

SAP DataProvider



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

SAP DataProvider	5
SAP DataProvider	6
Funktionsbeschreibung	8
Funktionsbausteine	14
DataProvider-Objekt	16
Technische Referenz	21

SAP DataProvider

SAP DataProvider

Einsatzmöglichkeiten

Der SAP DataProvider (in dieser Dokumentation kurz DataProvider genannt) ist eine Standardschnittstelle zur Datenbeschaffung, die von Controls im SAP GUI verwendet wird.

Der DataProvider

- verwaltet die Daten
- bildet die Daten in Standard-Windows-Datentypen und -Datenobjekten ab
- steuert die Datenkonvertierung

Der DataProvider besteht aus folgenden Komponenten:

- ABAP-Funktionsbausteine

ABAP-Funktionsbausteine bilden die Schnittstelle des DataProvider zum Anwendungsprogrammierer bzw. Schalenprogrammierer. Sie arbeiten mit Datentypen, die im ABAP üblich sind, z.B. internen Tabellen.
- Funktionsaufrufe im SAP GUI

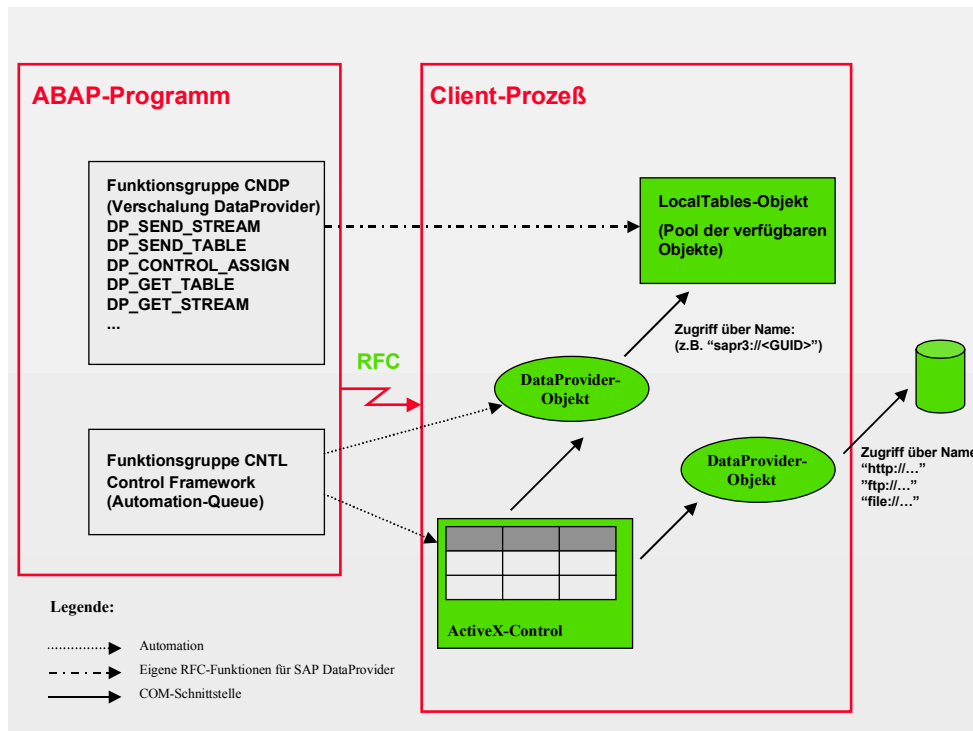
Im SAP GUI existieren zwei Funktionsaufrufe, die über RFC erreichbar sind. Die eine Funktion sendet jeweils eine Tabelle an das SAP GUI, die andere holt jeweils eine Tabelle vom SAP GUI. Die mit diesem Aufruf übergebene Tabelle ist ein Datenobjekt.
- LocalTables-Objekt

Das LocalTables-Objekt verwaltet alle Tabellen in einem Datenpool, der innerhalb eines Prozeßraums global ist.
- DataProvider-Objekt

Das DataProvider-Objekt ermöglicht den Zugriff auf die Datenobjekte im LocalTables-Objekt. Sie können Referenzen auf Daten im LocalTables-Objekt anfordern oder dort ablegen.

Da das DataProvider-Objekt mit Datentypen und Datenobjekten arbeitet, die als Windows-Standards gelten, benötigt der Benutzer keine Kenntnisse über interne Tabellen oder ABAP-Datentypen.

Zusätzlich bietet das DataProvider-Objekt Zugriff auf ftp:- und http:-Server sowie auf Dateien, die auf lokalen oder auf File-Servern gespeichert sind.



Der DataProvider wird in den meisten ActiveX-Controls verwendet, die von SAP zur Verfügung gestellt werden. Beispiele für solche Controls sind Text Edit Control, HTML Control, Image Control, Tree Control und Desktop OfficeIntegration.

Einführungshinweise

Sie benötigen den DataProvider aus folgenden Gründen:

- Mit dem Standardmechanismus der Automation Queue ist es nicht möglich, Datenvolumen über 256 Byte mit einem Methodenaufruf zu übergeben oder BLOB-Daten (Binary Large Object) wie Office-Dokumente, Bilder oder Editor-Inhalte zu übertragen.
- Externe Datenquellen (d.h. Quellen, die außerhalb des R/3-Systems oder des aktuellen ABAP-Programms liegen) müssen ebenfalls zugänglich gemacht werden. Beispiele für solche externen Quellen sind das Info-Repository sowie lokale Dateien.

Einschränkungen

Der SAP DataProvider ist nur für Windows-Plattformen (Windows95 oder Windows NT ab 4.0) verfügbar.

Funktionsbeschreibung

Funktionsbeschreibung

Der SAP DataProvider arbeitet nach dem Push-Modell, d.h. bevor ein Control eine Referenz auf ein Datenobjekt erhalten kann, werden die Daten immer vom Server an den Client als interne Tabelle (über RFC) übertragen.

Jedes Datenobjekt hat einen eindeutigen Namen, der in der Regel die Syntax

```
sapr3://<GUID>
```

aufweist und als URL bezeichnet wird. Um eine Referenz auf ein Datenobjekt zu erhalten, müssen Sie dessen URL angeben.

Die Daten werden auf der Client-Seite häufig als interne Tabellen dargestellt, aber es gibt auch typische Schnittstellen für Massendaten. Diese Formen der Darstellung werden als **Medien** bezeichnet. Beim Zugriff auf die Daten geben Sie das gewünschte Medium an. Die Daten werden dann entsprechend dargestellt und bei Bedarf konvertiert.

Jedes Datenobjekt auf Client-Seite wird durch die Attribute *Type* und *SubType* beschrieben, die mit den im Internet benutzten MIME-Typen eng verwandt sind und als **Format** bezeichnet werden. Konvertierungen zwischen unterschiedlichen Medien werden über das Format des Datenobjekts gesteuert, da verschiedene Konvertierungen für verschiedene Formate möglich sind. Sie können außerdem Datenobjekte eines Formats in Datenobjekte eines anderen Formats konvertieren. Solche Konvertierungen werden über das Quell- und das Zielformat gesteuert.

Für den Datenrücktransport vom Client an den Server gilt das gleiche Prinzip. Sie können Daten über jedes beliebige Medium in das DataProvider-Objekt stellen. Falls keine URL vorhanden ist, müssen Sie dem Datenobjekt über den SAP DataProvider eine URL zuordnen. Anschließend können Sie das Datenobjekt unter Angabe dieser URL als interne Tabelle an den Anwendungsserver übertragen.

Um den Zugriff auf Nicht-R/3-Datenquellen zu ermöglichen, kann der SAP DataProvider auch URLs in den Namensräumen http:, ftp: und file: verarbeiten. Die Vorgehensweise für diese Namensräume ist mit der oben beschriebenen identisch, mit der Ausnahme, daß die Daten vom Anwendungsserver an den Client nicht über ein Push-Modell übergeben, sondern in einem Pull-Modell vom entsprechenden Server geladen werden.

Verwendung von MIME-Typen

Zur Beschreibung von BLOB-Daten (Binary Large Object) werden MIME-Typen verwendet. Über MIME-Typen können Sie die Konvertierung unterschiedlicher Typen von BLOB-Daten steuern und Zugriff auf die Datenobjekte selbst erhalten.

Durch Angabe eines MIME-Typs oder eines Teils eines MIME-Typs vor dem Zugriff auf einen Server können Sie sicherstellen, daß eine spätere Datenübergabe nur dann angestoßen wird, wenn Daten des entsprechenden Typs vorhanden sind.

Liegen Daten in mehreren verschiedenen Formaten vor, können Sie über die Angabe des MIME-Typs bestimmen, welches Format verwendet werden soll. Dabei gelten folgende Regeln:

Angabe des MIME-Typs	Verarbeitung
Typ und Subtyp werden angegeben	Nur Daten dieses MIME-Typs werden verarbeitet.

Funktionsbeschreibung

Typ oder Subtyp werden angegeben	Es werden nur Daten verarbeitet, die dem angegebenen Teil entsprechen. Der andere Teil wird ignoriert oder als Maskenzeichen interpretiert.
Weder Typ noch Subtyp werden angegeben	Alle Daten werden verarbeitet. Dies kann zu unnötigen Server-Zugriffen und damit zu schlechterer Performance führen.

Für BLOB-Daten müssen Sie immer einen MIME-Typ angeben. Anderenfalls werden die Daten nicht angenommen, und es tritt ein Fehler auf.

- Wenn für ein Datenobjekt kein passender Standard-MIME-Typ vorhanden ist, können Sie einen beliebigen Namen verwenden:
 - Der Haupttyp sollte immer **application** genannt werden.
 - Der Subtyp sollte immer mit **x-** beginnen (zum Beispiel `application/x-myspecialtype`).
- Wenn der MIME-Typ nicht relevant ist, sollten Sie **application/x-unknown** verwenden.
 - Wenn Sie Daten eines eigenen MIME-Typs, der kein Standard ist, von/zu einem http:-Server transportieren möchten, wählen Sie eine der beiden folgenden Möglichkeiten:
 - Benutzen Sie den Standard-MIME-Typ **application/octet-stream**.
 - Registrieren Sie Ihren eigenen MIME-Typ auf dem http:-Server.



Wenn BLOB-Daten als interne Tabelle an den Client transportiert werden, sollten Sie die Größe der BLOB-Daten angeben. Anderenfalls wird als Größe die Anzahl der Zeilen multipliziert mit der Tabellenbreite angenommen.

Typisierte Tabellen

Wenn Sie typisierte Tabellen transportieren möchten, müssen Sie die Struktur einer Tabelle auf dem Client kennen.

Als MIME-Typ verwenden Sie immer **application/x-r3table**. Zusätzlich wird eine Tabelle des Typs RFC_FIELDS übertragen, welche die eigentliche Datentabelle beschreibt. Für diesen Datenobjekttyp gibt es ein Konvertierungsobjekt, das den Erfordernissen typisierter Tabellen entspricht.

Für den Transport typisierter Tabellen stehen Funktionsbausteine in der Funktionsgruppe CNDP zur Verfügung.



- Beim Transport typisierter Tabellen vom Client an ein ABAP-Programm sollten Sie zuerst eine leere Tabelle desselben Typs an den Client transportieren. Dann kann das Programm/Control auf dem Client diese Tabelle füllen, und Sie können die Tabelle anschließend zu einem beliebigen Zeitpunkt zurück an den Server transportieren.
- Es ist nicht möglich, typisierte interne Tabellen von oder zu http:-, ftp:- oder file:-Servern zu transportieren, da dafür ein bekanntes und sicheres Protokoll für die Serialisierung interner

Funktionsbeschreibung

Tabellen erforderlich wäre. Es ist sinnvoller, auf OLE-DB und Active Data Object (ADO) auszuweichen. Das dafür notwendige Formatkonvertierungsobjekt ist unter dem MIME-Typ **application/x-rowset** registriert.

- Es gibt kein Deltamanagement im RFC zwischen dem SAP GUI und dem Anwendungsserver. Aus diesem Grund sollten Sie keine Tabellen an den Client schicken, modifizieren und dann wieder vom Anwendungsserver anfordern. Sie können versuchen, die Modifikation in einer eigenen Tabelle zu speichern. Bei BLOB-Daten ist dies jedoch nicht sinnvoll, da die Modifikation von BLOB-Daten in ABAP-Programmen sehr umständlich ist.

Datenübergabe vom ABAP-Programm an den Client

1. Das ABAP-Programm ruft CONTROL_CREATE oder einen anderen ähnlichen Baustein auf, um ein Control zur Handhabung der BLOB-Daten zu erzeugen.
2. Das ABAP-Programm liest die BLOB-Daten in eine interne Tabelle.
3. Das ABAP-Programm ruft einen Funktionsbaustein auf, z.B. DP_SEND_STREAM. Dieser Funktionsbaustein enthält Parameter für die interne Tabelle mit den Daten, die Größe der BLOB-Daten in Byte und den MIME-Typ der BLOB-Daten.
 - a) Der aufgerufene Funktionsbaustein stellt Systeminformationen bereit und ruft DP_CREATE_URL auf.
 - b) DP_CREATE_URL erzeugt einen eindeutigen Namen (sapr3://<GUID>) für das aktuelle Datenobjekt.
 - c) DP_PUT_CLIENT_TABLE wird im SAP GUI (SAPAWRFC.DLL) aufgerufen. An diese Funktion werden die interne Tabelle, die Größe und der MIME-Typ der Daten, die URL sowie die Systeminformationen übergeben.
 - i) DP_PUT_CLIENT_TABLE koppelt die interne Tabelle an eine ISAPrfclTab-Schnittstelle an.
 - ii) Eine Referenz auf die ISAPrfclTab-Schnittstelle, den MIME-Typ, die Größe der BLOB-Daten und weitere Systeminformationen werden im globalen Objekt LocalTables zusammen mit der URL gespeichert. Zusätzlich wird noch eine Kennung für den aktuellen Kontext gespeichert.
 - iii) DP_CREATE_URL liefert als Ergebnis den Namen (URL), unter dem eine Referenz auf die interne Tabelle in LocalTables gespeichert wurde.
4. Das unter Punkt 1 erzeugte Control kennt und generiert ein DataProvider-Objekt. Falls nicht, siehe Punkt 5.
 - a) Die URL des Datenobjekts wird dem Control über CONTROL_SET_PROPERTY oder CONTROL_CALL_METHOD mitgeteilt.
 - i) Das Control erzeugt ein DataProvider-Objekt.
 - ii) Im DataProvider-Objekt ruft das Control (zum Beispiel) die Methode SetDataFromUrl mit der URL als Parameter auf.

Das DataProvider-Objekt erzeugt eine Referenz auf das globale LocalTables-Objekt.

Die URL wird an das LocalTables-Objekt übergeben. Dadurch erhält das DataProvider-Objekt eine Referenz auf die ISAPrfclTab-Schnittstelle.
 - b) Das Control liest die Daten über eine Eigenschaft des DataProvider-Objekts.

Funktionsbeschreibung

- i) Das DataProvider-Objekt liefert die Daten entsprechend dem Zugriffsmedium, das vom Control gefordert wird. Der Zugriff auf die Daten erfolgt nicht zwingend über die ISAPrflTab-Schnittstelle. Er kann über alle Standard-COM-Schnittstellen (IStream, ILockBytes, IUnknown, VARIANT,...) laufen.
 - c) Siehe Punkt 8.
5. Das ABAP-Programm ruft den Funktionsbaustein DP_CREATE auf, der auf dem Client ein DataProvider-Objekt erzeugt.
6. Das ABAP-Programm ruft die Methode SetDataFromUrl des DataProvider-Objekts auf und übergibt die URL als wesentlichen Parameter.
 - a) Das DataProvider-Objekt erzeugt eine Referenz auf das globale LocalTables-Objekt.
 - b) Die URL wird an das LocalTables-Objekt übergeben. Dadurch erhält das DataProvider-Objekt eine Referenz auf die ISAPrflTab-Schnittstelle und alle Zusatzinformationen.
7. Das unter Punkt 1 erzeugte Control kennt den SAP DataProvider nicht.
 - a) Das ABAP-Programm ruft CONTROL_ASSIGN_PROPERTY2 auf. Damit wird eine Eigenschaft des DataProvider-Objekts einer Eigenschaft des Control zugewiesen.
Abhängig vom Datentyp, vom MIME-Type und der verwendeten Eigenschaft (Medium) werden die Daten an das Control übergeben.
8. Die Transaktion wird beendet.
9. Eine neue Transaktion wird gestartet.
 - a) Das SAP GUI erkennt, daß die letzte Transaktion beendet wurde.
 - b) Das SAP GUI erzeugt eine Referenz auf das globale LocalTables-Objekt und informiert dieses Objekt über die Beendigung der letzten Transaktion, indem es die Kontextkennung der beendeten Transaktion übergibt.
 - i) Das LocalTables-Objekt löscht alle Referenzen auf Tabellen mit der angegebenen Kontextkennung und gibt diese Referenzen frei (Garbage Collection).

Andere Datenquellen

Falls das Datenobjekt nicht aus dem ABAP-Programm, sondern aus einer externen Quelle stammt, können Sie dieselbe Vorgehensweise mit folgenden Ausnahmen anwenden:

- Lassen Sie Punkt 2 und 3 für die Übergabe an den Client weg.
- Bilden Sie die Referenz auf das Datenobjekt über die URL.

Momentan werden die Namensräume ftp:, http: und file: unterstützt.

Datenübergabe vom Client an das ABAP-Programm

1. Das ABAP-Programm erzeugt über den Aufruf von CONTROL_CREATE oder eines anderen ähnlichen Bausteins ein Control.
2. Das unter Punkt 1 erzeugte Control kennt das DataProvider-Objekt. Falls nicht, siehe Punkt 3.
 - a) Das Control erzeugt ein DataProvider-Objekt.

Funktionsbeschreibung

- b) Das Control weist dem DataProvider-Objekt eine Referenz auf seine Daten in einem geeigneten Medium zu.
 - c) Das Control ruft die Methode SaveDataToUrl im DataProvider-Objekt auf. Entweder gibt das Control die URL an, oder das DataProvider-Objekt generiert eine URL mit der Syntax „sapr3://<GUID>“.
 - i) Das DataProvider-Objekt erzeugt eine Referenz auf das globale LocalTables-Objekt.
 - ii) Sowohl die URL als auch eine Referenz auf das Datenobjekt werden an dieses globale Objekt übergeben.
 - iii) Siehe Punkt 4.
3. Das unter Punkt 1 erzeugte Control kennt das DataProvider-Objekt nicht.
- a) Das ABAP-Programm erzeugt ein DataProvider-Objekt über DP_CREATE.
 - b) Das ABAP-Programm erzeugt ein entsprechendes Format-Objekt.
 - c) Das ABAP-Programm ruft CONTROL_ASSIGN_PROPERTY2 auf. Dadurch wird eine Eigenschaft des Control einer Eigenschaft des Format-Objekts zugewiesen. Diese Control-Eigenschaft enthält die Daten oder eine Referenz auf die Daten.
 - d) Das ABAP-Programm ruft die Methode SaveDataToUrl im DataProvider-Objekt auf.
 - e) Siehe 2.c.
4. Die Daten werden an das ABAP-Programm übertragen.
- a) Das ABAP-Programm ruft den Funktionsbaustein DP_GET_STREAM auf. Dieser Funktionsbaustein enthält als Parameter eine interne Tabelle, in der die Daten abgelegt werden sollen, sowie die URL des Datenobjekts aus 2.c.
 - b) DP_GET_STREAM ruft die Funktion DP_GET_CLIENT_TABLE im SAP GUI (SAPAWRFC.DLL) auf und übergibt die URL und die Tabelle.
 - i) DP_GET_CLIENT_TABLE erzeugt eine Referenz auf das globale LocalTables-Objekt.
 - ii) Die URL wird an das LocalTables-Objekt übergeben und liefert als Antwort eine Referenz auf das Datenobjekt.
 - iii) Das Datenobjekt wird in die interne Tabelle kopiert, die an DP_GET_CLIENT_TABLE übergeben wurde.
 - iv) Die Referenz auf das Datenobjekt wird aus dem LocalTables-Objekt gelöscht und freigegeben.
 - c) Als Ergebnis liefert DP_GET_STREAM die Daten, den MIME-Typ und die Größe der BLOB-Daten in Byte.

Andere Datenspeicherung

Sie können Daten auch auf ftp:-, http:- oder file:-Servern speichern. Dabei weisen Sie nur die Daten zu und rufen die Methode SaveDataToUrl im DataProvider-Objekt auf.

Da diese oder ähnliche Verfahren für das Senden und Empfangen von Daten immer wieder benötigt werden, sind die wichtigsten Schritte in ABAP-Funktionsbausteinen gekapselt.

Funktionsbausteine

Funktionsbausteine

ABAP-Funktionsbausteine, die vom SAP DataProvider verwendet werden

Funktionsbaustein	Beschreibung
DP_CONTROL_ASSIGN	Weist ein DataProvider-Objekt einem Control zu
DP_CONTROL_ASSIGN_STREAM	Weist eine nicht typisierte interne Tabelle einer Control-Eigenschaft zu
DP_CONTROL_ASSIGN_TABLE	Weist eine typisierte interne Tabelle einer Control-Eigenschaft zu
DP_CONTROL_ASSIGN_URL	Weist eine URL über den SAP DataProvider einer Control-Eigenschaft zu
DP_CONTROL_GET	Weist eine Control-Eigenschaft einem DataProvider-Objekt zu
DP_CONTROL_GET_STREAM	Weist eine Control-Eigenschaft einer nicht typisierten internen Tabelle zu
DP_CONTROL_GET_TABLE	Weist eine Control-Eigenschaft einer typisierten internen Tabelle zu
DP_CREATE	Erzeugt ein DataProvider-Objekt
DP_CREATE_FROM_STREAM	Erzeugt ein DataProvider-Objekt aus einer nicht typisierten internen Tabelle
DP_CREATE_FROM_TABLE	Erzeugt ein DataProvider-Objekt aus einer typisierten internen Tabelle
DP_CREATE_URL	Erzeugt eine Referenz im LocalTables-Datenpool, die auf eine nicht typisierte interne Tabelle zeigt
DP_CREATE_URL_FROM_TABLE	Erzeugt eine Referenz im LocalTables-Datenpool, die auf eine typisierte interne Tabelle zeigt
DP_DESTROY	Zerstört ein DataProvider-Objekt
DP_GET_FIELDS_FROM_TABLE	Liefert die Struktur RFC_FIELDS von einer internen Tabelle
DP_GET_OBJECT	Liefert das Automation-Objekt, das in einem DataProvider-Objekt initialisiert wurde
DP_GET_STREAM	Liest einen Stream aus einem DataProvider-Objekt in eine nicht typisierte interne Tabelle
DP_GET_STREAM_FROM_URL	Liest die Daten einer URL in eine nicht typisierte interne Tabelle
DP_GET_TABLE	Liest eine typisierte interne Tabelle aus einem DataProvider-Objekt
DP_GET_TABLE_FROM_CLIENT	Liest eine typisierte interne Tabelle aus dem LocalTables-Datenpool

Funktionsbausteine

DP_SEND_STREAM	Sendet eine nicht typisierte interne Tabelle an ein DataProvider-Objekt
DP_SEND_TABLE	Sendet eine typisierte interne Tabelle an ein DataProvider-Objekt
DP_SET_OBJECT	Weist ein Objekt einem DataProvider-Objekt zu
DP_SET_URL	Setzt eine URL im DataProvider-Objekt oder im Format-Objekt

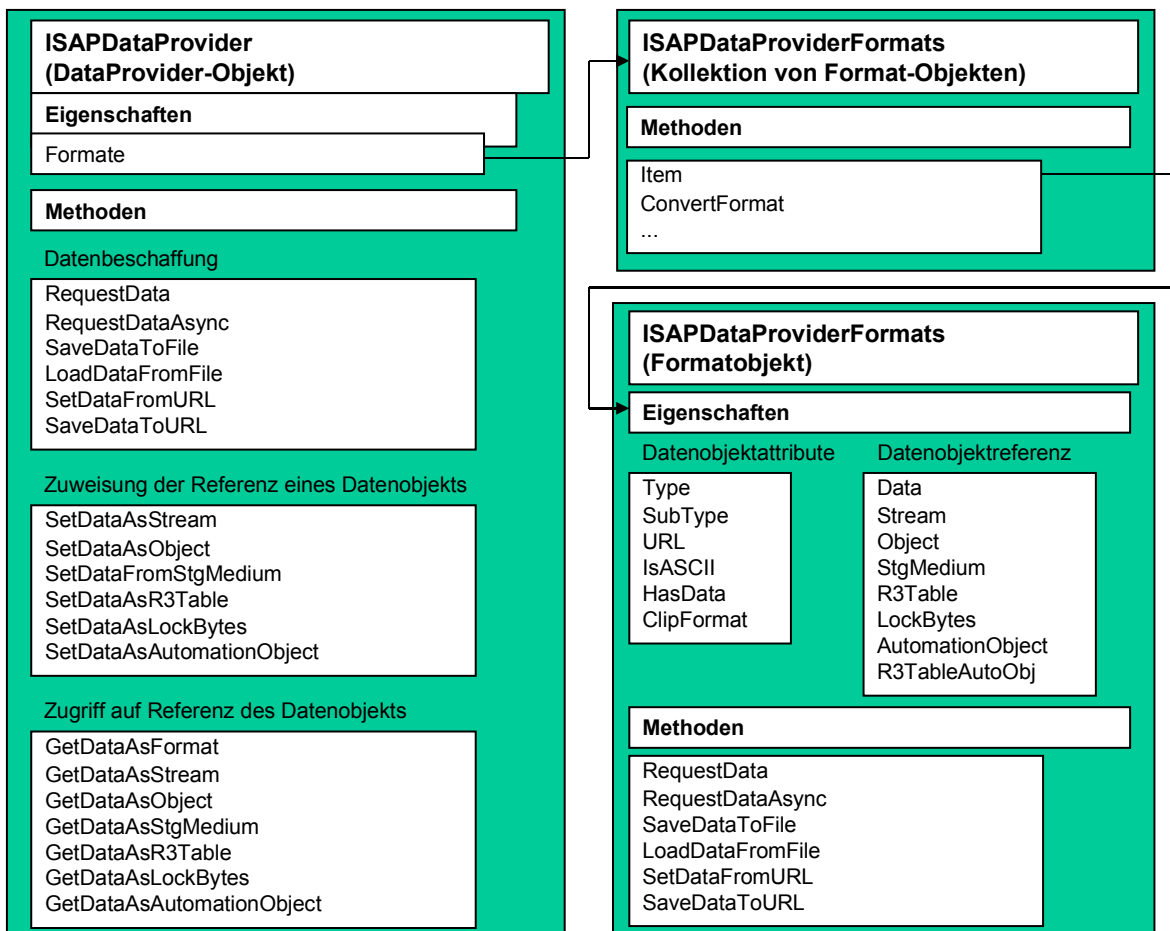
Ausführliche Informationen finden Sie unter der Funktionsgruppe CNDP im R/3-System.

DataProvider-Objekt

DataProvider-Objekt

Allgemeine Funktionsbeschreibung

DataProvider-Objekt



Das DataProvider-Objekt auf Client-Seite verwaltet die Daten in Format-Objekten, auf die Sie über die Kollektion *DataProvider.Formats* zugreifen. Jedes Format-Objekt wird entweder durch seinen MIME-Typ und die Referenz auf ein Datenobjekt oder durch eine URL zur Beschaffung der Daten beschrieben.

Die Anzahl der Format-Objekte oder DataProvider-Objekte hängt von den Daten ab:

- Werden auf dem Client mehrere logisch identische Datenobjekte in verschiedenen Formaten benötigt, muß es mehrere Format-Objekte geben.

Ein Beispiel dafür wäre ein Text, der einmal als RTF-Text und einmal als WinWord-Dokument vorliegt.

DataProvider-Objekt

- Werden auf dem Client mehrere Datenobjekte benötigt, die logisch nicht identisch sind, müssen Sie mehrere DataProvider-Objekte verwenden.
Dies ist kein Problem, da die Erzeugung eines DataProvider-Objekts nur geringe Laufzeit benötigt.

Anlegen neuer Format-Objekte

Sie können neue Format-Objekte folgendermaßen anlegen:

- automatisch durch Formatkonvertierungen
- explizit durch Zuweisung

Bei expliziter Zuweisung muß der Benutzer dafür sorgen, daß die Daten logisch identisch sind.



Da sich in einem Format-Objekt praktisch jedes beliebige Datenobjekt befinden kann, können Sie nicht davon ausgehen, daß mehrere Format-Objekte in einem DataProvider-Objekt immer identische Daten enthalten, da nicht jedes Datenobjekt DataChange-Ereignisse zur Verfügung stellt.

Da eine Konvertierung nach jedem DataChange zudem sehr teuer wäre, ist nur direkt nach dem Aufruf von *DataProvider.Formats.ConvertFormat* gewährleistet, daß Format-Objekte identische Daten enthalten.

Datenzugriff

Sie haben folgende Möglichkeiten, auf Daten zuzugreifen:

- direkt über das Format-Objekt
- über Methoden des DataProvider-Objekts mit Angabe des gewünschten Formats (MIME-Typs)



SAP empfiehlt Ihnen, immer über das DataProvider-Objekt auf Daten zuzugreifen.

Das DataProvider-Objekt und die Format-Objekte enthalten noch weitere Informationen, die ausführlich in der Schnittstellen-Dokumentation beschrieben werden.

Daten werden entweder über eine URL oder durch die direkte Zuweisung von Daten beschafft. Wenn Daten über eine URL beschafft werden, können Sie entweder ein neues Format-Objekt anlegen oder die Methode *DataProvider.SetDataFromUrl* aufrufen. Sie können außerdem über den MIME-Typ des Format-Objekts angeben, welche MIME-Typen für Sie interessant sind.

Je nach Bedarf können Sie die Daten

- synchron oder
- asynchron

anfordern. Bei asynchroner Beschaffung wird der Nutzer der Daten über Ereignisse des DataProvider-Objekts über den Status seiner Anfrage informiert.

Die Beschaffung von Daten über http:, ftp: und file: sowie aus dem LocalTables-Objekt läuft für den Benutzer transparent ab. Wenn Sie Daten aus dem LocalTables-Objekt adressieren,

DataProvider-Objekt

müssen Sie die Daten zuerst über den Funktionsbaustein DP_CREATE_URL aus dem ABAP-Programm exportieren. Die Daten werden im Hauptspeicher des Client abgelegt.

Senden von Daten

Über das DataProvider-Objekt können Sie Daten senden an:

- einen Server
 - Dazu verwenden Sie die Methode *SaveDataToUrl* im DataProvider-Objekt bzw. im Format-Objekt. Berücksichtigen Sie dabei eventuelle Probleme mit Schreibberechtigungen.
- ein ABAP-Programm
 - Dazu geben Sie eine URL an, die **nicht** mit http:, ftp: oder file: beginnt. Wenn Sie keine URL angeben, wird eine eindeutige URL („sapr3://<GUID>“) generiert.
 - Über die Funktionsbausteine DP_GET_TABLE, DP_GET_STREAM oder DP_GET_OBJECT kann das ABAP-Programm auf die Daten zugreifen, die über diese URL adressiert werden.

Einheitliche Datenbeschaffung

Um die Datenbeschaffung zu vereinheitlichen, werden Datenobjekte, die vom DataProvider-Objekt beschafft werden, immer über eine URL mit einem eindeutigen Namen adressiert. Gegenwärtig werden http:, ftp:, file: und sapr3: unterstützt (sapr3: bezieht sich auf Tabellen, die über RFC aus einem R/3-System beschafft wurden).

Um die Verarbeitung der vom DataProvider-Objekt beschafften Datenobjekte auf dem Client so weit wie möglich zu standardisieren, müssen allen Datenobjekten MIME-Typen zugewiesen werden. Das DataProvider-Objekt nimmt daher nur Daten mit einem MIME-Typ auf. Entweder geben Sie den MIME-Typ beim Setzen der Daten an, oder er wird aus dem HTTP-ResponseHeader bzw. dem im ABAP-Programm angegebenen MIME-Typ bestimmt. Gegenwärtig wird der MIME-Typ **application/x-unknown** für die Namensräume ftp: und file: verwendet. Daten, die aus einem ABAP-Programm übertragen werden, müssen ebenfalls mit einem MIME-Typ übergeben werden. Für typisierte interne Tabellen wird der MIME-Typ **application/x-r3table** benutzt.

Sie können Daten dem SAP DataProvider auch direkt zuweisen. Dabei verwenden Sie keine URL, sondern weisen die Daten über Medien (Schnittstellen) zu, die dem DataProvider-Objekt bekannt sind. Bei der direkten Zuweisung von Daten müssen Sie immer den MIME-Typ angeben.



Wenn Sie typisierte interne Tabellen verwenden, benutzen Sie nicht den MIME-Typ **application/x-r3table**, da für solche Tabellen spezielle Funktionsbausteine zur Verfügung stehen. Diese Funktionsbausteine erzeugen und übergeben Zusatzinformationen, die für diesen MIME-Typ benötigt werden.

Einheitliche Datendarstellung

Der SAP DataProvider bildet Datenobjekte in verschiedenen Formen ab, die als **Medien** bezeichnet werden.

Ein Medium stellt die Daten in einem bestimmten Speichermedium (bzw. einer bestimmten Schnittstelle) dar; Format und Inhalt bleiben unverändert. Die verwendeten Medien sind

DataProvider-Objekt

größtenteils Standard-COM-Schnittstellen, doch für typisierte Tabellen werden auch SAP-spezifische Schnittstellen verwendet. Der Zugriff auf OLE-DB-Rowsets ist ebenfalls möglich.

Das DataProvider-Objekt unterstützt folgende Medien:

Medium	Datendarstellung	Hinweis
Stream	IStream-Schnittstelle	Entspricht einem Byte-Stream mit Cursor
LockBytes	ILockBytes-Schnittstelle	Entspricht einem Byte-Stream ohne Cursor
Object	IUnknown-Schnittstelle	Entspricht einem COM-Objekt
AutomationObject	IDispatch-Schnittstelle	Entspricht einem OLE-Automation-Objekt ACHTUNG: Das AutomationObject hängt stark vom MIME-Typ ab. In der Regel unterscheiden sich die Methoden und Eigenschaften des über IDispatch dargestellten Objekts für jedes Format-Objekt erheblich. Die Standardimplementierung unterstützt AutomationObject nur für OLE-Dokumente eines OLE-Automation-Server. Bei anderen Typen hängt dies von einem Mediumkonvertierungsobjekt ab. Informationen darüber, welche Möglichkeiten ein entsprechendes AutomationObject einer Konvertierung bietet, finden Sie in der Dokumentation des Konvertierungsobjekts.
R3Table	Interne Tabelle Der Datenzugriff erfolgt über die ISAPDPR3Table-Schnittstelle. Diese Schnittstelle liefert die ISAPrflTab-Schnittstelle zusammen mit den Rowset-Schnittstellen, die für OLE-DB notwendig sind.	Um auf das OLE-DB-Rowset zuzugreifen, verwenden Sie die ISAPrflTab-Schnittstelle oder konvertieren die Format-Objekte von application/x-r3table in application/x-rowset . Die Konvertierung ist einfacher und erfordert keine Kopie der Daten. In Visual Basic ist dies außerdem der einzige Weg, auf ein RecordSet zuzugreifen.
STGMEDIUM	STGMEDIUM-Struktur	Diese Struktur ist zwar keine Schnittstelle, kann aber auch über Reference-Counting verwaltet werden.

DataProvider-Objekt

R3TableAutoObj	IDispatch-Schnittstelle	Diese automatisierbare Schnittstelle hat die Methoden der ISAPDPR3Table-Schnittstelle.
Data	VARIANT	Enthält immer eine Kopie der Daten. ACHTUNG: Data hängt vom MIME-Typ ab. In der Regel sind die im VARIANT gespeicherten Daten von Format-Objekt zu Format-Objekt sehr unterschiedlich.
File	Die Daten können in eine Datei geschrieben oder aus einer Datei gelesen werden.	

Bei direkter Zuweisung von Daten müssen Sie die gleichen Medien verwenden:

- Bei den meisten Medien ist die Zuweisung sehr einfach, da Sie nur eine Referenz zuweisen müssen.
- Bei den Medien Data und File müssen Sie eine Kopie der Daten im Format-Objekt des DataProvider-Objekts anlegen.



Es werden nicht alle Medien für ein Format unterstützt. Außerdem unterstützen die Standardkonvertierungen nicht die Konvertierung aller Medien. Informationen über die verfügbaren Medien für ein Format finden Sie in der Dokumentation des Mediumkonvertierungsobjekts.

Technische Referenz

ISAPDataProvider

```
interface ISAPDataProvider : IDispatch
```

```
HRESULT get_PreferedFormat([out, retval] BSTR *)
```

Nicht implementiert

```
HRESULT put_PreferedFormat([in ] BSTR)
```

Nicht implementiert

```
HRESULT get_DefaultConsumer([out, retval] VARIANT *retval)
```

Nicht implementiert

```
HRESULT put_DefaultConsumer([in ] VARIANT Consumer)
```

Nicht implementiert

```
HRESULT get_Formats([out, retval] ISAPDataProviderFormats **retval)
```

Liefert die Kollektion der aktuell im DataProvider-Objekt vorhandenen Format-Objekte als ISAPDataProviderFormats-Schnittstelle in *retval*.

Der Aufrufer ist dafür verantwortlich, die Referenz auf den Parameter *retval* freizugeben.

```
HRESULT get_Consumers([out, retval] ISAPDataProviderConsumers **retval)
```

Liefert die Kollektion der aktuell am DataProvider-Objekt registrierten Consumers.

Alle Objekte in dieser Kollektion erhalten die Ereignisse des DataProvider. Um einen Consumer in dieser Kollektion zu registrieren, muß der DataProvider die Standard-OLE-Mechanismen für die Ereignisbehandlung verwenden (siehe OLE2 SDK, IConnectionPoint, IConnectionPointContainer).

```
HRESULT get_IsModified([out, retval] BOOL *retval)
```

Nicht implementiert

```
HRESULT RequestData([in] BSTR Type, [in] BSTR SubType, [in] VARIANT Consumer)
```

Startet eine Datenübergabe für den angegebenen MIME-Typ (*Type/SubType*).

Technische Referenz

Damit die Datenübergabe erfolgen kann, muß in der Format-Kollektion ein passendes Format-Objekt vorhanden sein, das wiederum eine gültige URL enthalten muß.

Die Daten werden synchron beschafft. Läuft dieser Vorgang erfolgreich ab, können die Daten anschließend vom entsprechenden Format-Objekt angefordert werden. Der Consumer sowie alle Objekte in der Consumer-Kollektion werden mit Hilfe von Ereignissen über Status und Erfolg der Datenübergabe informiert. Liegen bereits Daten im angegebenen Format vor, werden diese mit den neuen Daten überschrieben.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Datenobjekts. Optional
SubType	MIME-Typ des Datenobjekts. Optional
Consumer	Zusätzliches Objekt, das nur an den Ereignissen dieser Anfrage interessiert ist

```
HRESULT RequestDataAsync( [in] BSTR Type, [in] BSTR SubType,
[in] VARIANT Consumer)
```

Startet eine asynchrone Datenübergabe für den angegebenen MIME-Typ (*Type/SubType*).

Damit die Datenübergabe erfolgen kann, muß in der Format-Kollektion ein passendes Format-Objekt vorhanden sein, das wiederum eine gültige URL enthalten muß.

Die Daten werden asynchron beschafft. Auch wenn dieser Vorgang erfolgreich abläuft, können die Daten erst dann vom entsprechenden Format-Objekt angefordert werden, wenn das entsprechende DataReady-Ereignis ausgelöst wird. Der Consumer sowie alle Objekte in der Consumer-Kollektion werden mit Hilfe von Ereignissen über Status und Erfolg der Datenübergabe informiert. Liegen bereits Daten im angegebenen Format vor, werden diese mit den neuen Daten überschrieben.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Datenobjekts. Optional
SubType	MIME-Typ des Datenobjekts. Optional
Consumer	Zusätzliches Objekt, das nur an den Ereignissen dieser Anfrage interessiert ist

```
HRESULT AbortRequest( [in] BSTR Type,
[in] BSTR SubType)
```

Bricht eine asynchrone Datenübergabe für den angegebenen MIME-Typ (*Type/SubType*) ab.

```
HRESULT GetLastChange( [in] BSTR Type, [in] BSTR SubType, [out,retval]
VARIANT *Delta)
```

Nicht implementiert

HRESULT SaveDataToFile([in] BSTR Type, [in] BSTR SubType, [in] BSTR FileName)

Speichert die Daten des passenden Format-Objekts gemäß *Type/SubType* in der über *FileName* angegebenen Datei.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Datenobjekts. Optional
SubType	MIME-Typ des Datenobjekts. Optional
FileName	Pfad der Datei, in die die Daten geschrieben werden sollen

HRESULT GetDataAsStream([in] BSTR Type, [in] BSTR SubType, [out,retval] IStream **Stream)

Liefert die Daten des passenden Format-Objekts gemäß *Type/SubType* als IStream-Schnittstelle.

Der Aufrufer ist dafür verantwortlich, *Stream* freizugeben.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Datenobjekts. Optional
SubType	MIME-Typ des Datenobjekts. Optional
Stream	IStream-Schnittstelle zu Daten im entsprechenden Format-Objekt

HRESULT GetDataAsObject([in] BSTR Type, [in] BSTR SubType, [out,retval] IUnknown **Object)

Liefert die Daten des passenden Format-Objekts gemäß *Type/SubType* als IUnknown-Schnittstelle.

Der Aufrufer ist dafür verantwortlich, *Object* freizugeben.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Datenobjekts. Optional
SubType	MIME-Typ des Datenobjekts. Optional
Object	IUnknown-Schnittstelle zu Daten im entsprechenden Format-Objekt

HRESULT GetDataAsStgMedium([in] BSTR Type, [in] BSTR SubType, [in] DWORD DesiredTyMed,

Technische Referenz

```
[out,retval ] STGMEDIUM *StgMedium)
```

Liefert die Daten des passenden Format-Objekts gemäß *Type/SubType* als STGMEDIUM.

DesiredTymed gibt an, welcher Teil der Struktur *StgMedium* gefüllt werden soll.



Der Aufrufer ist dafür verantwortlich, *StgMedium* über die OLE2-API freizugeben. Das vom DataProvider-Objekt zurückgegebene STGMEDIUM hat in der Regel das Element `punkForRelease` auf dem *StgMedium* belegt. Daher ist es **nicht** zulässig, einzelne Elemente innerhalb *StgMedium* mit entsprechenden Windows-APIs freizugeben.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Datenobjekts. Optional
SubType	MIME-Typ des Datenobjekts. Optional
DesiredTymed	Pfad der Datei, in die die Daten geschrieben werden sollen
StgMedium	STGMEDIUM-Struktur, welche die Daten enthält

```
HRESULT GetDataAsR3Table( [in] BSTR Type, [in] BSTR SubType,
[out,retval ] ISAPDPR3Table **Table)
```

Liefert die Daten des passenden Format-Objekts gemäß *Type/SubType* als ISAPDPR3Table.

Der Aufrufer ist dafür verantwortlich, *Table* freizugeben.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Datenobjekts. Optional
SubType	MIME-Typ des Datenobjekts. Optional
Table	ISAPDPR3Table-Schnittstelle für den Direktzugriff auf die interne Tabelle

```
HRESULT GetDataAsR3TableExt( [in] BSTR Type, [in] BSTR SubType,
[out, retval] VARIANT *var3Table)
```

Liefert die Daten des passenden Format-Objekts gemäß *Type/SubType* als VARIANT vom Typ VT_DISPATCH.

Diese IDispatch ist Teil der ISAPDPR3Table. Der Aufrufer ist dafür verantwortlich, *var3Table* über `VariantClear` freizugeben.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Datenobjekts. Optional

SubType	MIME-Typ des Datenobjekts. Optional
vaR3Table	Variant, das die Schnittstelle als VT_DISPATCH enthält

```
HRESULT GetDataAsFormat( [in] BSTR Type, [in] BSTR SubType,
[out, retval] ISAPDataProviderFormat **FormatObj)
```

Liefert die Daten des passenden Format-Objekts gemäß *Type/SubType* als ISAPDataProviderFormat.

Der Aufrufer ist dafür verantwortlich, *FormatObj* freizugeben.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Datenobjekts. Optional
SubType	MIME-Typ des Datenobjekts. Optional
FormatObj	ISAPDataProviderFormat-Schnittstelle für den Direktzugriff auf das Format-Objekt

```
HRESULT LoadDataFromFile( [in] BSTR Type, [in] BSTR SubType,
[in] BSTR FileName)
```

Liest die Daten aus der Datei *FileName* in das passende Format-Objekt gemäß *Type/SubType*.

Zuvor müssen Sie das Format-Objekt über die Format-Kollektion anlegen.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Datenobjekts. Optional
SubType	MIME-Typ des Datenobjekts. Optional
FileName	Pfad und Name der Datei, die gelesen werden soll

```
HRESULT SetDataAsStream( [in] BSTR Type, [in] BSTR SubType,
[in ] IStream *Stream)
```

Stellt die mit *Stream* übergebenen Daten in das Format-Objekt (*Type/SubType*).

Wenn das Format-Objekt nicht vorhanden ist, wird ein neues Format-Objekt angelegt. Alle im Format-Objekt gespeicherten Referenzen auf andere Datenobjekte werden freigegeben.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Datenobjekts. Dies ist ein Mußparameter .
SubType	MIME-Typ des Datenobjekts. Dies ist ein Mußparameter .

Technische Referenz

Stream	IStream-Schnittstelle, die die neuen Daten enthält
--------	--

```
HRESULT SetDataAsObject( [in] BSTR Type, [in] BSTR SubType,
[in ] IUnknown *Object)
```

Stellt die mit *Object* übergebenen Daten in das Format-Objekt (*Type/SubType*).

Wenn das Format-Objekt nicht vorhanden ist, wird ein neues Format-Objekt angelegt. Alle im Format-Objekt gespeicherten Referenzen auf andere Datenobjekte werden freigegeben.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Datenobjekts. Dies ist ein Mußparameter .
SubType	MIME-Typ des Datenobjekts. Dies ist ein Mußparameter .
Object	IUnknown-Schnittstelle zu einem beliebigen Objekt

```
HRESULT SetDataFromStgMedium( [in] BSTR Type, [in] BSTR SubType,
[in ] STGMEDIUM *StgMedium)
```

Stellt die mit *StgMedium* übergebenen Daten in das Format-Objekt (*Type/SubType*).

Wenn das Format-Objekt nicht vorhanden ist, wird ein neues Format-Objekt angelegt. Alle im Format-Objekt gespeicherten Referenzen auf andere Datenobjekte werden freigegeben. In *StgMedium* darf `pUnkForRelease` **nicht** gesetzt werden. Das Format-Objekt installiert ein eigenes `pUnkForRelease` in *StgMedium*, um die Lebensdauer der Daten zu kontrollieren.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Datenobjekts. Dies ist ein Mußparameter .
SubType	MIME-Typ des Datenobjekts. Dies ist ein Mußparameter .
StgMedium	STGMEDIUM mit den neuen Daten

```
HRESULT SetDataAsR3Table([in ] ISAPDPR3Table *Table)
```

Stellt die mit *Table* übergebenen Daten in das Format-Objekt (*Type/SubType*).

Wenn das Format-Objekt nicht vorhanden ist, wird ein neues Format-Objekt angelegt. Alle im Format-Objekt gespeicherten Referenzen auf andere Datenobjekte werden freigegeben. Da *Type/SubType* Teil der ISAPDPR3Table-Schnittstelle ist, müssen diese Parameter nicht angegeben werden.

```
HRESULT SetDataAsR3TableExt([in ] VARIANT varR3Table)
```

Stellt die mit *varR3Table* übergebenen Daten in ein Format-Objekt des Formats *Type/SubType* gemäß *varR3Table*.

Wenn das Format-Objekt nicht vorhanden ist, wird ein neues Format-Objekt angelegt. Alle im Format-Objekt gespeicherten Referenzen auf andere Datenobjekte werden freigegeben. Da *Type/SubType* Teil der ISAPDPR3Table-Schnittstelle ist, müssen diese Parameter nicht angegeben werden. *VaR3Table* muß ein Variant des Typs VT_DISPATCH sein, der den IDispatch-Teil einer ISAPDPR3Table-Schnittstelle enthält.

```
HRESULT SetDataFromURL([in] BSTR Type, [in] BSTR SubType, [in] BSTR URL)
```

Stellt die über *URL* adressierten Daten in das passende Format-Objekt (*Type/SubType*).

Sie müssen das Format-Objekt zuvor nicht über die Format-Kollektion anlegen. Wurde ein Format-Objekt angelegt, wird dieses auch benutzt. Ansonsten wird abhängig vom MIME-Typ der Daten ein Format-Objekt erzeugt. Passen die Daten der URL nicht zum angegebenen MIME-Typ, werden keine Daten übergeben.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Datenobjekts. Optional
SubType	MIME-Typ des Datenobjekts. Optional
Url	URL-Zeiger auf die Daten. Unterstützt werden die Namensräume ftp:, http: und sapr3:.

```
HRESULT SaveDataToURL( [in] BSTR Type, [in] BSTR SubType, [in] BSTR DesiredURL, [out, retval] BSTR *URL)
```

Speichert die Daten des passenden Format-Objekts in der angegebenen *DesiredURL*.

Das Format-Objekt muß zuvor über die Format-Kollektion angelegt werden und Daten enthalten. Wird keine *DesiredURL* angegeben, wird eine URL der Form „sapr3://<GUID>“ generiert und als *URL* zurückgegeben. Diese kann dann aus einem ABAP-Programm geladen werden. Der Aufrufer ist dafür verantwortlich, *Url* mit sysFreeString freizugeben.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Datenobjekts. Optional
SubType	MIME-Typ des Datenobjekts. Optional
DesiredUrl	URL-Zeiger auf die Daten. Unterstützt werden die Namensräume ftp:, http: und sapr3:.
Url	URL-Zeiger auf die Daten. Unterstützt werden die Namensräume ftp:, http: und sapr3:.

```
HRESULT GetDataAsLockBytes( [in] BSTR Type, [in] BSTR SubType,
```

Technische Referenz

```
[out,retval ] ILockBytes **LockBytes)
```

Liefert die Daten des passenden Format-Objekts gemäß *Type/SubType* als ILockBytes.

Der Aufrufer ist dafür verantwortlich, *LockBytes* freizugeben.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Datenobjekts. Optional
SubType	MIME-Typ des Datenobjekts. Optional
LockBytes	ILockBytes-Schnittstelle zu den Daten im entsprechenden Format-Objekt

```
HRESULT SetDataAsLockBytes( [in] BSTR Type, [in] BSTR SubType,
[in ] ILockBytes *LockBytes)
```

Stellt die mit *LockBytes* übergebenen Daten in das Format-Objekt (*Type/SubType*).

Wenn das Format-Objekt nicht vorhanden ist, wird ein neues Format-Objekt angelegt. Alle im Format-Objekt gespeicherten Referenzen auf andere Datenobjekte werden freigegeben.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Datenobjekts. Dies ist ein Mußparameter .
SubType	MIME-Typ des Datenobjekts. Dies ist ein Mußparameter .
LockBytes	ILockBytes-Schnittstelle, die die neuen Daten enthält. Siehe OLE2-Referenz

```
HRESULT SetDataAsAutomationObject( [in] BSTR Type, [in] BSTR SubType,
[in ] Idispatch *Object)
```

Stellt das mit *Object* übergebene automatisierbare Objekt in das Format-Objekt (*Type/SubType*).

Wenn das Format-Objekt nicht vorhanden ist, wird ein neues Format-Objekt angelegt. Alle im Format-Objekt gespeicherten Referenzen auf andere Datenobjekte werden freigegeben.



Das Medium AutomationObject, das über eine IDispatch-Schnittstelle dargestellt wird, hängt vom MIME-Typ der Daten ab und kann nur durch eine Format- oder Mediumkonvertierung entstehen. Informationen, welches Objekt hier zurückgegeben wird, finden Sie in der Dokumentation des entsprechenden Konvertierungsobjekts. Für typisierte interne Tabellen wird immer der MIME-Typ **application/x-r3table** verwendet. Das AutomationObject dieses MIME-Typs ist das SAP-TableObject aus der SAP-TableFactory (siehe die Dokumentation zum RFC-SDK). Interne Tabellen können außerdem über den MIME-Typ **application/x-rowset** angesprochen werden. In diesem Fall wird ein RecordSet-Objekt (siehe OLE-DB-Referenz) als AutomationObject verwendet. Um ein Objekt des Typs **application/x-rowset** aus einem Objekt des Typs **application/x-r3table** zu erzeugen, müssen Sie die Methode ISAPDataProviderFormats::ConvertFormat der Format-Kollektion verwenden.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Datenobjekts. Dies ist ein Mußparameter .
SubType	MIME-Typ des Datenobjekts. Dies ist ein Mußparameter .
Object	IDispatch-Schnittstelle, die das neue Objekt darstellt. Siehe OLE2-Referenz

```
HRESULT GetDataAsAutomationObject( [in] BSTR Type, [in] BSTR SubType,
[out, retval] IDispatch **Object)
```

Liefert das automatisierbare Objekt innerhalb des passenden Format-Objekts gemäß *Type/SubType* als IDispatch.

Der Aufrufer ist dafür verantwortlich, *Object* freizugeben.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Datenobjekts. Optional
SubType	MIME-Typ des Datenobjekts. Optional
Object	IDispatch-Schnittstelle zum automatisierbaren Objekt im entsprechenden Format-Objekt

ISAPDataProviderFormats

```
interface ISAPDataProviderFormats : IDispatch
```

```
HRESULT Add( [in] BSTR Type,
[in] BSTR SubType,
[in] BSTR Url,
[in] BSTR Description,
[in] BSTR Server,
[out, retval] ISAPDataProviderFormat **retval)
```

Fügt ein neues Format-Objekt zur Kollektion hinzu und gibt das neue Objekt in *retval* zurück.

Der Aufrufer ist dafür verantwortlich, die Referenz auf *retval* freizugeben.

Wenn kein *Type/SubType* angegeben wird, dann wird dieser bei der Datenübergabe (mit RequestData) bestimmt. Wird ein *Type/SubType* angegeben, werden bei einer Datenübergabe nur Daten dieses Typs angefordert und verarbeitet.

Parameter

Parameter	Beschreibung
-----------	--------------

Technische Referenz

Type	MIME-Typ des neuen Format-Objekts. Optional
SubType	MIME-Typ des neuen Format-Objekts. Optional
Url	URL-Zeiger auf die Daten des neuen Format-Objekts
Description	Zusätzliche Beschreibung des Format-Objekts. Sie hat keine Auswirkung auf weitere Verarbeitungsschritte.
Server	Name des Systems, von dem die Daten kommen. Optional
RetVal	ISAPDataProviderFormat-Schnittstelle zum neuen Format-Objekt

```
HRESULT Remove( [in] BSTR FormatName,
[in] BSTR SubFormatName,
[out, retval] BOOL *retval)
```

Löscht das passende Format-Objekt aus der Kollektion.

RetVal ist TRUE, falls ein passendes Format-Objekt gefunden wurde.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des zu löschenden Format-Objekts. Optional
SubType	MIME-Typ des zu löschenden Format-Objekts. Optional
RetVal	TRUE, wenn das Format-Objekt gelöscht werden konnte

```
HRESULT RemoveAll()
```

Löscht alle Format-Objekte aus der Kollektion.

```
HRESULT get_Item( [in] BSTR FormatName, [in] BSTR SubFormatName,
[out, retval] ISAPDataProviderFormat **retval)
```

Liefert das passende Format-Objekt gemäß *Type/SubType* aus der Kollektion.

Der Aufrufer ist dafür verantwortlich, die Referenz auf *retval* freizugeben.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Format-Objekts. Optional
SubType	MIME-Typ des Format-Objekts. Optional
RetVal	ISAPDataProviderFormat-Schnittstelle zum Format-Objekt

```
HRESULT ConvertFormat( [in] BSTR Type, [in] BSTR SubType,
[in] BSTR NewType, [in] BSTR NewSubType,
[in] SapDPMediumType
```

Technische Referenz

```
NewMedium, [in] BOOL DoAutoConversion,
[in] VARIANT Consumer, [out, retval]
ISAPDataProviderFormat **retval);
```

Konvertiert das passende Format-Objekt (*Type/SubType*) in ein neues Format-Objekt (*NewType/NewSubType*).

Dazu muß ein Formatkonvertierungsobjekt installiert sein. *NewMedium* gibt das Medium für das neue Format-Objekt an, mit dem nach der Konvertierung bevorzugt gearbeitet werden soll. Das neue Format-Objekt wird zur Kollektion hinzugefügt und als *retval* zurückgegeben. Der Aufrufer ist dafür verantwortlich, die Referenz auf *retval* freizugeben.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Quellformat-Objekts der Konvertierung
SubType	MIME-Typ des Quellformat-Objekts der Konvertierung
NewType	MIME-Typ des Zielformat-Objekts der Konvertierung
NewSubType	MIME-Typ des Zielformat-Objekts der Konvertierung
NewMedium	Bevorzugtes Medium, das vom neuen Format-Objekt zur Verfügung gestellt werden soll
DoAutoConversion	Steuert automatische Konvertierungen, falls das Quellformat-Objekt neue Daten erhält. Wenn dieser Parameter TRUE ist, wird ConvertFormat erneut aufgerufen, wenn ein neues Datenobjekt in das Quellformat-Objekt gestellt wird.
Consumer	Zusätzlicher Ereignisempfänger nur für diese Konvertierungsfunktionen
RetVal	ISAPDataProviderFormat-Schnittstelle des neuen Format-Objekts

```
HRESULT ConvertFormatAsync( [in] BSTR Type,
[in] BSTR SubType, [in] BSTR NewType,
[in] BSTR NewSubType,
[in] BOOL DoAutoConversion, [in] VARIANT
Consumer, [out, retval] ISAPDataProviderFormat
**retval)
```

Legt ein neues Format-Objekt (*NewType/NewSubType*) an und gibt dieses als *retval* zurück.

Der Aufrufer ist dafür verantwortlich, die Referenz auf *retval* freizugeben. Die Daten werden erst in das neue Format-Objekt konvertiert, wenn sie für das passende Format-Objekt (*Type/SubType*) verfügbar sind. Ansonsten verhält sich diese Methode wie ConvertFormat.

```
HRESULT _NewEnum([out, retval] LPUNKNOWN* retval)
```

Liefert den Standard-OLE-Zähler der Kollektion.

```
HRESULT Force( [in] BSTR Type, [in] BSTR SubType,
[in] BSTR Url, [in] BSTR Description,
```

Technische Referenz

```
[in] BSTR Server, [out, retval] ISAPDataProviderFormat
**retval)
```

Erzwingt das passende Format-Objekt gemäß *Type/SubType*.

Wenn bereits ein passendes Format-Objekt in der Kollektion vorhanden ist, dann wird dieses in *retVal* zurückgegeben. Ansonsten wird ein neues Format-Objekt angelegt und in *retVal* zurückgegeben. Der Aufrufer ist dafür verantwortlich, die Referenz auf *retVal* freizugeben.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des neuen Format-Objekts. Optional
SubType	MIME-Typ des neuen Format-Objekts. Optional
Url	URL-Zeiger auf die Daten des neuen Format-Objekts
Description	Zusätzliche Beschreibung des Format-Objekts. Sie hat keine Auswirkung auf weitere Verarbeitungsschritte.
Server	Name des Systems, von dem die Daten kommen. Optional
RetVal	ISAPDataProviderFormat-Schnittstelle zum neuen Format-Objekt

```
HRESULT ConvertByClipFormat( [in] BSTR Type,
                             [in] BSTR SubType,
                             [in] long cf,
                             [in] SapDPMediumType NewType,
                             [in] BOOL DoAutoConversion,
                             [in] VARIANT Consumer,
                             [out, retval] ISAPDataProviderFormat
**retval)
```

Konvertiert das passende Format-Objekt gemäß *Type/SubType* in ein neues Format-Objekt.

Das neue Format-Objekt muß das angegebene Clipboard-Format unterstützen. *NewMedium* gibt das Medium für das neue Format-Objekt an, mit dem nach der Konvertierung bevorzugt gearbeitet werden soll.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Quellformat-Objekts der Konvertierung
SubType	MIME-Typ des Quellformat-Objekts der Konvertierung
Cf	Clipboard-Format des Zielformat-Objekts
NewType	Bevorzugtes Medium, das vom neuen Format-Objekt zur Verfügung gestellt werden soll
DoAutoConversion	Steuert automatische Konvertierungen, falls das Quellformat-Objekt neue Daten erhält

Consumer	Zusätzlicher Ereignisempfänger nur für diese Konvertierungsfunktionen
RetVal	ISAPDataProviderFormat-Schnittstelle des neuen Format-Objekts

ISAPDataProviderFormat

```
interface ISAPDataProviderFormat : IDispatch
```

```
HRESULT get_Data([out, retval] VARIANT *retval)
```

Liefert die Daten des Format-Objekts als VARIANT.

Der Inhalt von *retval* hängt vom MIME-Typ des Format-Objekts ab. Der Aufrufer ist dafür verantwortlich, *retval* mit VariantClear freizugeben.

```
HRESULT put_Data([in] VARIANT vaData)
```

Stellt die in *vaData* enthaltenen Daten in das Format-Objekt.

In den meisten Fällen können Sie nicht feststellen, ob die Daten dem aktuellen MIME-Typ entsprechen. Achten Sie also darauf, Daten des richtigen Typs zu verwenden. Der Inhalt von *vaData* hängt vom MIME-Typ des Format-Objekts ab. Alle im Format-Objekt gespeicherten Referenzen auf andere Datenobjekte werden freigegeben.

```
HRESULT get_IsUpdateable([out, retval] BOOL *retval)
```

Liefert TRUE, wenn die Daten im Format-Objekt geändert werden dürfen.

```
HRESULT get_IsCacheable([out, retval] BOOL *retval)
```

Liefert TRUE, wenn die Daten auf dem Client persistent gespeichert werden dürfen.

```
HRESULT get_Type([out, retval] BSTR *retval)
```

Liefert den Typ des aktuellen Format-Objekts.

Der Aufrufer ist dafür verantwortlich, *retval* über SysFreeString freizugeben.

```
HRESULT get_IsASCII([out,retval] BOOL *retval)
```

Liefert TRUE, wenn die Daten im Format-Objekt nur aus alphanumerischen Zeichen bestehen.

```
HRESULT get_SubType([out, retval] BSTR *retval)
```

Liefert den SubType des aktuellen Format-Objekts.

```
HRESULT get_URL([out, retval] BSTR *retval)
```

Technische Referenz

Liefert die URL, über die die Daten beschafft werden.

Wenn die Daten direkt über eine `SetDataAs...`-Methode zugewiesen wurden, wird ein Leerstring in `retval` zurückgegeben.

```
HRESULT get_Description([out, retval] BSTR *retval)
```

Liefert eine zusätzliche Beschreibung des Format-Objekts in `retval`.

Der Aufrufer ist dafür verantwortlich, `retval` über `SysFreeString` freizugeben.

```
HRESULT get_NeedsConversion([out, retval] BOOL *retval)
```

Liefert TRUE, wenn das Format-Objekt durch eine Konvertierung entstanden ist und dem Quellformat-Objekt der Konvertierung zwischenzeitlich ein neues Datenobjekt zugewiesen wurde.

```
HRESULT get_HasData([out, retval] BOOL *retval)
```

Liefert TRUE, wenn das Format-Objekt Daten enthält.

```
HRESULT get_AutoConversion([out, retval] BOOL *retval)
```

Liefert TRUE, wenn das Format durch eine Konvertierung mit `bDoAutoConversion = TRUE` entstanden ist.

```
HRESULT put_AutoConversion([in ] BOOL NewVal)
```

Setzt das `AutoConversion`-Kennzeichen des Format-Objekts auf den neuen Wert.

```
HRESULT get_ClipFormat([out, retval] CLIPFORMAT *retval)
```

Liefert das Clipboard-Format des Format-Objekts.

Das Clipboard-Format hängt vom MIME-Typ des Format-Objekts ab und wird in der Registry für verschiedene MIME-Typen hinterlegt.

```
HRESULT put_ClipFormat([in] CLIPFORMAT retval)
```

Weist dem Format-Objekt ein Clipboard-Format zu.

```
HRESULT put_IsUpdateable([in ] BOOL NewVal)
```

Setzt die Eigenschaft `IsUpdateable` auf `NewVal`.

Wenn `IsUpdateable` den Wert TRUE hat, können die Daten des Format-Objekts geändert werden.

```
HRESULT put_IsCacheable([in ] BOOL NewVal)
```

Wenn `IsCacheable` den Wert `TRUE` annimmt, können die Daten des Format-Objekts persistent auf dem Client gespeichert werden.

```
HRESULT put_Description([in ] BSTR NewDesc)
```

Fügt die angegebene Beschreibung *NewDesc* in das Format-Objekt ein.

Diese Beschreibung ist keine semantische Bedeutung.

```
HRESULT RequestData([in] VARIANT Consumer)
```

Enthält die Eigenschaft `URL` des Format-Objekts eine gültige URL, werden die Daten, auf die diese URL zeigt, synchron geladen und im Format-Objekt gespeichert.

Über Ereignisse werden alle `Consumers` des `DataProvider`-Objekts sowie der angegebene *Consumer* über den Status und den Erfolg der Datenübergabe informiert.

```
HRESULT RequestDataAsync([in] VARIANT Consumer)
```

Enthält die Eigenschaft `URL` des Format-Objekts eine gültige URL, werden die Daten, auf die diese URL zeigt, asynchron geladen und im Format-Objekt gespeichert.

Über Ereignisse werden alle `Consumers` des `DataProvider`-Objekts sowie der angegebene *Consumer* über den Status und den Erfolg der Datenübergabe informiert.

```
HRESULT SaveDataToFile([in] BSTR FileName)
```

Speichert die im Format-Objekt enthaltenen Daten in die Datei, die unter *FileName* angegeben ist.

```
HRESULT LoadDataFromFile([in] BSTR FileName)
```

Lädt die Daten der unter *FileName* angegebenen Datei in das Format-Objekt.

```
HRESULT get_Stream([out,retval ] IStream **Stream)
```

Liefert die Daten des Format-Objekts in *Stream* zurück.

Der Aufrufer ist dafür verantwortlich, die Referenz freizugeben.

```
HRESULT get_Object( [out,retval ] IUnknown **Object)
```

Liefert das Objekt im Format-Objekt in *Object* zurück.

Der Aufrufer ist dafür verantwortlich, die Referenz auf *Object* freizugeben.

```
HRESULT get_AutomationObject( [out,retval ] IDispatch **Object)
```

Liefert das `AutomationObject` des Format-Objekts in *Object* zurück.

Technische Referenz

Der Aufrufer ist dafür verantwortlich, die Referenz auf *Object* freizugeben. Ob aus den Daten des Format-Objekts ein IDispatch abgeleitet wird, hängt vom MIME-Typ des Format-Objekts ab.

```
HRESULT GetDataAsStgMedium( [in] DWORD DesiredTyMed,  
[out,retval ] STGMEDIUM *StgMedium)
```

Liefert die Daten des Format-Objekts in *StgMedium* zurück.

Der Aufrufer ist dafür verantwortlich, die Referenz auf *StgMedium* (siehe OLE2-API) freizugeben. Der gewünschte TYMED (siehe OLE2-API) wird in *DesiredTyMed* angegeben. Sie können verschiedene TYMED-Objekte über einen | verknüpfen.

```
HRESULT get_R3Table([out,retval ] ISAPDPR3Table **Table)
```

Liefert die Daten des Format-Objekts als *ISAPDPR3Table*.

Der Aufrufer ist dafür verantwortlich, die Referenz freizugeben. Ein Teil der *ISAPDPR3Table*-Schnittstelle ist die *ISAPrcfITab*-Schnittstelle. Sie erlaubt Ihnen den Zugriff auf die interne Tabelle.

```
HRESULT putref_Stream([in ] IStream *Stream)
```

Speichert die Referenz auf *Stream* im Format-Objekt.

Alle im Format-Objekt gespeicherten Referenzen auf andere Datenobjekte werden freigegeben. Diese Referenz wird solange im Format-Objekt gehalten, bis neue Daten zugewiesen werden oder das Format-Objekt selbst freigegeben wird. Da der MIME-Typ des Stream in der Regel nicht bestimmt werden kann, muß der Benutzer einen *Stream* angeben, der dem aktuellen MIME-Typ entspricht.

```
HRESULT putref_Object([in ] IUnknown *Object)
```

Speichert die Referenz auf *Object* im Format-Objekt.

Alle im Format-Objekt gespeicherten Referenzen auf andere Datenobjekte werden freigegeben. Diese Referenz wird solange im Format-Objekt gehalten, bis neue Daten zugewiesen werden oder das Format-Objekt selbst freigegeben wird. Da der MIME-Typ des Objekts in der Regel nicht bestimmt werden kann, muß der Benutzer ein *Object* angeben, das dem aktuellen MIME-Typ entspricht.

```
HRESULT SetDataFromStgMedium([in ] STGMEDIUM *StgMedium)
```

Füllt das Format-Objekt mit den Daten aus *StgMedium*.

Alle im Format-Objekt gespeicherten Referenzen auf andere Datenobjekte werden freigegeben. Diese Referenz wird solange im Format-Objekt gehalten, bis neue Daten zugewiesen werden oder das Format-Objekt selbst freigegeben wird. Da der MIME-Typ des Objekts in der Regel nicht bestimmt werden kann, muß der Benutzer ein *StgMedium* angeben, das dem aktuellen MIME-Typ entspricht.

```
HRESULT putref_R3Table([in ] ISAPDPR3Table *Table)
```

Speichert die Referenz auf *Table* im Format-Objekt.

Alle im Format-Objekt gespeicherten Referenzen auf andere Datenobjekte werden freigegeben. Diese Referenz wird solange im Format-Objekt gehalten, bis neue Daten zugewiesen werden oder das Format-Objekt selbst freigegeben wird. Da der MIME-Typ in ISAPDPR3Table enthalten ist, werden Daten mit dem falschen MIME-Typ nicht angenommen.

```
HRESULT SetDataFromURL([in ] BSTR Url)
```

Liest die Daten der angegebenen URL und speichert sie im Format-Objekt.

Alle im Format-Objekt gespeicherten Referenzen auf andere Datenobjekte werden freigegeben. Diese Referenz wird solange im Format-Objekt gehalten, bis neue Daten zugewiesen werden oder das Format-Objekt selbst freigegeben wird. Daten mit nicht passendem MIME-Typ werden nicht angenommen.

```
HRESULT SaveDataToURL([in ] BSTR DesiredUrl,[out, retval] BSTR *Url)
```

Übergibt die Daten im Format-Objekt an *DesiredUrl*.

Wird keine *DesiredUrl* angegeben, werden die Daten im globalen LocalTables-Objekt gespeichert, und es wird eine eindeutige URL generiert. Diese URL wird in *URL* an den Aufrufer zurückgegeben. Der Aufrufer ist dafür verantwortlich, *Url* über SysFreeString freizugeben.

```
HRESULT putref_AutomationObject( [in ] IDispatch *Object)
```

Speichert die Referenz auf *Object* im Format-Objekt.

Alle im Format-Objekt gespeicherten Referenzen auf andere Datenobjekte werden freigegeben. Diese Referenz wird solange im Format-Objekt gehalten, bis neue Daten zugewiesen werden oder das Format-Objekt selbst freigegeben wird. Da der MIME-Typ von IDispatch in der Regel nicht bestimmt werden kann, muß der Benutzer ein *Object* angeben, das dem aktuellen MIME-Typ entspricht.

```
HRESULT putref_LockBytes([in] ILockBytes *LockBytes)
```

Speichert die Referenz auf *LockBytes* im Format-Objekt.

Alle im Format-Objekt gespeicherten Referenzen auf andere Datenobjekte werden freigegeben. Diese Referenz wird solange im Format-Objekt gehalten, bis neue Daten zugewiesen werden oder das Format-Objekt selbst freigegeben wird. Da der MIME-Typ von ILockBytes in der Regel nicht bestimmt werden kann, muß der Benutzer ein *LockBytes* angeben, das dem aktuellen MIME-Typ entspricht.

```
HRESULT get_LockBytes([out, retval ] ILockBytes **LockBytes)
```

Liefert die Daten des Format-Objekts in *LockBytes* zurück.

Der Aufrufer ist dafür verantwortlich, die Referenz freizugeben.

```
HRESULT putref_UserObj([in] IUnknown *AnyObj)
```

Technische Referenz

Über die Eigenschaft `UserObj` können Sie jedes beliebige Objekt zum Format-Objekt hinzufügen. Durch diesen Aufruf werden bereits vorhandene Referenzen auf ein `UserObj` freigegeben.

```
HRESULT get_UserObj( [out, retval] IUnknown **AnyObj)
```

Liefert das zum Format-Objekt hinzugefügte `UserObj`.

Der Aufrufer ist dafür verantwortlich, die Referenz auf *AnyObj* freizugeben.

```
HRESULT get_R3TableAutoObj( [out, retval] IDispatch **AnyObj)
```

Liefert die `IDispatch`-Schnittstelle der `ISAPDPR3Table`-Schnittstelle, falls Daten im Medium `R3Table` vorhanden sind.

```
HRESULT putref_R3TableAutoObj( [in ] IDispatch *Object)
```

Object muß das `IDispatch` einer `ISAPDPR3Table`-Schnittstelle sein.

```
HRESULT Clone ([in] BSTR Range, [out,retval] ISAPDataProviderFormat **ppRet)
```

Erstellt aus dem aktuellen Format-Objekt ein neue Format-Objekt und gibt dieses in *ppRet* zurück.

Der Aufrufer ist dafür verantwortlich, die Referenz freizugeben. Das neue Format-Objekt wird nicht zur Format-Kollektion hinzugefügt. Der Parameter *Range* beschreibt den Teil des Format-Objekts, der als Klon zurückgegeben werden soll. Die Standardimplementierung unterstützt diese Funktion nicht. Klon-Objekte können nur von Mediumkonvertierungsobjekten erzeugt werden und sind damit vom MIME-Typ des Format-Objekts abhängig.

```
HRESULT Range([out,retval ] BSTR *Range)
```

Gibt den Bereich des Format-Objekts in *Range* zurück.

Der Aufrufer muß *Range* über `SysFreeString` freigeben. Ein Format-Objekt hat nur dann einen *Range*, wenn es über die Methode `Clone` erzeugt wird.

```
HRESULT put_IsASCII([in] BOOL newVal)
```

Setzt die Eigenschaft `IsASCII` auf `TRUE`.

ISAPDataProviderEvents

```
interface ISAPDataProviderEvents : IDispatch
```

```
HRESULT OnStart([in] BSTR Type, [in] BSTR SubType)
```

Dieses Ereignis wird ausgelöst, wenn ein Format-Objekt beginnt, Daten des Formats *Type/SubType* zu beschaffen, die über seine URL adressiert werden.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Format-Objekts
SubType	MIME-Typ des Format-Objekts

```
HRESULT OnProgress([in] BSTR Type, [in] BSTR SubType, [in] USHORT PerCentDone)
```

Dieses Ereignis wird ausgelöst, wenn ein Format-Objekt Daten des Formats *Type/SubType* beschafft, die über seine URL adressiert werden.

PerCentDone gibt den Status des Vorgangs an. Der Wert von *PerCentDone* kann im Bereich zwischen 0 und 100 liegen.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Format-Objekts
SubType	MIME-Typ des Format-Objekts
PerCentDone	Gibt den Status des Ladevorgangs an

```
HRESULT OnDone([in] BSTR Type, [in] BSTR SubType, [in] SCODE ErrorCode)
```

Dieses Ereignis wird ausgelöst, wenn ein Format-Objekt die Beschaffung von Daten des Formats *Type/SubType* beendet, die über seine URL adressiert werden.

Für die Auslösung des Ereignisses spielt es keine Rolle, ob die Datenbeschaffung erfolgreich war oder nicht.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Format-Objekts
SubType	MIME-Typ des Format-Objekts
ErrorCode	0: Erfolgreiche Datenübergabe Sonst: Siehe OLE2 oder INETSDK ErrorCodes

```
HRESULT OnDataReady([in] BSTR Type, [in] BSTR SubType)
```

Dieses Ereignis wird ausgelöst, wenn Daten des Formats *Type/SubType* im Format-Objekt verfügbar werden.

Dies kann durch einen synchronen oder asynchronen Ladevorgang bedingt sein, aber auch durch eine direkte Zuweisung über eine *SetDataAs...*-Methode.

Technische Referenz

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Format-Objekts
SubType	MIME-Typ des Format-Objekts

```
HRESULT OnDataChange([in] BSTR Type, [in] BSTR SubType)
```

Dieses Ereignis wird ausgelöst, wenn **neue** Datenobjekte des Formats *Type/SubType* im Format-Objekt verfügbar werden.

Dies kann durch einen synchronen oder asynchronen Ladevorgang bedingt sein, aber auch durch eine direkte Zuweisung über eine *SetDataAs...*-Methode. Dieses Ereignis hat die gleiche Bedeutung wie das Ereignis *OnDataReady*.



Es werden keine Ereignisse ausgelöst, wenn sich Daten innerhalb eines Datenobjekts ändern. Dabei müssten Sie gewährleisten, daß alle möglichen Datenobjekte entsprechende Ereignisse erzeugen. Da dies jedoch nicht immer der Fall ist, können Sie nicht davon ausgehen, daß diese Ereignisse immer auftreten.

Parameter

Parameter	Bezeichnung
Type	MIME-Typ des Format-Objekts
SubType	MIME-Typ des Format-Objekts

```
HRESULT OnConversionStart([in] BSTR Type, [in] BSTR SubType)
```

Dieses Ereignis wird ausgelöst, bevor ein Format-Objekt in ein neues Format-Objekt des Formats *Type/SubType* konvertiert wird.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Format-Objekts
SubType	MIME-Typ des Format-Objekts

```
HRESULT OnConversionDone([in] BSTR Type, [in] BSTR SubType, [in] SCODE  
ErrorCode)
```

Dieses Ereignis wird ausgelöst, nachdem ein Format-Objekt in ein neues Format-Objekt des Formats *Type/SubType* konvertiert wurde.

Parameter

Parameter	Beschreibung
Type	MIME-Typ des Format-Objekts

SubType	MIME-Typ des Format-Objekts
ErrorCode	0: Erfolgreiche Datenübergabe Sonst: Siehe OLE2 oder INETSDK ErrorCodes

ISAPDPR3Table

Die ISAPDPR3Table-Schnittstelle wird hauptsächlich zur Kapselung interner Tabellen genutzt. An dieser Stelle konnte nicht auf die ISAPrclTab-Schnittstelle zurückgegriffen werden, da im Zusammenhang mit dem DataProvider-Objekt folgende zusätzliche Anforderungen bestehen:

- Die Schnittstelle muß über OLE Automation ansprechbar sein, damit sie aus ABAP-Programmen heraus angesprochen werden oder anderen Objekten zugewiesen werden kann.
- Neben den eigentlichen Daten besitzen die Format-Objekte des DataProvider-Objekts zusätzliche Attribute wie die RFC_FIELDS-Tabelle zur Beschreibung typisierter Tabellen, MIME-Typen, die Größe des Datenobjekts sowie Systeminformationen über Caching-Optionen und Updates usw.
- Format-Objekte des Formats **application/x-r3Table** benötigen eine zusätzliche Beschreibung des Tabellentyps (Schemainformation).

```
interface ISAPDPR3Table : IDispatch
```

```
HRESULT get_SubType([out,retval ] BSTR *retval)
```

Liefert den Datentyp.

Der Aufrufer ist dafür verantwortlich, *retval* über SysFreeString freizugeben.

```
HRESULT get_SubType([out,retval ] BSTR *retval)
```

Liefert den Datensubtyp.

Der Aufrufer ist dafür verantwortlich, *retval* über SysFreeString freizugeben.

```
HRESULT get_Date([out,retval ] BSTR *retval)
```

Liefert das Datum der Daten.

Der Aufrufer ist dafür verantwortlich, *retval* über SysFreeString freizugeben. Das Datum ist standardmäßig das aktuelle Datum. Der Anwendungsprogrammierer kann diesen Wert jedoch ändern, um das Datum der letzten logischen Tabellenänderung anzugeben. Damit können Sie Einträge in einem Cache synchronisieren.

```
HRESULT get_Time([out,retval ] BSTR *retval)
```

Liefert die Uhrzeit der Daten.

Der Aufrufer ist dafür verantwortlich, *retval* über SysFreeString freizugeben. Die Uhrzeit ist standardmäßig die aktuelle Uhrzeit. Der Anwendungsprogrammierer kann diesen Wert jedoch

Technische Referenz

ändern, um das Datum der letzten logischen Tabellenänderung anzugeben. Damit können Sie Einträge in einem Cache synchronisieren.

```
HRESULT get_IsASCII([out,retval ] BOOL *retval)
```

Liefert TRUE, wenn die Daten der Tabelle nur aus ASCII-Zeichen bestehen, FALSE, wenn die Tabelle jedes beliebige Zeichen enthalten kann.

Für nicht typisierte Tabellen liefert `get_IsASCII` TRUE, wenn die Tabelle eine Spalte des ABAP-Datentyps „C“ enthält, ansonsten ist die Spalte vom ABAP-Datentyp „X“.

```
HRESULT get_IsStructured([out,retval ] BOOL *retval)
```

Liefert TRUE, wenn das Objekt eine typisierte Tabelle mit Feldinformationen ist.

```
HRESULT get_IsCacheable([out,retval ] BOOL *retval)
```

Liefert TRUE, wenn die Tabelle in einem Cache gespeichert werden kann.

```
HRESULT get_Description([out,retval ] BSTR *retval)
```

Liefert eine zusätzliche Beschreibung der Tabelle.

Die Beschreibung hat keine besondere semantische Bedeutung. Der Aufrufer ist dafür verantwortlich, *retval* über `SysFreeString` freizugeben.

```
HRESULT get_Platform([out,retval ] BSTR *retval)
```

Liefert die Betriebssystemplattform, auf der die Tabelle erstellt wurde.

Diese Anweisung ist ab Release 4.5A veraltet. Der Aufrufer ist dafür verantwortlich, *retval* über `SysFreeString` freizugeben.

```
HRESULT get_Size([out,retval] ULONG *retval)
```

Liefert die Größe der Daten in der Tabelle (in Byte).

Dieser Wert muß nicht zwingend die Anzahl der Zeilen multipliziert mit der Spaltenbreite sein, da die letzten Zeilen nicht unbedingt komplett gefüllt sein müssen.

```
HRESULT put_Size([in] ULONG Size)
```

Setzt die Größe der Daten in der Tabelle (in Byte).

Bei Bedarf können entsprechend viele Zeilen erzeugt bzw. gelöscht werden.

```
HRESULT get_RowLength([out,retval] ULONG *retval)
```

Liefert die Tabellenbreite.

```
HRESULT get_DataTable([out,retval] ISAPrfcITab **retval)
```

Liefert die ISAPrfcITab-Schnittstelle der Tabelle, welche die Daten enthält.

```
HRESULT get_FormatTable([out,retval] ISAPrfcITab **retval)
```

Liefert die ISAPrfcITab-Schnittstelle der Tabelle, welche die Beschreibung der Tabellenstruktur enthält.

Die darunterliegende Tabelle ist immer vom Typ RFC_FIELDS. Wird NULL zurückgegeben, handelt es sich um ein BLOB-Datenobjekt (Binary Large Object) mit nur einer Spalte der Breite RowLength, die entweder vom ABAP-Datentyp „C“ (get_IsASCII == TRUE) oder vom ABAP-Datentyp „X“ ist.

```
HRESULT CopyDataTable([in] ISAPrfcITab *Table)
```

Kopiert die Daten in der Tabelle in die Tabelle *Table*.

Bei typisierten Tabellen **muß** die Tabelle *Table* der Struktur der Daten entsprechen. In diesem Fall wird zeilenweise kopiert. Bei untypisierten Tabellen kann die Spaltenbreite von *Table* von der Spaltenbreite der Datentabelle dieses Objekts abweichen.

```
HRESULT CopyFormatTable([in] ISAPrfcITab *Table)
```

Kopiert die Felddescription in der Tabelle in die Tabelle *Table*.

Die Tabelle *Table* muß immer vom Typ RFC_FIELDS sein.

```
HRESULT get_IntRep([out,retval] SapDPIntRep *retval)
```

Liefert die Darstellung ganzer Zahlen der internen Tabelle.

Dieser Aufruf ist ab Release 4.5A veraltet.

```
HRESULT CopyPropertiesTable([in] ISAPrfcITab *Table)
```

Nicht implementiert

```
HRESULT put_IsASCII([in ] BOOL newVal)
```

Setzt die Eigenschaft IsASCII auf *newVal*.

Bei typisierten Tabellen ist diese Eigenschaft nicht relevant. Bei BLOB-Daten wird diese Eigenschaft für alle notwendigen Codepage-Konvertierungen und für Konvertierungen in ein OLE-VARIANT verwendet.

Standardmediumkonvertierungen

Das Format-Objekt des DataProvider ermöglicht Standardkonvertierungen für allgemein übliche Medien. Für bestimmte MIME-Typen jedoch können Sie die Standardkonvertierungen mit Hilfe von Mediumkonvertierungsobjekten überschreiben.

Technische Referenz

Typische Standardmediumkonvertierungen sind z.B.:

- Darstellung interner Tabellen als Stream
- Zugriff auf die Automation-Schnittstelle für OLE-Dokumente
- Einlesen von Dateien
- Schreiben von Dateien

Standardkonvertierungen des SAP DataProvider

VON	IN	IStream	ILockBytes	IUnknown	IDispatch	ISAPDR3Table	STGMEDIUM
IStream	Ja	Ja	Falls der Stream ein zusammengesetztes Dokument ist	Falls der Stream ein zusammengesetztes Dokument ist und der Server OLE Automation erlaubt	Ja	Für nicht typisierte Tabellen	
ILockBytes	Ja	Ja	Dokument ist	Server OLE Automation erlaubt	Ja	Nein	
IUnknown	Falls IStream implementiert ist	Nein	Ja	Falls das Objekt OLE Automation erlaubt	Nein	Nein	
IDispatch		Nein	Ja	Ja	Nein	Nein	
ISAPDR3Table	Ja	Ja	Falls die Tabelle ein zusammengesetztes Dokument enthält	Falls die Tabelle ein zusammengesetztes Dokument enthält und der Server OLE Automation erlaubt	Ja	Für nicht typisierte Tabellen	
STGMEDIUM	Ja	Ja	Nein	Nein		Ja	

Die Standardkonvertierung kann zusätzlich IStream-Objekte aus Dateien erzeugen bzw. IStream-Objekte in Dateien umwandeln. Das bedeutet, daß jedes Medium, das in ein IStream umgewandelt werden kann, aus einer Datei erzeugt bzw. in eine Datei geschrieben werden kann.

Standardkonvertierung in Datentyp VARIANT:

- Wenn ein IStream erzeugt werden kann, wird ein VARIANT vom Typ BSTR generiert, falls die Eigenschaft „IsASCII“ des Format-Objekts TRUE ist. Ansonsten wird ein SAFEARRAY von VT_UI1 erzeugt.



Bei dieser Operation werden die Daten **kopiert**.

- Wenn das Medium AutomationObject im Format-Objekt unterstützt wird, wird ein VARIANT vom Typ VT_DISPATCH erzeugt.



Bei dieser Operation wird nur die Referenz kopiert, solange das AutomationObject keine Standardeigenschaft besitzt. Wenn eine Standardeigenschaft implementiert ist, hängt es vom Empfänger der Daten ab, ob das AutomationObject oder die Standardeigenschaft verwendet wird.