgenommenen Einstellungen werden nur verwendet, wenn kein Querverweis auf ein bestehendes Fensterobjekt definiert wurde. Besteht ein solcher Querverweis, werden die Einstellungen des Objekts verwendet.

Objekt Seite bearbeiten

Koordinaten

X/Y-Position des Objekts.

Suchbegriff

Hier kann ein Suchbegriff für die Seite eingegeben werden. Innerhalb eines Programms kann mit dem Befehl SelectPage eine Liste mit allen Suchbegriffen einer bestimmten Gruppe von Seiten aufgeblendet werden.

Testen in

Um das Seitenobjekt zu testen, muß hier der Name des Fensterobjekts eingetragen werden, in das die Seite später im Programm geladen werden soll.

Rollbalken

Durch Aktivieren dieser Checkboxen werden dem Fenster horizontale und vertikale Rollbalken zugeordnet. Neben den Checkboxen befinden sich je zwei Eingabefelder. In dem ersten Eingabefeld wird die Anzahl der Pixel eingetragen, um die mit dem Rollbalken verschoben werden kann. Im zweiten Eingabefeld wird die minimale Anzahl der Pixel angegeben, um die verschoben wird.

Beispiel: Sie tragen im ersten Feld den Wert 320 ein, im zweiten den Wert 16, d.h. der Fensterinhalt kann um insgesamt 320 Pixel verschoben werden. Bei jeder Scrollaktion wird um mindestens 16 Pixel verschoben.

Script

Durch Anklicken dieses Schalters wird ein Editorfenster aufgerufen, in dem die Befehls-Scripts für die Seite bearbeitet werden können.

Objekt 'Fenster' bearbeiten

In diesem Dialogfenster können Sie die Eigenschaften eines Fensterobjekts einstellen. Folgende Einstellungen sind möglich:

Koordinaten

X/Y-Position des Objekts. Die Koordinaten des Objekts können auch im Testmodus durch einfaches Verschieben mit der Maus geändert werden.

Dimensionen

Breite/Höhe des Objekts. Die Dimensionen des Objekts können auch im Testmodus durch einfaches Vergrößern/Verkleinern mit der Maus geändert werden.

Fenster-Typ

Es sind fünf verschiedene Fenstertypen zulässig:

- 0 - Einfachfenster ohne Rahmen
- 1 - Einfachfenster mit Strichrahmen
- 2 - Standardfenster
- 3 - Dialogfenster
- 4 - 3D-Dialogfenster

Alias

Das Fensterobjekt kann vom Programm aus entweder durch den Objektnamen oder durch eine Kurzbezeichnung, die hier eingegeben werden kann, angesprochen werden.

Titel

Hier wird der Text eingegeben, der in der Titelzeile des Fensters erscheinen soll.

Mauszeiger

Hier kann ein Mauszeiger für das Fenster ausgewählt werden. Standardmäßig ist der Pfeil eingestellt.

Rollbalken

Durch Aktivieren dieser Checkboxes werden dem Fenster horizontale und vertikale Rollbalken zugeordnet. Neben den Checkboxes befinden sich je zwei Eingabefelder. In dem ersten Eingabefeld wird die Anzahl der Pixel eingetragen, um die mit dem Rollbalken verschoben werden kann. Im zweiten Eingabefeld wird die minimale Anzahl der Pixel angegeben, um die verschoben wird.

Beispiel: Sie tragen im ersten Feld den Wert 320 ein, im zweiten den Wert 16, d.h. der Fensterinhalt kann um insgesamt 320 Pixel verschoben werden. Bei jeder Scrollaktion wird um mindestens 16 Pixel verschoben.

Größe = Ausgabebereich

Ist diese Checkbox aktiviert, dann werden die Angaben für die Breite und Höhe des Fensters als Größe des Ausgabebereiches des Fensters (ohne Rahmen und Titelleiste)

interpretiert. Das ist z.B. sinnvoll, wenn unabhängig von Art und Größe des Rahmens und Titels die Ausgabefläche eines Fensters immer die gleiche Größe haben soll.

Clipping aktivieren

Ist dies Checkbox aktiviert (Standardeinstellung) dann darf das Fensterobjekt nicht innerhalb seiner Kind-Objekte (Schalter, Unterfenster usw.) zeichnen. Durch Deaktivierung dieser Checkbox können grafische Ausgaben auch über die Kind-Objekte gemacht werden.

Unsichtbar laden

Wird diese Checkbox aktiviert, dann wird das Fenster nach dem Laden nicht angezeigt. Das dient dazu, das der Aufbau der Fensterinhalte unsichtbar im Hintergrund erfolgen kann. Das Fenster muß dann vom Programm mit dem Befehl Show sichtbar geschaltet werden.

Mittig zentrieren

Ist diese Checkbox aktiviert, dann wird das Fenster beim Laden automatisch auf dem Bildschirm (Fenster-Typ 2 und 3) bzw. im Vaterfenster (Fenster-Typ 0 und 1) zentriert.

Fenster-Parameter

Hier können die Eigenschaften des Fensters eingestellt werden. Dazu gehören:

- Fenster schließbar
- Fenster verschiebbar
- Fenster in der Größe veränderbar
- Fenster hat einen Minimal-Schalter
- Fenster hat einen Maximal-Schalter
- Fenster hat Tabulatorkontrolle

Hintergrund

Hier kann das Füllmuster für den Fensterhintergrund eingestellt werden.

Script

Durch Anklicken dieses Schalters wird ein Editorfenster aufgerufen, in dem die Befehls-Scripts für das Fenster bearbeitet werden können.

Script-Editor

In diesem Dialogfenster kann ein Befehls-Script bearbeitet werden. Dieses Script wird bei der Auslösung bestimmter Systemereignisse (Fenster öffnen, Schalter anklicken usw.) ausgeführt und beinhaltet Anweisungen, die den Ablauf eines RSE-Author-Programms steuern. Eine Liste aller RSE-Author-Befehle finden Sie im [Befehls-Index](#) dieses Online-Handbuchs.

Gitter einstellen

In diesem Dialogfenster können Einstellungen für ein Hilfsgitter vorgenommen werden, das die Positionierung der Elemente in einem Bearbeitungsfenster erleichtert.

Elementliste bearbeiten

In dieser Liste, die durch einen rechten Mausklick in ein Bearbeitungsfenster eröffnet wird, können Sie die Reihenfolge der Elemente in einem Programm-Objekt verändern.

Um ein Objekt an eine andere Stelle in der Liste zu verschieben, selektieren Sie einfach die Bezeichnung des Elements mit der Maus. Danach können Sie die Elementbezeichnung mit den Pfeiltasten bei gleichzeitig gedrückter Shift-Taste an eine andere Stelle in der Liste verschieben.

Objekte einlagern

In dieser Dialogbox wählt man Objekte aus einer anderen Programm-Datei aus, die man in die gerade bearbeitete Datei einlagern möchte.

Die Dialogbox besitzt zu diesem Zweck zwei Listen. In der linken Liste sind alle Objekte der Programm-Datei, aus der eingelagert werden soll, aufgelistet. In der rechten Liste werden alle Objekte eingetragen, die tatsächlich übernommen werden sollen.

Mit dem Einfügen- und Löschen-Schalter kann man Objekte in die rechte Liste einfügen oder wieder entfernen.

Hat man alle Objekte ausgewählt, dann kann die Übernahme der Objekte durch Anklicken des OK-Schalters angestoßen werden. Um den Vorgang abubrechen, betätigen Sie den Abbrechen-Schalter.

Objektbezeichnung ändern

In diesem Dialogfenster kann die Bezeichnung eines Programm-Objekts geändert werden.

Schriften einstellen

In diesem Dialogfenster können alle Parameter für eine Schrift interaktiv eingestellt werden. In einem Unterfenster wird eine Schriftprobe angezeigt, die den Einstellungen ständig angepaßt wird.

Auf diese Weise kann die gewünschte Schrift für eine Textausgabe, eine Schalterbeschriftung oder ein Listenelement interaktiv erzeugen.

Durch Anklicken des Einstellen-Schalters wird ein weiteres Dialogfenster (Schriften zuordnen) aufgerufen, in dem festgelegt werden kann, welche der Systemschriften das Autorensystem verwenden soll.

Durch Anklicken des Standard-Schalters kann die aktuell eingestellte Schrift zur Standard-Schrift erklärt werden. Diese Schrift wird in der Folge beim Anlegen von Textblöcken, Schaltern, Listen und Eingabefeldern automatisch eingestellt.

Hypertext-Attribute einstellen

In diesem Dialogfenster können auf der Basis der aktuell eingestellten Schriftart zusätzliche Attribute für Hypertextfelder eingestellt werden. In einem Unterfenster wird eine Schriftprobe angezeigt, die den Einstellungen ständig angepaßt wird.

Durch Anklicken des Standard-Schalters können die aktuell eingestellten Attribute zu Standard-Attributen erklärt werden. Diese Attribute werden in der Folge beim Anlegen von Textblöcken automatisch für die Hypertextfelder eingestellt.

Schriften zuordnen

In diesem Fenster wird festgelegt, welche der im System vorhandenen Schriften vom Autorensystem verwendet werden sollen.

Die Dialogbox enthält zwei Listen. In der linken Dialogbox sind alle Schriften aufgeführt, die auf dem aktuellen Rechnersystem vorhanden sind. In der rechten Liste werden die Schriften eingetragen, wie sie vom Autorensystem verwendet werden sollen.

Diese Einstellungen werden in einer Datei namens 'juggler.fnt' gespeichert. Diese Datei muß immer zusammen mit dem Laufzeitsystem (exw.exe) und dem zugehörigen Lernprogramm ausgeliefert werden.

Ist keine Datei 'juggler.fnt' vorhanden, dann wird die Reihenfolge der im aktuellen Computersystem vorhandenen Schriften verwendet. Diese Reihenfolge kann aber bei verschiedenen Rechnersystemen voneinander abweichen.

Um die Änderungen in der Schriftenliste abzuspeichern, muß die Dialogbox mit dem OK-Schalter geschlossen werden.

Copyright-Information anzeigen

In dieser Dialogbox werden Informationen zum Copyright und der Version des Autorensystems angezeigt.

System-Information anzeigen

In dieser Dialogbox werden Ihnen Informationen über das Computersystem und das Betriebssystem, auf dem das Autorensystem läuft, angezeigt.

Folgende Informationen werden angezeigt:

Typ des Computersystems
Mikroprozessor
Grafische Oberfläche
Grafikauflösung

Textattribute ändern

Während der Eingabe eines Textblocks können jederzeit die Attribute sowie die Hinter- und Vordergrundfarbe des Textes verändert werden. Das ist z.B. sinnvoll, wenn ein einzelnes Wort unterstrichen, kursiv oder in einer anderen Farbe dargestellt werden soll. Attributanweisungen werden innerhalb des Textes in geschweifte Klammern eingeschlossen. Folgende Befehlskürzel sind innerhalb der Klammern erlaubt:

- w0 - Strichstärke leicht
- w1 - Strichstärke normal
- w2 - Strichstärke fett

- +i - kursive Darstellung einschalten
- i - kursive Darstellung abschalten

- +c - kursive Darstellung einschalten
- c - kursive Darstellung abschalten

- +u - Unterstreichung einschalten
- u - Unterstreichung abschalten

- +o - Umrandung einschalten
- o - Umrandung abschalten

- +k - Durchstreichung einschalten
- k - Durchstreichung abschalten

- +s - Schattenwurf einschalten
- s - Schattenwurf abschalten

- +t - Transparenz einschalten
- t - Transparenz abschalten

- fx - Vordergrundfarbe x einstellen
- gx - Hintergrundfarbe x einstellen

Beispiel

{+t}Dieses Wort wird {+u}unterstrichen{-u} dargestellt. Die folgenden Worte werden dagegen {w2f4}fett und in der Farbe rot{w1f0} dargestellt. Die nächste Sequenz {+o+s+k}kombiniert mehrere Attribute{-o-s-k}.

Hypertextfelder definieren

Während der Eingabe eines Textblocks können sogenannte Hypertextfelder definiert werden. Dabei handelt es sich um farblich (oder durch ein anderes Attribut) hervorgehobene Stichworte, zu denen genauere Erklärungen abgerufen werden können. Um ein Hypertextfeld zu definieren, muß der betreffende Begriff einfach in eckige Klammern ('[.....]') eingeschlossen werden. Wollen Sie innerhalb eines Textblocks eckige Klammern als Zeichen ausgeben, dann setzen Sie einfach ein Backslash vor das Zeichen ('\[, '\]').

Zur Definition der Farbe und Schriftattribute von Hypertextfeldern kann ein Dialogfenster aufgerufen werden, in dem diese Parameter interaktiv eingestellt werden können.

Durch Anklicken des Schalters 'Aktualisieren' im Bearbeitungsfenster für Textblöcke werden alle Hypertextfelder in einer Listbox rechts neben dem Editorfenster aufgelistet. Durch Doppelklicken eines Hypertextfeldes in dieser Liste wird ein Dialogfenster zur Bearbeitung des Feldes eröffnet.

Objekt testen

Das von Ihnen bearbeitete Objekt befindet sich zur Zeit im Testmodus. In diesem Modus wird eine Instanz des Objekts auf dem Bildschirm eröffnet, wobei die Eigenschaften des Objekts so gut wie möglich eingestellt werden. Die Fenster des Ressource-Editors sind in diesem Modus gesperrt. Um in den Ressource-Editor zurückzukehren, muß der Testmodus zuvor beendet werden. Dies erfolgt, indem das Testfenster geschlossen wird. Das Testfenster kann entweder durch sein System-Menü oder durch Betätigung der ESC-Taste geschlossen werden.

Sollte beim Drücken der ESC-Taste keine Reaktion erfolgen, dann klicken Sie bitte mit der Maus die Innenfläche des Testfensters an und drücken Sie danach erneut die ESC-Taste.

Wenn ein Dialogfenster mit dem Attribut 'nicht schließbar' versehen ist, dann muß das Fenster durch Drücken der Leertaste (SPACE) geschlossen werden.

