

ABAP als OLE-Automation-Controller



HELP.BCFESDE6

Release 4.6C



Copyright

© Copyright 2001 SAP AG. Alle Rechte vorbehalten.

Weitergabe und Vervielfältigung dieser Publikation oder von Teilen daraus sind, zu welchem Zweck und in welcher Form auch immer, ohne die ausdrückliche schriftliche Genehmigung durch SAP AG nicht gestattet. In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden.

Die von SAP AG oder deren Vertriebsfirmen angebotenen Software-Produkte können Software-Komponenten auch anderer Software-Hersteller enthalten.

Microsoft[®], WINDOWS[®], NT[®], EXCEL[®], Word[®], PowerPoint[®] und SQL Server[®] sind eingetragene Marken der Microsoft Corporation.

IBM[®], DB2[®], OS/2[®], DB2/6000[®], Parallel Sysplex[®], MVS/ESA[®], RS/6000[®], AIX[®], S/390[®], AS/400[®], OS/390[®] und OS/400[®] sind eingetragene Marken der IBM Corporation.

ORACLE[®] ist eine eingetragene Marke der ORACLE Corporation.

INFORMIX[®]-OnLine for SAP und Informix[®] Dynamic Server[™] sind eingetragene Marken der Informix Software Incorporated.

UNIX[®], X/Open[®], OSF/1[®] und Motif[®] sind eingetragene Marken der Open Group.

HTML, DHTML, XML, XHTML sind Marken oder eingetragene Marken des W3C[®], World Wide Web Consortium, Massachusetts Institute of Technology.

JAVA[®] ist eine eingetragene Marke der Sun Microsystems, Inc.

JAVASCRIPT[®] ist eine eingetragene Marke der Sun Microsystems, Inc., verwendet unter der Lizenz der von Netscape entwickelten und implementierten Technologie.

SAP, SAP Logo, R/2, RIVA, R/3, ABAP, SAP ArchiveLink, SAP Business Workflow, WebFlow, SAP EarlyWatch, BAPI, SAPPHIRE, Management Cockpit, mySAP.com Logo und mySAP.com sind Marken oder eingetragene Marken der SAP AG in Deutschland und vielen anderen Ländern weltweit. Alle anderen Produkte sind Marken oder eingetragene Marken der jeweiligen Firmen.

Symbole

Symbol	Bedeutung
	Achtung
	Beispiel
	Hinweis
	Empfehlung
	Syntax

Inhalt

ABAP als OLE-Automation-Controller	5
ABAP als OLE-Automation-Controller	6
Einführung	7
Externe Anwendungen in R/3 registrieren.....	8
OLE-Typinformation zur Konvertierung in R/3 laden	10
Konvertierungsregeln	11
Arbeiten mit dem OLE-Object Browser.....	12
Implementierung	13
Beispiele.....	14
R/3 als OLE-Automation-Server.....	15
Verwandte ABAP-Schlüsselwörter	16
Einführung	17
CREATE OBJECT	18
SET PROPERTY.....	20
GET PROPERTY	22
CALL METHOD	24
FREE OBJECT	26

ABAP als OLE-Automation-Controller

ABAP als OLE-Automation-Controller

Die Programmiersprache ABAP unterstützt die OLE2-Automationstechnik. Daher können Sie aus R/3 heraus Desktop-Anwendungen aufrufen, die OLE2 unterstützen.

Themenübersicht

[Einführung \[Seite 7\]](#)

[Externe Anwendungen in R/3 registrieren \[Seite 8\]](#)

[OLE-Typinformation zur Konvertierung in R/3 laden \[Seite 10\]](#)

[Konvertierungsregeln \[Seite 11\]](#)

[Arbeiten mit dem OLE-Object Browser \[Seite 12\]](#)

[Implementierung \[Seite 13\]](#)

[Beispiele \[Seite 14\]](#)

[R/3 als OLE-Automation-Server \[Seite 15\]](#)

[Verwandte ABAP-Schlüsselwörter \[Seite 16\]](#)



Wenn Sie diese Thematik anhand eines konkreten Programmbeispiels kennenlernen möchten, dann lesen die entsprechende Lektion im [Tutorial zu Kommunikationsschnittstellen. \[Extern\]](#)

Einführung

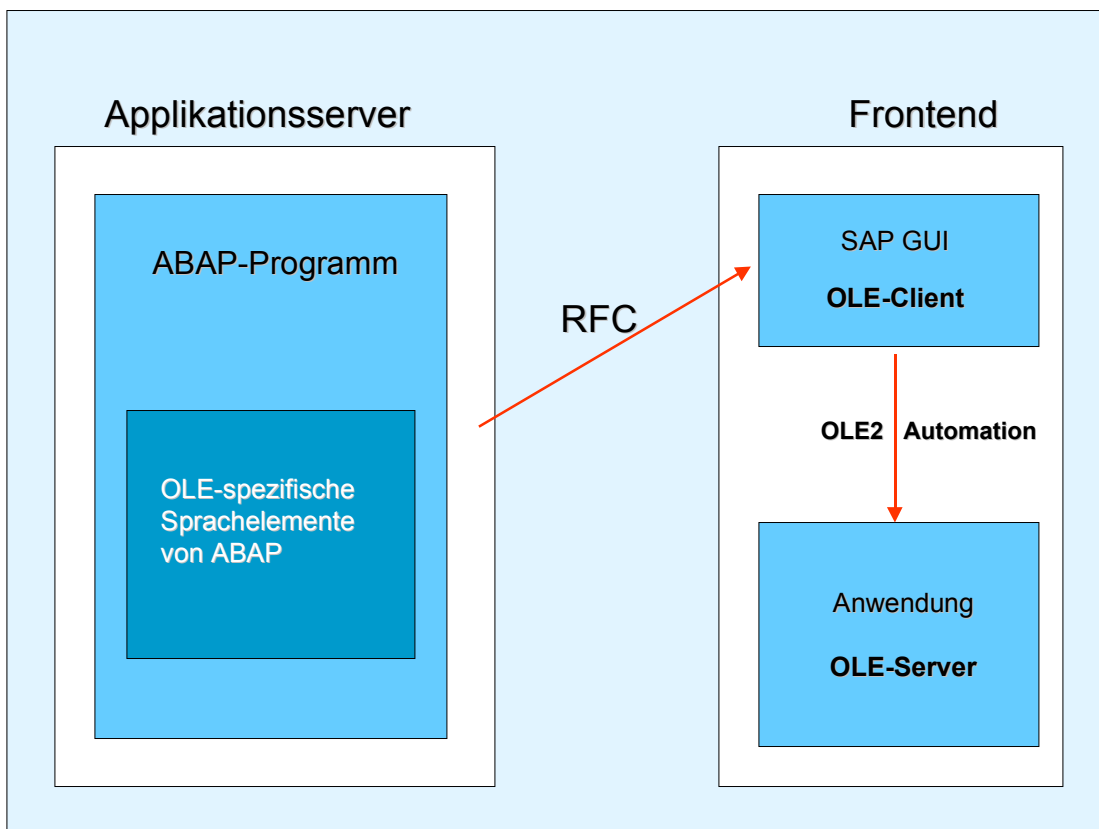
Aufgrund der Open-Object-Schnittstelle unterstützt ABAP die OLE2-Automationstechnik. Desktop-Anwendungen, die ihre Funktionalität in Form eines OLE2-Automation-Servers anbieten (wie z.B. Excel oder WinWord), können daher in R/3 integriert werden.

Alle von ABAP gesteuerten Anwendungen müssen im R/3 registriert sein.

Die folgenden ABAP-Schlüsselwörter steuern Anwendungen:

- CREATE OBJECT
- SET PROPERTY
- GET PROPERTY
- CALL METHOD
- FREE OBJECT

Bei einem Aufruf aus einem ABAP-Programm fungiert der SAPGUI als OLE-Client und die Desktop-Anwendung als der OLE-Server.



Externe Anwendungen in R/3 registrieren

Externe Anwendungen in R/3 registrieren

Alle ABAP-gesteuerten Anwendungen müssen in die Tabelle TOLE in der ABAP Workbench eingetragen werden. Um diese Tabelle zu pflegen, wählen Sie *Entwicklung* → *Programmierumfeld* → *OLE2* → *OLE2 Konfiguration* (Transaktion **SOLE**).

Der Schlüssel der Tabelle TOLE (OLE-Anwendung) wird in der CREATE-Anweisung als Objektklassenname verwendet. Jeder Eintrag enthält alle zur Erzeugung eines OLE-Objektes benötigten Informationen.

Die Tabelle TOLE gibt außerdem an, ob es zu einer Anwendung Typinformationen (TypeInfo) gibt.

Die **TypeInfo** beschreibt alle Objekte, die eine bestimmte Anwendung verarbeiten kann, einschließlich aller Methoden, Attribute und Parameter.

Die Anzeige *Sicht "Pflegeview für OLE Anwendungen" anzeigen: Übersicht* enthält alle Informationen über die OLE-Anwendungsnamen, die Versionsnummern der OLE-Anwendungen und die Klassen-IDs in Zeichenformat.

Wenn Sie spezifischere Informationen benötigen, wählen Sie *Detail*. Es erscheint dann die Anzeige *Sicht "Pflegeview für OLE-Anwendungen" anzeigen: Detail* mit folgenden Feldern:

- OLE-Anwendung
Der Anwendungsname bestimmt die Objektklasse beim Erzeugen eines Objektes.
- Versionsnummer
Wenn mehrere Versionen einer OLE-Anwendung existieren, verwenden Sie die Versionsnummer aus der Registrierungsdatenbank auf dem Präsentationsserver, um zwischen den einzelnen Versionen unterscheiden zu können.
- CLSID
OLE-Klassenidentifikation im Zeichenformat, wie in der Registrierungsdatenbank auf dem Präsentationshost angegeben. Die CLSID-Nummer wird an den Frontend geschickt.
- CLSID TypeLib
OLE-Klassenidentifikation im Zeichenformat. Die Typenbibliothek einer OLE-Anwendung hat teilweise ihre eigene CLSID.
- OLE-Objektnamen
Der Name des ersten erzeugten OLE-Objektes. Dieses Feld wird nur im Object Browser verwendet.
- TypeInfo Key
Schlüssel für die TypeInfo einer OLE-Anwendung. Unter diesem Schlüssel wird die Type-Info in der Datenbank gespeichert. NO_TYPELIB gibt an, daß Sie für diese Anwendung keine TypeInfo verwenden wollen.
- Include Programm
Include-Programm mit Konstantendefinitionen. Das Include-Programm muß manuell gepflegt werden. Gegenwärtig wird dieser Eintrag nur zu Dokumentations- und Verweiszwecken verwendet.

Externe Anwendungen in R/3 registrieren

- Sprache
Die Sprache des folgenden Textes.
- Text
Beschreibung einer OLE-Anwendung.

OLE-Typinformation zur Konvertierung in R/3 laden

OLE-Typinformation zur Konvertierung in R/3 laden

Sie können die Typinformationen vom Präsentationsserver in Tabelle OLELOAD des R/3-Systems laden (in der ABAP Workbench). Wählen Sie dazu *Entwicklung* → *Programmierumfeld* → *OLE2* → *TypeInfo laden* (Transaktion SOLI). Die Anzeige *TypeInfo geladen* mit folgenden Informationen erscheint:

- Anwendung
- Version
- CLSID der Anwendung
- CLSID der Objektbibliothek

Mit *TypeInfo laden* wird ein Teil der TypeLib in die SAP-Datenbank geladen.



Dies ist nur dann möglich, wenn die betreffende Anwendung auf Ihren PC geladen ist.

Der ABAP-Prozessor kann dann die nötigen Typkonvertierungen durchführen und ist damit unabhängig von der Sprache der OLE-Anwendung. D.h. ein ABAP-Programm kann Methoden- und Attributenamen in jeder Sprache verwenden, für die Typinformationen geladen sind, und kann gleichzeitig immer noch Anwendungen verarbeiten, die in einer anderen Sprache installiert wurden.

Für alle von SAP-Standardanwendungen verwendeten OLE-Anwendungen wird die englische Version in der Tabelle OLELOAD ausgeliefert.

Auch wenn ein OLE-Server keine TypeInfo hat, kann er von ABAP verarbeitet werden. In diesem Fall müssen jedoch die Namen der Methoden und Attribute in derselben Sprache sein, in der auch die Frontend-Anwendung installiert wurde.

Konvertierungsregeln

Bei der Parameterübergabe erfolgt die Konvertierung immer über den Zeichentyp:

Der ABAP-Typ wird nach den ABAP-Konventionen in CHAR konvertiert. Danach wird CHAR unter Verwendung der entsprechenden OLE-Typen nach OLE konvertiert:

ABAP-Typ → CHAR → *OLE-Typ*

Ohne TypeInfo, oder wenn der OLE-Typ nicht definiert ist, wird der OLE-Typ je nach ABAP-Typ wie folgt gesetzt:

ABAP-Typen	OLE-Typen
I	Int
P,F	Double
D	Datum
T	Zeit
Andere	Zeichen

Gegenwärtig ist die Länge von Zeichenvariablen auf maximal 255 begrenzt.



Bei einigen Objekten hängt die Semantik der Parameter vom OLE-Typ ab (s. Dokumentation der Server-Anwendung).

Arbeiten mit dem OLE-Object Browser

Arbeiten mit dem OLE-Object Browser

Der Object Browser liefert eine Liste der OLE-Anwendungen mit geladenen Typinfos in Englisch. Diese Informationen werden in Tabelle INDX gespeichert.

Wählen Sie *Entwicklung* → *Programmierumfeld* → *OLE2* → *Object Browser* (Transaktion SOLO).

Sie können eine OLE-Anwendung auch im Dialog mit dem Object Browser anlegen. Dann können Sie die Methoden aufrufen oder die Attribute lesen und schreiben. Die Ergebnisse jedes Aufrufs werden direkt angezeigt.



Diese Informationen müssen vom Kunden geladen werden. SAP liefert keine Initialeinträge aus.

Implementierung

Die Kommunikation zwischen den Anwendungsservern und dem Frontend erfolgt über die RFC-Schnittstelle. Der SAPGUI enthält besondere OLE-Funktionen, die Sie aus einem ABAP-Programm mit normalen RFC-Aufrufen ansprechen können (Destination 'SAPGUI').

Die Implementierung erfolgt über die ABAP-Schlüsselwörter CREATE OBJECT und FREE OBJECT (die die Funktionsbausteine CREATE_OBJECT und FREE_OBJECT aufrufen) und über C-Funktionen, die als RFC-Server-Programme geschrieben sind.

Bei [CREATE OBJECT \[Seite 18\]](#) liest der ABAP-Prozessor den entsprechenden Eintrag in Tabelle TOLE und verwendet die gefundene OLE-Klassen-ID zum Aufruf des Funktionsbausteins OLE_CREATE_OBJECT auf dem Frontend über RFC. Dann werden die zur Erzeugung eines OLE-Objektes benötigten Funktionsbausteine aus der Windows OLE-Bibliothek aufgerufen und ein Objekt-Handle wird an den ABAP-Prozessor zurückgegeben. Aus OLE-Sicht stellt der SAPGUI den eigentlichen OLE-Client dar.

Die Anweisungen [CALL METHOD \[Seite 24\]](#), [GET PROPERTY \[Seite 22\]](#) und [SET PROPERTY \[Seite 20\]](#) werden auf dem Anwendungsserver gebündelt und in Form einer internen Tabelle an den SAPGUI-Funktionsbaustein OLE_FLUSH_CALL übergeben.

[FREE OBJECT \[Seite 26\]](#) ruft den SAPGUI-Funktionsbaustein OLE_RELEASE_OBJECT auf, der neben dem OLE-Aufruf zusätzlich alle für dieses Objekt belegten Ressourcen auf dem Frontend freigibt.

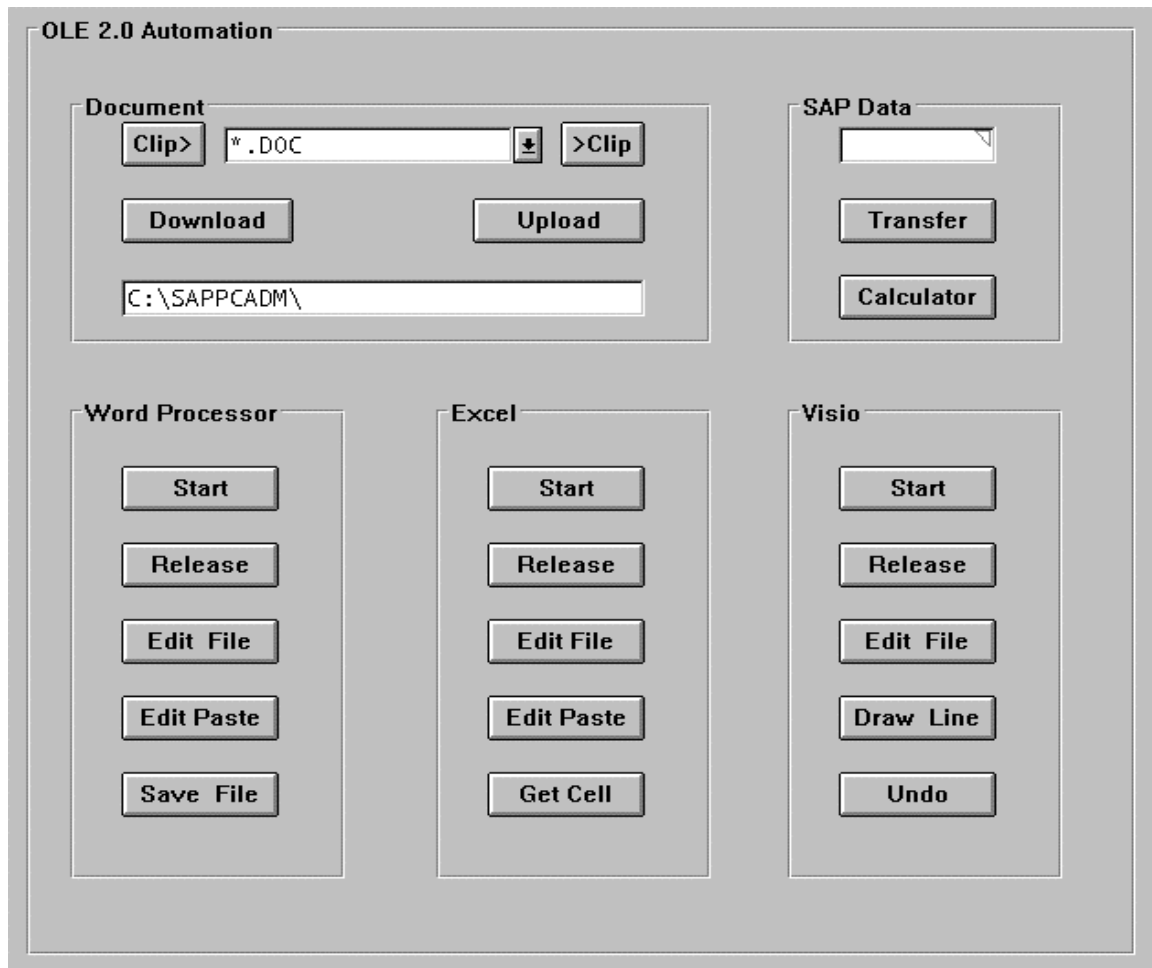
Auch das Lesen der TypeInfo ist über einen RFC-Aufruf an den SAPGUI implementiert.

Beispiele

Beispiele

Beispiele und OLE-Demonstrationen sind in WinWord, Excel und Visio verfügbar. Wählen Sie *Entwicklung* → *Programmierumfeld* → *OLE2* → *OLE-Demo* (Transaktion OLE) in der ABAP Workbench.

Mit Transaktion OLE können Sie drei OLE-Server aktivieren und über die entsprechenden Drucktasten zahlreiche Methoden und Attribute testen:



Ein einfaches Programmbeispiel finden Sie auch im [Tutorial zu Kommunikationsschnittstellen. \[Extern\]](#)

R/3 als OLE-Automation-Server

Im Gegensatz zu ABAP als OLE-Automation-Controller bietet R/3 auch Teile seiner Funktionalität als Automation-Server an. Dies bedeutet, daß alle über RFC aufrufbaren Funktionsbausteine von jedem beliebigen OLE-Automation-Controller aufgerufen werden können. Dazu muß das Server-Programm RFCSRV.EXE auf dem Frontend installiert sein.



Wenn Sie diese Thematik anhand eines konkreten Programmbeispiels kennenlernen möchten, dann lesen die entsprechende Lektion im [Tutorial zu Kommunikationsschnittstellen](#). [Extern]

Verwandte ABAP-Schlüsselwörter

Verwandte ABAP-Schlüsselwörter

[Einführung \[Seite 17\]](#)

[CREATE OBJECT \[Seite 18\]](#)

[SET PROPERTY \[Seite 20\]](#)

[GET PROPERTY \[Seite 22\]](#)

[CALL METHOD \[Seite 24\]](#)

[FREE OBJECT \[Seite 26\]](#)

Einführung

Seit R/3-Release 3.0 wurde der Befehlssatz des ABAP-Interpreters erweitert. Er umfaßt jetzt auch Schlüsselwörter, die es dem Anwendungsprogrammierer erlauben, externe Objekte zu verarbeiten. Als erstes Objektmodell wurde dabei OLE2 unterstützt.

Mit den ABAP-Schlüsselwörtern können Sie alle Anwendungen aus einem ABAP-Programm heraus steuern, deren Funktionalität in Form eines OLE2 Automation Server vorliegt:

Beispiele für solche Anwendungen sind die Microsoft-Produkte EXCEL oder WinWord.



Weitere Informationen zu OLE2 finden Sie unter **OLE2** in der Online-Hilfe des ABAP-Editors.

CREATE OBJECT**CREATE OBJECT**

Das ABAP-Schlüsselwort CREATE OBJECT erzeugt ein Objekt der Klasse "obj class".

Grundform: CREATE OBJECT obj class.

Zusatz: LANGUAGE langu

Um einen OLE-Automation-Server (z.B. EXCEL) aus ABAP heraus anzusprechen, muß der Server bei SAP registriert sein. Mit Transaktion SOLE (s. [Externe Anwendungen in R/3 registrieren \[Extern\]](#)) können Sie einen Automation-Server einer Klasse zuordnen.

Die Anweisung CREATE erzeugt das Ursprungsobjekt dieser Klasse, das dann mit den verwandten Schlüsselwörtern weiterverarbeitet werden kann. Der Rückgabewert SY-SUBRC gibt das Ergebnis des Erzeugungsvorgangs an. Er kann folgende Werte annehmen:

- SY-SUBRC = 0:
Objekt erfolgreich erzeugt.
- SY-SUBRC = 1:
SAPGUI-Kommunikationsfehler.
- SY-SUBRC = 2:
Fehler beim SAPGUI-Funktionsaufruf. Die Frontend-Ports der OLE-Implementierungsbausteine von SAP sind nur unter Windows und Apple Macintosh implementiert.
- SY-SUBRC = 3:
Der OLE-API-Aufruf erzeugte einen Fehler - möglicherweise ein Speicherplatzproblem.
- SY-SUBRC = 4:
Das Objekt ist bei SAP nicht registriert.

Der Zusatz LANGUAGE legt die für die Methoden- und Attributenamen der Objektklasse gewählte Sprache fest. Wenn Sie keine Sprache angeben, wird Englisch als Standard verwendet.

CREATE OBJECT gehört zu einer Gruppe von Schlüsselwörtern, mit denen Sie externe Objekte in ABAP verarbeiten können. Gegenwärtig wird nur das Objektmodell OLE2 unterstützt. Alle zu verarbeitenden Objekte müssen daher vom Typ OLE2_OBJECT sein. Dieser Typ sowie andere erforderliche Daten sind im Include-Baustein OLE2INCL definiert.



So erzeugen Sie ein EXCEL-Objekt.

```
INCLUDE OLE2INCL.  
DATA EXCEL TYPE OLE2_OBJECT.  
CREATE OBJECT EXCEL 'Excel.Application'.
```

Siehe auch:

[SET PROPERTY \[Seite 20\]](#), [GET PROPERTY \[Seite 22\]](#), [CALL METHOD \[Seite 24\]](#) und [FREE OBJECT \[Seite 26\]](#).

SET PROPERTY**SET PROPERTY**

Das ABAP-Schlüsselwort SET PROPERTY setzt das Attribut *p* des Objekts *obj* abhängig vom Inhalt von Feld *f*. Das Objekt *obj* muß vom Typ OLE2_OBJECT sein.

Grundform: SET PROPERTY OF obj p = f.

Zusatz: NO FLUSH

In der Regel puffert der ABAP-Prozessor alle aufeinanderfolgenden OLE-Anweisungen und schickt sie gebündelt an den Präsentationsserver. Dennoch kann eine Anweisung auch auf die Ergebnisse vorhergehender Anweisungen zugreifen.

Wenn Sie im Debugging-Modus arbeiten, sollten Sie allerdings nicht vergessen, daß die Werte der Rückgabeparameter erst unmittelbar vor der Ausführung der ersten nicht-OLE ABAP-Anweisung angezeigt werden können. Eine Anweisung, die sich auf ein noch nicht durch eine OLE-Anweisung erzeugtes Objekt bezieht, beendet die automatische Bündelung.

Der Rückgabewert SY-SUBRC gibt an, ob alle gebündelten Anweisungen erfolgreich ausgeführt wurden. Er kann folgende Werte annehmen:

- SY-SUBRC = 0:
Alle Anweisungen wurden erfolgreich ausgeführt.
- SY-SUBRC = 1:
Bei der Kommunikation mit dem Präsentationsserver trat ein Systemfehler auf. Das Feld SY-MSGLI enthält eine kurze Beschreibung des Fehlers.
- SY-SUBRC = 2:
Ein Methodenaufruf verursachte einen Fehler.
- SY-SUBRC = 3:
Das Setzen eines Attributs verursachte einen Fehler.
- SY-SUBRC = 4:
Das Lesen eines Attributs verursachte einen Fehler.
In den letzten drei Fällen erscheint auf dem Präsentationsserver ein Dialogfenster mit einem Fehlerhinweis.

Der Zusatz NO FLUSH setzt den Sammelvorgang fort, selbst wenn die nächste Anweisung keine OLE-Anweisung ist. So können Sie z.B. in einer Schleife eine Reihe von Attributen setzen und in einer einzigen Transportoperation auf den Präsentationsserver herunterladen.

Wenn Sie NO FLUSH verwenden, müssen Sie als Programmierer sicherstellen, daß Sie sich nicht auf den Inhalt der Rückgabeparameter verlassen, die ja noch nicht gefüllt sind. Auch müssen alle Objekte als Bündel initialisiert werden, d.h. sie müssen durch einen schon ausgeführten OLE-Aufruf bereits erzeugt worden sein.

Jede FREE-Anweisung führt immer zu einem Herunterladen des Puffers.

SET PROPERTY gehört zu einer Gruppe von Schlüsselwörtern, mit denen Sie externe Objekte in ABAP verarbeiten können. Gegenwärtig wird nur das Objektmodell OLE2 unterstützt. Alle zu verarbeitenden Objekte müssen daher vom Typ OLE2_OBJECT sein. Dieser Typ sowie andere erforderliche Daten sind im Include-Baustein OLE2INCL definiert.

SET PROPERTY



So setzen Sie das Attribut 'Visible' (sichtbar) eines EXCEL-Arbeitsblatts.

```
INCLUDE OLE2INCL.  
DATA EXCEL TYPE OLE2_OBJECT.  
CREATE OBJECT EXCEL 'Excel.Application'.  
SET PROPERTY OF EXCEL 'Visible' = 1.
```

Siehe auch:

[CREATE OBJECT \[Seite 18\]](#), [GET PROPERTY \[Seite 22\]](#), [CALL METHOD \[Seite 24\]](#) und [FREE OBJECT \[Seite 26\]](#).

GET PROPERTY

GET PROPERTY

Das ABAP-Schlüsselwort GET PROPERTY kopiert das Attribut *p* des Objekts *obj* in das Feld *f*. Das Objekt *obj* muß vom Typ OLE2_OBJECT sein.

Grundform: GET PROPERTY OF obj p = f.

Zusatz: NO FLUSH

In der Regel puffert der ABAP-Prozessor alle aufeinanderfolgenden OLE-Anweisungen und schickt sie gebündelt an den Präsentationsserver. Dennoch kann eine Anweisung auch auf die Ergebnisse vorhergehender Anweisungen zugreifen.

Wenn Sie im Debugging-Modus arbeiten, sollten Sie allerdings nicht vergessen, daß die Werte der Rückgabeparameter erst unmittelbar vor der Ausführung der ersten nicht-OLE ABAP-Anweisung angezeigt werden können. Eine Anweisung, die sich auf ein noch nicht durch eine OLE-Anweisung erzeugtes Objekt bezieht, beendet die automatische Bündelung.

Der Rückgabewert SY-SUBRC gibt an, ob alle gebündelten Anweisungen erfolgreich ausgeführt wurden. Er kann folgende Werte annehmen:

- SY-SUBRC = 0:
Alle Anweisungen wurden erfolgreich ausgeführt.
- SY-SUBRC = 1:
Bei der Kommunikation mit dem Präsentationsserver trat ein Systemfehler auf. Das Feld SY-MSGLI enthält eine kurze Beschreibung des Fehlers.
- SY-SUBRC = 2:
Ein Methodenaufruf verursachte einen Fehler.
- SY-SUBRC = 3:
Das Setzen eines Attributs verursachte einen Fehler.
- SY-SUBRC = 4:
Das Lesen eines Attributs verursachte einen Fehler.
In den letzten drei Fällen erscheint auf dem Präsentationsserver ein Dialogfenster mit einem Fehlerhinweis.

GET PROPERTY gehört zu einer Gruppe von Schlüsselwörtern, mit denen Sie externe Objekte in ABAP verarbeiten können. Gegenwärtig wird nur das Objektmodell OLE2 unterstützt. Alle zu verarbeitenden Objekte müssen daher vom Typ OLE2_OBJECT sein. Dieser Typ sowie andere erforderliche Daten sind im Include-Baustein OLE2INCL definiert.

Der Zusatz NO FLUSH setzt den Sammelvorgang fort, selbst wenn die nächste Anweisung keine OLE-Anweisung ist. So können Sie z.B. in einer Schleife eine Reihe von Attributen setzen und in einer einzigen Transportoperation auf den Präsentationsserver herunterladen.

Wenn Sie NO FLUSH verwenden, müssen Sie als Programmierer sicherstellen, daß Sie sich nicht auf den Inhalt der Rückgabeparameter verlassen, die ja noch nicht gefüllt sind. Auch müssen alle Objekte als Bündel initialisiert werden, d.h. sie müssen durch einen schon ausgeführten OLE-Aufruf bereits erzeugt worden sein.

Jede FREE-Anweisung führt immer zu einem Herunterladen des Puffers.

GET PROPERTY



So lesen Sie das Attribut 'Visible' (sichtbar) eines EXCEL-Arbeitsblatts:

```
INCLUDE OLE2INCL.  
DATA: EXCEL TYPE OLE2_OBJECT.  
      VISIBLE TYPE I.  
CREATE OBJECT EXCEL 'Excel.Application'.  
GET PROPERTY OF EXCEL 'Visible' = VISIBLE.
```

Siehe auch:

[SET PROPERTY \[Seite 20\]](#), [CREATE OBJECT \[Seite 18\]](#), [CALL METHOD \[Seite 24\]](#) und [FREE OBJECT \[Seite 26\]](#).

CALL METHOD**CALL METHOD**

Das ABAP-Schlüsselwort CALL METHOD ruft die Methode *m* des Objekts *obj* auf. Dabei kann *m* ein Literal oder eine Variable sein.

Grundform: CALL METHOD OF obj m.

Zusätze:

1. ... = f
2. ... EXPORTING p1 = f1... pn = fn
3. NO FLUSH.

In der Regel puffert der ABAP-Prozessor alle aufeinanderfolgenden OLE-Anweisungen und schickt sie gebündelt an den Präsentationsserver. Dennoch kann eine Anweisung auch auf die Ergebnisse vorhergehender Anweisungen zugreifen.

Wenn Sie im Debugging-Modus arbeiten, sollten Sie allerdings nicht vergessen, daß die Werte der Rückgabeparameter erst unmittelbar vor der Ausführung der ersten nicht-OLE ABAP-Anweisung angezeigt werden können. Eine Anweisung, die sich auf ein noch nicht durch eine OLE-Anweisung erzeugtes Objekt bezieht, beendet die automatische Bündelung.

Der Rückgabewert SY-SUBRC gibt an, ob alle gebündelten Anweisungen erfolgreich ausgeführt wurden. Er kann folgende Werte annehmen:

- SY-SUBRC = 0:
Alle Anweisungen wurden erfolgreich ausgeführt.
- SY-SUBRC = 1:
Bei der Kommunikation mit dem Präsentationsserver trat ein Systemfehler auf. Das Feld SY-MSGLI enthält eine kurze Beschreibung des Fehlers.
- SY-SUBRC = 2:
Ein Methodenaufruf verursachte einen Fehler.
- SY-SUBRC = 3:
Das Setzen eines Attributs verursachte einen Fehler.
- SY-SUBRC = 4:
Das Lesen eines Attributs verursachte einen Fehler.
In den letzten drei Fällen erscheint auf dem Präsentationsserver ein Dialogfenster mit einem Fehlerhinweis.

CALL METHOD gehört zu einer Gruppe von Schlüsselwörtern, mit denen Sie externe Objekte in ABAP verarbeiten können. Gegenwärtig wird nur das Objektmodell OLE2 unterstützt. Alle zu verarbeitenden Objekte müssen daher vom Typ OLE2_OBJECT sein. Dieser Typ sowie andere erforderliche Daten sind im Include-Baustein OLE2INCL definiert.

Zusätze

- **Zusatz 1** ... = f

CALL METHOD

Der Zusatz =f speichert den Rückgabewert der Methode in der Variablen f. Der Rückgabewert kann auch vom Typ OLE2_OBJECT sein. Dieser Zusatz muß immer an erster Stelle vor anderen Zusätzen stehen.

- **Zusatz 2** ... EXPORTING p1 = f1... pn = fn

Der Zusatz EXPORTING übergibt Werte von Feldern an die Parameter der Methode. p1, p2,... sind entweder Schlüsselwortparameter oder Positionsparameter.

Wenn die Parameter der Reihe nach zugeordnet werden, müssen p1, p2,... mit einem "#" beginnen, gefolgt von der Positionsnummer des Parameters. Gegenwärtig werden nur Positionsparameter unterstützt.

Die EXPORTING-Parameter stehen immer am Ende der Anweisung.

- **Zusatz 3** ... NO FLUSH.

Der Zusatz NO FLUSH setzt den Sammelvorgang fort, selbst wenn die nächste Anweisung keine OLE-Anweisung ist. So können Sie z.B. in einer Schleife eine Reihe von Attributen setzen und in einer einzigen Transportoperation auf den Präsentationsserver herunterladen.

Wenn Sie NO FLUSH verwenden, müssen Sie als Programmierer sicherstellen, daß Sie sich nicht auf den Inhalt der Rückgabeparameter verlassen, die ja noch nicht gefüllt sind. Auch müssen alle Objekte als Bündel initialisiert werden, d.h. sie müssen durch einen schon ausgeführten OLE-Aufruf bereits erzeugt worden sein.

Jede FREE-Anweisung führt immer zu einem Herunterladen des Puffers.



So öffnen Sie eine EXCEL-Datei mit der Methode 'Open':

```
INCLUDE OLE2INCL.
DATA EXCEL      TYPE OLE2_OBJECT.
DATA WORKBOOK  TYPE OLE2_OBJECT.

CREATE OBJECT EXCEL 'Excel.Application'.
CALL METHOD OF EXCEL 'WORKBOOKS' = WORKBOOK.

CALL METHOD OF WORKBOOK 'Open' EXPORTING #1 = 'C:\EX1.XLS'.
```

Siehe auch:

[SET PROPERTY \[Seite 20\]](#), [GET PROPERTY \[Seite 22\]](#), [CREATE OBJECT \[Seite 18\]](#) und [FREE OBJECT \[Seite 26\]](#).

FREE OBJECT

FREE OBJECT

Das ABAP-Schlüsselwort FREE OBJECT gibt den für das Objekt *obj* benötigten Speicherplatz frei. Das Objekt kann dann nicht weiter verarbeitet werden.

Der Rückgabewert *SY-SUBRC* gibt an, ob das gewünschte Objekt freigegeben wurde oder nicht.

- SY-SUBRC = 0:
Objekt erfolgreich freigegeben.
- SY-SUBRC = 1:
Fehler in der SAPGUI-Kommunikation.
- SY-SUBRC = 2:
Fehler beim SAPGUI-Funktionsaufruf.

Die OLE-Funktionsbausteine sind nur unter Windows implementiert.

FREE OBJECT gehört zu einer Gruppe von Schlüsselwörtern, mit denen Sie externe Objekte in ABAP verarbeiten können. Gegenwärtig wird nur das Objektmodell OLE2 unterstützt. Alle zu verarbeitenden Objekte müssen daher vom Typ OLE2_OBJECT sein. Dieser Typ sowie andere erforderliche Daten sind im Include-Baustein OLE2INCL definiert.



So geben Sie ein EXCEL-Objekt frei:

```
INCLUDE OLE2INCL.  
DATA: EXCEL TYPE OLE2_OBJECT.  
CREATE OBJECT EXCEL 'Excel.Application'.  
FREE OBJECT EXCEL.
```

Siehe auch:

[SET PROPERTY \[Seite 20\]](#), [GET PROPERTY \[Seite 22\]](#), [CALL METHOD \[Seite 24\]](#) und [CREATE OBJECT \[Seite 18\]](#).