
OfficeLink

Release 15.5.0.9

CONTACT Software

Sep. 25, 2018

1	Einleitung	1
2	Excel	2
2.1	Einführung	2
2.2	Dokumente bearbeiten	3
2.3	Daten abgleichen	4
2.4	PowerReports	9
3	Outlook	48
3.1	Einführung	48
3.2	E-Mails und Anlagen	49
4	PowerPoint	52
4.1	Einführung	52
4.2	Dokumente bearbeiten	52
4.3	Daten abgleichen	54
5	Project	59
5.1	Einführung	59
5.2	Dokumente bearbeiten	59
5.3	Projekt-Synchronisation	60
6	Visio	62
6.1	Einführung	62
6.2	Dokumente bearbeiten	62
6.3	Daten abgleichen	64
7	Word	68
7.1	Einführung	68
7.2	Dokumente bearbeiten	68
7.3	Daten abgleichen	70

Einleitung

CONTACT OfficeLink stellt eine tiefe Integration der Microsoft Office Werkzeuge Word, Excel, PowerPoint, Outlook, Visio und Project zur Verfügung. Die Integrationen bieten zum Beispiel die Möglichkeit, Dokumente im System zu suchen, zur Bearbeitung zu öffnen und daraufhin Änderungen in der Datenbank zu speichern. Weiterhin können Dokumentstatus oder -eigenschaften geändert oder direkt aus der Microsoft-Office-Anwendung heraus neue Dokumente im System angelegt werden.

Tab. 1.1: Liste der OfficeLink Anwendungsmöglichkeiten

	Excel	Powerpoint	Word	Project	Visio	Outlook
Dokumentsuche in CDB	X	X	X	X	X	X(1)
Änderungen in CDB speichern	X	X	X	X	X	
Neu anlegen in CDB	X	X	X	X	X	X
Index erzeugen in CDB	X	X	X	X	X	
CDB Status ändern	X	X	X	X	X	
CDB Dokumenteigenschaften anzeigen	X	X	X	X	X	
Synchronisation mit CDB Metadaten	X	X	X		X	
Synchronisation mit CDB Projekten				X		
PowerReports	X					

Bemerkung:

1. Für das Anhängen eines Dokumentes aus dem System an eine Email.
-

Wichtig: Es müssen nicht alle unterstützten Office Anwendungen installiert sein. Nicht zu empfehlen ist jedoch, verschiedene Office Versionen gleichzeitig zu benutzen.

Bemerkung: *CONTACT OfficeLink* kann nur in Verbindung mit einem Windows-Client verwendet werden. Wenn sie mit *CONTACT OfficeLink* arbeiten, darf nur ein Windows-Client geöffnet sein.

2.1 Einführung

- *Excel-Oberfläche:*

Die Excel-Anwendung wird durch die Installation um die OfficeLink-Symboleiste erweitert.

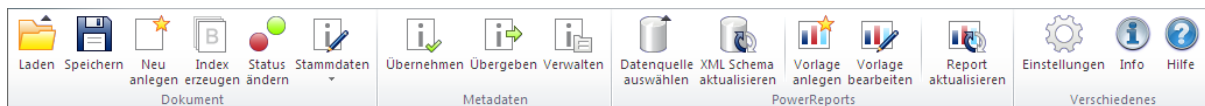


Abb. 2.1: Excel: OfficeLink-Menüleiste

- *Standardoperationen:*

Die System Microsoft Office Integration bietet für Excel die Möglichkeit, Dokumente im System zu suchen, zur Bearbeitung zu öffnen und daraufhin Änderungen in der Datenbank zu speichern. Weiterhin können Dokument-Status oder -Eigenschaften geändert, oder direkt aus Excel heraus neue Dokumente im System angelegt werden.

- *Datenabgleich:*

Die Kopplung bietet die Möglichkeit des Datenabgleichs mit dem System. Voraussetzung dafür ist die Verwendung von vorbereiteten Dokumentvorlagen, die entsprechende System-Variablen enthalten (siehe [Vorlagen zum Datenabgleich erstellen](#) (Seite 4)). Der Datenabgleich wird immer für das aktive Dokument durchgeführt. Das aktive Dokument muss aus dem System zur Bearbeitung geöffnet worden sein.

- *PowerReports:*

PowerReports stellen eine Infrastruktur zur einfachen und schnellen Erstellung von Reports mit Microsoft Excel bereit. Die Architektur ermöglicht die weitreichende Nutzung von Microsoft Excel Funktionen in System-Reports. Ein großer Vorteil der PowerReports Architektur ist beispielsweise die Möglichkeit, Graphiken und Diagramme in System-Reports einzubinden, um Informationen automatisch und ansprechend anhand der Produkt- und Projektdaten zu visualisieren.

- *Einstellungen:*

Auf die Einstellungsmöglichkeiten im Optionsdialog wird im OfficeLink Administrationsbuch eingegangen. Wenden Sie sich bei Fragen an ihren System-Administrator.

2.2 Dokumente bearbeiten

2.2.1 Dokumentensuche im System

Die Schaltfläche [*Laden*] öffnet die Dokumentensuchmaske im System. Die vorgegebene Suchbedingung schränkt die Suche automatisch auf Excel Dokumente ein. Nach Eingabe weiterer Suchkriterien und Ausführen der Suche öffnet sich die Ergebnistrefferverliste mit Vorschaufenster. Durch Auswahl eines oder mehrerer Dokumente werden diese zur Bearbeitung geöffnet.

2.2.2 Im System speichern

Die Schaltfläche [*Speichern*] speichert die Änderungen am aktiven Dokument direkt in das System.

2.2.3 Neuanlage im System

Die Schaltfläche [*Neu anlegen*] führt für das aktive Dokument eine Dokumentenneuanlage im System durch. Dabei öffnet sich zunächst die Maske zur Dokumentenneuanlage. Nach dem Ausfüllen und Bestätigen der Neuanlagemaske wird das Dokument in das System übertragen und abschließend aus dem System neu zur Bearbeitung geöffnet. Das Originaldokument wird bei erfolgreichem Verlauf der Neuanlage geschlossen. Im Fehlerfall, beispielsweise aufgrund fehlender Berechtigungen im System, bleibt das Dokument geöffnet.

Bemerkung: Da mehrere Dateiformate vom System unterstützt werden, wird dort zunächst das Speicherformat ermittelt, bevor die Neuanlagemaske der System-Dokumentenverwaltung geöffnet wird: *Als Standardformat benutzen:* Durch Setzen dieser Checkbox wird das ausgewählte Format als Standardspeicherformat festgelegt. Bei der zukünftigen Neuanlage von Dokumenten wird der Auswahldialog nicht mehr angezeigt und stattdessen das Standardspeicherformat verwendet. Das Standardspeicherformat kann jederzeit im Einstellungsdialog geändert oder entfernt werden. *Excel Open XML Dateiformat und Makros:* In dem Excel Open XML Dateiformat .xlsx können keine Makros gespeichert werden. Beim Versuch Dokumente mit Makros in diesem Format zu speichern, wird eine entsprechende Meldung angezeigt.

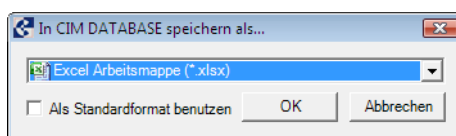


Abb. 2.2: Microsoft Excel: Speicherformatauswahldialog

2.2.4 Index erzeugen im System

Die Schaltfläche [*Index erzeugen*] erzeugt für das aktive Dokument einen neuen Index im System. Dabei öffnet sich zunächst die Maske zur Indexerzeugung. Nach dem Ausfüllen und Bestätigen der Indexmaske werden die aktuellen Änderungen in dem neuen Dokumentenindex im System gesichert.

2.2.5 Dokumentenstatus ändern

Die Schaltfläche [*Status ändern*] führt für das aktive Dokument eine System-Statusänderungsoperation durch.

Bemerkung: Da der Anwender durch die Statusänderung ggf. die Rechte im System verliert, das Dokument erneut zu speichern, speichert die Kopplung das Dokument automatisch, bevor die Statusänderungsoperation aufgerufen wird. Dabei wird ggf. eine entsprechende Meldung angezeigt, die bestätigt werden muss.

2.2.6 Stammdatenänderungsdialog im System öffnen

Die Schaltfläche [*Stammdaten*] öffnet im System den Dialog zum Ändern der Stammdaten für das aktive Dokument. Diese Schaltfläche ist geteilt. Über den unteren Teil kann im System statt des Änderungsdialoges ein Informationsdialog für das aktive Dokument geöffnet werden.

2.3 Daten abgleichen

2.3.1 Metadatenübernahme aus dem System

Durch Betätigung der Schaltfläche [*Übernehmen*] werden die Werte aller System-Variablen mit lesendem Zugriff mit den entsprechenden Werten der zugehörigen System-Attribute aktualisiert. Vom Anwender im Dokument geänderte System- Variablen mit lesendem Zugriff werden dabei überschrieben. Falls ein Fehler im Ausdruck einer Variablen vorliegt, das referenzierte Objekt nicht existiert oder die Beziehung nicht eindeutig ist, wird der Wert auf ein Leerzeichen gesetzt.

2.3.2 Metadatensynchronisation mit dem System

Durch Betätigung der Schaltfläche [*Übergeben*] werden die Werte aller Variablen mit schreibenden Zugriff auf Änderungen überprüft. Wurden Werte im Dokument verändert, werden die betroffenen Objekte nacheinander im System geöffnet. Die Änderungen werden in den Änderungsdialogen vorgeblendet und müssen für jedes Objekt bestätigt werden. Nach Abschluss der Datenübernahme in das System wird eine Metadatenübernahme für System-Variablen mit lesendem Zugriff durchgeführt.

2.3.3 Vorlagen zum Datenabgleich erstellen

Dieser Abschnitt beschreibt die Erstellung von Dokumentvorlagen mit System- Variablen zum Datenabgleich.

- Abgleich von Dokumentmetadaten (z.B. Titel, Dokument-Nr. ...)
- Abgleich von 1:1 referenzierten Objekten (z.B. Projektname des zugeordneten Projekts)
- Übernahme von 1:N referenzierten Objekten (z.B. Liste mit Offenen Punkten)

Formatbeschreibung für System-Variablen

System-Variablen in Excel Dokumenten haben folgendes Format: cdb.<Mode>.<Beziehung>.<Attribut>.<Kardinalität>

Mode Der Mode gibt an, ob eine Variable aus dem System ausschließlich gelesen oder auch nach Änderung im Dokument im System aktualisiert werden darf. Gültige Werte für den Modus sind:

- *r*

Nur Datenübernahme aus dem System möglich. Der übernommene Wert kann im Dokument verändert werden, jedoch nicht an das System übergeben werden. Bei erneuter Datenübernahme wird der geänderte Wert im Dokument wieder durch den System-Wert ersetzt.

- *w*

Nur Datenübergabe in das System möglich. Durch diesen Modus kann sichergestellt werden, dass ein Dokumentfeld durch eine Datenübernahme nicht verändert werden kann, eine Aktualisierung im System aber möglich ist (sinnvoll bei Feldern, die durch das Dokument berechnet werden). Dieser Modus kann nicht bei 1:N Beziehungen verwendet werden.

- *rw*

Datenübernahme aus dem System und Aktualisierung im System nach Änderung im Dokument möglich. Dieser Modus kann nicht bei 1:N Beziehungen verwendet werden.

- **s*

Wenn oben genannten Modewerten ein *s* hinzugefügt wird (d.h. *rs*, *ws* oder *rws*), dann gelten zwar die gleichen Regeln wie bei den jeweiligen beschriebenen Werten ohne *s*, jedoch laufen die Lese- bzw. Schreibaktionen vollständig serverseitig ab. Dabei kann optional das Feld *Parameter* verwendet werden. U.U. bedarf es allerdings für die Verwendung der serverseitigen Bearbeitung von Variablen vorbereitend administratives Customizing, welches im OfficeLink Administrationshandbuch erläutert wird.

Beziehung Die Beziehung gibt ausgehend vom Dokument den Pfad zu den durch die Variable repräsentierten Daten im System an. Als Beziehung können folgende Ausdrücke verwendet werden:

- *this*

Der konstante Ausdruck 'this' wird verwendet, um Attribute des Dokuments selbst anzugeben.

- *Name einer im System konfigurierten Beziehung*

Über eine Beziehung können Attribute von referenzierten Objekten übernommen und abgeglichen werden.

- *BY_ZNUM_ZIDX_FROM_<table_name>*

Verwendung einer impliziten Beziehung über die Attribute *z_nummer* und *z_index*. Hierzu wird der Ausdruck *BY_ZNUM_ZIDX_FROM_<table_name>* angegeben. *table_name* gibt dabei die Datenbanktabelle an, aus der ein Objekt über *z_nummer* und *z_index* implizit referenziert wird. Die angegebene Datenbanktabelle muss die Attribute *z_nummer* und *z_index* enthalten und die Kombination aus *z_nummer* und *z_index* muss eindeutig sein.

Attribut Das Attribut spezifiziert ein Attribut eines durch die Beziehung referenzierten System-Objekts.

Sachmerkmale als Attribute von Artikeln:

Wird eine Artikelbeziehung als Beziehung angegeben, können auch die Sachmerkmale des Artikels als Attribut angegeben werden. Dies kann qualifiziert über die Merkmalkennung oder unqualifiziert mittels Zählnummer erfolgen:

- *Sachmerkmale qualifiziert:*

`__SML_<mm_mk>`

Excel: Datenabgleich - Sachmerkmale qualifiziert

Sachmerkmal mit der Kennung 'L' des zugeordneten Artikels

```
"cdb.r.cdb_doc_to_part.__SML_L"
```

- *Sachmerkmale unqualifiziert:*

`__SMLN<nummer>` für Merkmalskennungen

`__SMLV<nummer>` für Merkmalswerte

Merkmalkennung und der zugehörige Wert müssen dabei dieselbe Nummer haben.

Excel: Datenabgleich - Sachmerkmale unqualifiziert

Kennung und Wert des erstes Sachmerkmals des zugeordneten Artikels

```
"cdb.r.cdb_doc_to_part.__SMLN1"
"cdb.r.cdb_doc_to_part.__SMLV1"
```

Wichtig: SML Attribute können nicht schreibend synchronisiert werden!

Kardinalität Gibt die Kardinalität der Beziehung an. Wenn keine Kardinalität angegeben wurde, wird von der Kardinalität 1 ausgegangen. Bei der Erstellung neuer Dokumentvorlagen sollte die Kardinalität explizit angegeben werden. Folgende Werte können für die Kardinalität verwendet werden:

- 1

Für Beziehungen vom Typ `this, BY_ZNUM_ZIDX_FROM_<table_name>` und für 1-stellige homogene Beziehungen.

- N

Für n-stellige homogene Beziehungen. Variablen, die n-stellige Beziehungen verwenden, müssen spaltenweise in der Kopfzeile einer Liste platziert werden (siehe Verwendung von System-Variablen im Dokument). Beim Abgleich mit dem System wird die Liste auf die zur Darstellung des Ergebnisses benötigte Anzahl Zeilen angepasst. Die erste Zeile nach der Kopfzeile dient dabei als Formatvorlage. Die Sortierung der Liste erfolgt anhand der ersten Spalte.

Excel: Datenabgleich - System-Variablen

Zeichnungsnummer des Dokuments:

```
"cdb.r.this.z_nummer"
```

Übernahme und Abgleich der Bemerkung des Dokuments:

```
"cdb.rw.this.z_bemerkung"
```

Übernahme der Artikelnummer des zugeordneten Artikels:

```
"cdb.r.cdb_doc_to_part.teilenummer"
```

Übernahme und Abgleich der Artikelbenennung des zugeordneten Artikels:

```
"cdb.rw.cdb_doc_to_part.benennung"
```

Übernahme der Begründung aus der Änderungshistorie mittels impliziter Beziehung:

```
"cdb.r.BY_ZNUM_ZIDX_FROM_aenderung.begrueend"
```

Abgleich von Gleitkommazahlen ins System Das Excel Dezimaltrennzeichen ist abhängig von den in der Windows Systemsteuerung eingestellten Regions- und Sprachoptionen. Für Attribute, die im System als *float* definiert sind, muss vor dem Abgleich das Dezimaltrennzeichen ggf. von Komma auf Punkt geändert werden. Damit die Kopplung solche Attribute erkennen und das Dezimaltrennzeichen korrigieren kann, muss an den Variablenausdruck der Dokumentenvariablen *.numeric* angehängt werden.

```
cdb.rw.cdbpcs_doc2project.effort_plan.1.numeric
```

Parameter Über dieses Feld können bei der serverseitigen Bearbeitung von Variablen zusätzliche für die jeweilige Variable spezifische konstante Werte übergeben werden. Das können beispielsweise Identifikatoren sein, welche das Verlinken von Ergebnissen aus 1:N Beziehungen mit einzelnen Dokumentfeldern statt mit dynamischen Listen ermöglichen. Beispielsweise wird der folgenden Variable der Parameter *PRIMARY* hinzugefügt, welches einem zuvor speziell dafür bereitgestelltem Userexit übermittelt, dass man den Ursprungsnamen der Primärdatei des Dokuments haben möchte.

```
cdb.rs.document2cdb_file.cdbf_original_name.N.string.PRIMARY
```

Definition von System-Variablen

Die Definition von System-Variablen erfolgt mit einem Assistenten, der über die Schaltfläche [*Verwalten*] der OfficeLink-Symbolleiste aufgerufen wird. Damit die Schaltfläche verfügbar ist, muss sie zuvor in dem Einstel-

lunsdialog aktiviert worden sein. Der Dialog bietet auf der linken Seite die Möglichkeit neue Variablen im System-Format anzulegen (siehe *Formatbeschreibung für System-Variablen* (Seite 4)). Auf der rechten Seite werden alle im Dokument enthaltenen System-Variablen angezeigt. Durch Markierung einer Variablen wird gleichzeitig die entsprechende Zelle in der Excel Tabelle markiert. Durch Betätigung der Schaltfläche *[Variable löschen]* kann die markierte Variable aus dem Dokument entfernt werden.

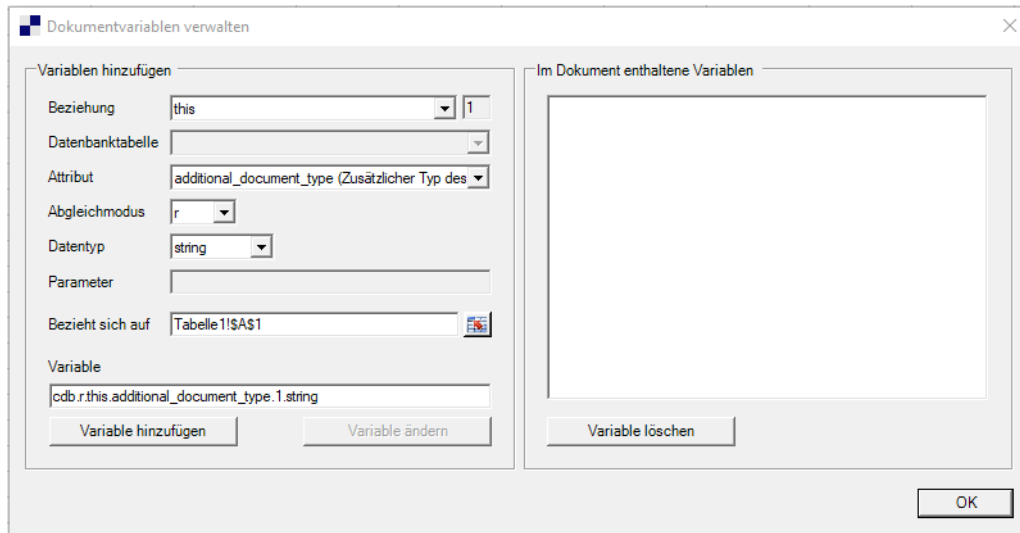


Abb. 2.3: Excel: Datenabgleich - Dialog zur Erstellung von System-Variablen

- *Beziehung:*

Die Beziehung gibt ausgehend vom Dokument den Pfad zu einem Attribut eines oder mehrerer referenzierter System-Objekte an. Der Katalog bietet alle im System konfigurierten Dokumentenbeziehungen an, bei denen das Dokument das Startobjekt darstellt. Für Attribute des Dokuments selbst, muss in diesem Feld *this* eingetragen werden.

Zur Verwendung einer impliziten Beziehung, die über die Schlüsselattribute Dokumentennummer und Index hergestellt wird, muss in diesem Feld *BY_ZNUM_ZIDX_FROM_* ausgewählt werden und im Feld Datenbanktabelle eine Relation als Datenquelle ausgewählt werden.

- *Datenbanktabelle:*

Eine Datenbanktabelle muss nur dann ausgewählt werden, wenn im Feld Beziehung der spezielle Eintrag *BY_ZNUM_ZIDX_FROM_* für implizite Beziehungen ausgewählt wurde. In der ausgewählten Relation muss die Kombination der Attribute *z_nummer* und *z_index* eindeutig sein.

- *Attribut:*

Das Attribut spezifiziert ein Attribut eines durch die Beziehung referenzierten System-Objekts.

- *Abgleichmodus:*

Der Modus gibt an, ob eine Variable aus dem System ausschließlich gelesen oder auch nach Änderung im Dokument im System aktualisiert werden darf:

r: nur lesend

w: nur schreibend

rw: lesender und schreibender Zugriff im System

rs: nur serverseitig lesend

ws: nur serverseitig schreibend

rws: serverseitig lesender und schreibender Zugriff im System

- *Parameter:*

Optionaler nur für das serverseitige Bearbeiten von Variablen verwendeter Wert.

- *Bezieht sich auf:*

In diesem Feld muss die Zelle angegeben werden, in der die System-Variable angezeigt werden soll. Die Excel Zelle sollte aus dem Katalog ausgewählt werden. Beim Öffnen des Katalogs wird der Dialog zur Variablenkonfiguration ausgeblendet und ein Auswahlfenster für die Excel Zelle angezeigt. Die Excel Zelle kann durch Mausklick auf die gewünschte Zelle ausgewählt und anschließend übernommen werden.

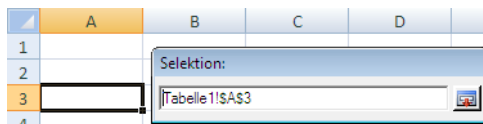


Abb. 2.4: Excel: Datenabgleich - Zelle auswählen

- *System Variable:*

In diesem Feld wird der aus den anderen Feldern erzeugte Ausdruck für die zu erstellenden System-Variable angezeigt. Das manuelle Ändern dieses Ausdrucks ist nur zur Angabe von Sachmerkmalen bei Artikelbeziehungen erforderlich.

Beim Hinzufügen der Variablen durch Betätigung der Schaltfläche [*Variable hinzufügen*] wird der in dem Feld System-Variable angezeigte Ausdruck überprüft und bei Ungültigkeit eine entsprechende Fehlermeldung angezeigt. Ansonsten wird die Variable als neue System-Variable in dem aktiven Dokument angelegt und mit dem Wert <Attribut> als Platzhalter initialisiert. Zellen, auf die sich ein Name bezieht, nehmen bei Aktualisierung den Wert des zugehörigen System- Attributs an. Ein Name kann zusätzlich von weiteren Zellen referenziert werden. Diese Zellen nehmen ebenfalls diesen Wert an.

Verwendung von System-Variablen im Dokument

System-Variablen sollten im Dokument für den Anwender kenntlich gemacht werden, um versehentliches Löschen oder Überschreiben zu vermeiden. Dokumentenvariablen können unter Microsoft Office Excel durch das Sperren der entsprechenden Zellen geschützt werden. Damit dieser Schutz aktiv wird, muss der Blattschutz aktiviert werden. Dabei darf kein Passwort vergeben werden, da die Kopplung beim Datenabgleich diesen Blattschutz temporär aufhebt. Ist bei der Aktivierung des Blattschutzes ein Passwort angegeben worden, wird der Anwender beim Datenabgleich zur Passwordeingabe aufgefordert. Wird das Passwort nicht oder falsch angegeben, schlägt der Datenabgleich fehl.

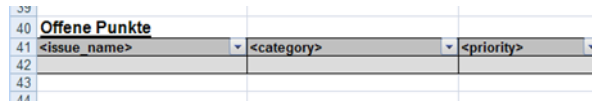
Bemerkung: Beim Datenabgleich von Variablen in das System werden Attributänderungen auch dann vorgeblendet, wenn das Feld laut Maskenkonfiguration durch den Anwender nicht interaktiv änderbar oder nur via Katalogauswahl änderbar wäre. Bei der Verwendung von System-Variablen mit schreibenden Zugriff ist dies zu berücksichtigen. Gegebenenfalls ist die Gültigkeit der Werte durch eine entsprechende Prüfung im User Exit abzusichern.

System-Variablen, die n-stellige Beziehungen verwenden, müssen unter Excel spaltenweise in der Kopfzeile einer Liste platziert werden. Beim Abgleich mit dem System wird die Liste auf die zur Darstellung des Ergebnisses benötigte Anzahl Zeilen angepasst. Die erste Zeile nach der Kopfzeile dient dabei als Formatvorlage. Die Sortierung der Liste erfolgt anhand der ersten Spalte.

Listen/Tabellen in Microsoft Office Excel für n-stellige Beziehungen:

1. Liste/Tabelle mit zwei Zeilen und der gewünschten Anzahl Spalten einfügen.
2. System-Variablen mit dem Assistenten in die erste Zeile einfügen (siehe *Definition von System-Variablen* (Seite 6)). Alle Variablen müssen dabei dieselbe Beziehung verwenden und von der Kardinalität N sein.
3. Platzhalter der System-Variablen in die gewünschten Spaltentitel ändern und wie gewünscht formatieren.

4. Zweite Zeile formatieren. Diese Zeile dient als Kopiervorlage zum Einfügen weiterer Zeilen beim Datenabgleich.



39			
40	Offene Punkte		
41	<issue_name>	<category>	<priority>
42			
43			
44			

Abb. 2.5: Excel: Datenabgleich - List/Tabelle für n-stellige Beziehungen [leer]

Bei Durchführung einer Datenübernahme wird die Tabelle auf die benötigte Anzahl Zeilen expandiert und mit der Ergebnismenge der verwendeten Beziehung gefüllt.



39			
40	Offene Punkte		
41	Name	Kategorie	Priorität
42	Dokumentation aktualisieren	zu vervollständigen	normal
43	Office 2007 Support	Korrektur	hoch
44	Service Level erhöhen	offen	niedrig
45			
46			

Abb. 2.6: Excel: Datenabgleich - List/Tabelle für n-stellige Beziehungen [gefüllt]

Automatisches Aktualisieren von Metadaten

Die Metadaten aller System-Variablen mit lesendem Zugriff werden beim Laden eines Dokuments aus CDB automatisch übernommen. Die Bedingungen dafür sind, dass das Dokument:

1. im Client im "Bearbeiten" Modus geöffnet wurde (oder direkt aus Office heraus über "Laden")
2. eine "Zeichnung" ist (d.h. die Eigenschaft "vorlage_kz" hat)
3. keine Vorlage ist (d.h. "vorlage_kz=0")

2.4 PowerReports

2.4.1 Einführung

Die Reports beziehen ihre Daten aus wiederverwendbaren und parametrisierbaren XML Datenquellen, die im System konfiguriert werden. XML Datenquellen können u.a. Beziehungen des Objekt-Modells (cdb.objects Frameworks) und konfigurierte Objektregeln als sogenannte Datenprovider verwenden. Datenquellen lassen sich damit auch ohne zusätzlichen Programmieraufwand rein konfigurativ festlegen! Soll beispielsweise ein Report für Artikel erstellt werden, der die Stückliste, das zugehörige Projekt und den zugehörigen EC enthält, so kann dieser Report ohne jegliche Programmierung erstellt werden, da die benötigten Beziehungen bereits im Objektmodell modelliert sind. Für komplexe Abfragen oder berechnete Daten können zusätzlich programmierte Datenprovider an eine XML Datenquelle angebunden werden. XML Datenquellen können also beliebige heterogene Daten aus verschiedenen Datenprovidern liefern.

Zellen in einem PowerReport können auch Hyperlinks enthalten, die direkt auf das betreffende Objekt im System verweisen. Zusammen mit den Single-sign-on Möglichkeiten verbinden sich das System und die Office Welt nahtlos miteinander.

Die Erstellung von Reports erfolgt direkt in Excel mit Unterstützung durch die System Office Integration. Nach Auswahl der gewünschten XML Datenquelle wird das für die Datenbindung benötigte XML Schema automatisch aus der konfigurierten Datenquelle erzeugt und in Excel geladen. Anschließend kann die Datenbindung via Drag/Drop hergestellt und das Layout gestaltet werden.

Neue Reports werden via Office Integration als Reportvorlage zur XML Datenquelle gespeichert. Die Reportvorlagen werden also als Dokumente im System verwaltet und der zugrundeliegenden Datenquelle zugeordnet. Mit der neuen Infrastruktur müssen neu erstellte Reports nicht mehr einzeln im System konfiguriert werden, damit diese an der Oberfläche für Anwender verfügbar sind. Stattdessen reicht die Freigabe der Reports durch Zuordnung von berechtigten Rollen oder Personen aus.

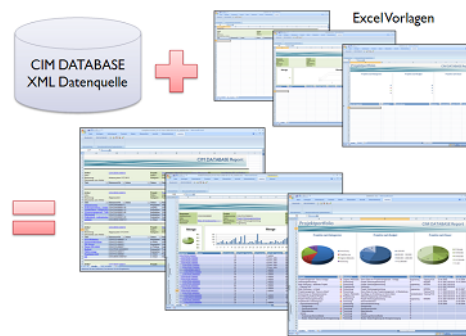


Abb. 2.7: Excel: PowerReports - Schema

2.4.2 Architektur

PowerReports beziehen ihre Daten aus XML Datenquellen, die in der Datenbank konfiguriert werden. Eine XML Datenquelle besteht aus einer beliebigen Anzahl wiederverwendbarer Datenprovider, die die darzustellenden Daten liefern. Eine Datenquelle liefert zwei Arten von Informationen in Form von temporären Dateien, die sich aus der Summe der zugeordneten Provider ergeben. Einerseits das XSD Schema zur Herstellung der Datenbindung in den Reportvorlagen, in der folgenden Abbildung rot dargestellt, und andererseits die Daten in Form einer XML Export Datei.

Die XML Exportdatei wird bei Aufruf eines Reports serverseitig erzeugt und zum Client übertragen. Gleichzeitig wird die im System-Vault zum Report hinterlegte Excelvorlage instanziiert und ebenfalls zum Client übertragen. Der Client verfügt durch die MS-Excel Integration über eine aktive Komponente, die die instanziierte Vorlage und die XML Export Datei zu einem fertigen Report zusammenführt.

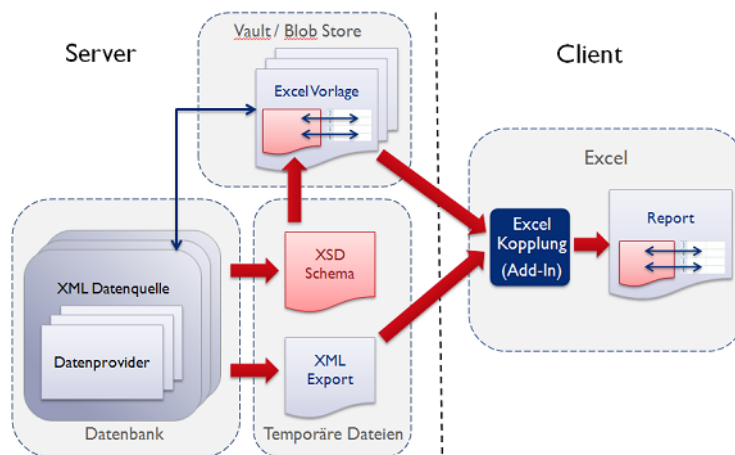


Abb. 2.8: Excel: PowerReports - Architekturübersicht

Das XSD Schema einer Datenquelle ergibt sich aus der Summe der Teilschemata der zugeordneten Provider. Die Datenprovider erzeugen dabei das Schema aus den zugrundeliegenden Datenbanktabellen automatisiert und ergänzen ggf. weitere Attribute. Das Schema eines Datenproviders kann zusätzlich auch völlig datenbankunabhängig erzeugt werden. Dies ist beispielsweise notwendig, wenn ein Provider ausschließlich berechnete Daten oder Daten aus externen Systemen liefert.

Der Aufruf eines Reports kann in Abhängigkeit von der Datenquellenkonfiguration kontextfrei oder kontextbezogen erfolgen. Ein kontextfreier Report liefert in der Regel Daten aus dem Gesamtdatenbestand, beispielsweise aller laufenden Projekte. Ein kontextbezogener Report liefert dagegen Daten, die sich auf eines oder mehrere ausgewählte Objekte beziehen, beispielsweise Informationen zu einem Projekt oder Engineering Change. Der Kontext ergibt sich durch den Aufruf über das Kontextmenü eines oder mehrerer selektierter Objekte. Die selektierten Objekten parametrisieren den Datenexport.

Zusätzlich zu der Parametrisierung durch den Kontext kann ein Dialog vorgeschaltet werden, der weitere Para-

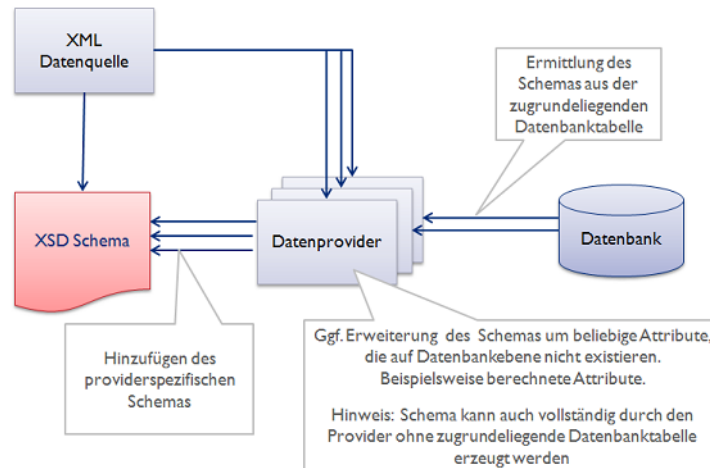


Abb. 2.9: Excel: PowerReports - Funktionsprinzip Schemaausleitung

meter vom Anwender abfragt. In der folgenden Abbildung sind die selektierten Objekte und weitere Parameter aus einem vorgeschalteten Dialog als 'Objekte und Parameter' dargestellt. Die Objekte und Parameter werden beim XML Export an die Datenprovider übergeben. Die Datenprovider ermitteln auf Basis dieser Argumente die Ergebnismenge aus der Datenbank und konstruieren daraus ReportData Objekte die anschließend unter Berücksichtigung des Rechtesystems in eine gemeinsame XML Datei exportiert werden.

Datenprovider können durch hierarchische Anordnung miteinander verkettet werden. Ein untergeordneter Provider erhält in diesem Fall nicht die Eingabeobjekte der XML Datenquelle, sondern die Ergebnismenge des übergeordneten Providers als Argumente. Dies ermöglicht die Parametrisierung von Daten Providern abhängig vom Ergebnis eines anderen Providers.

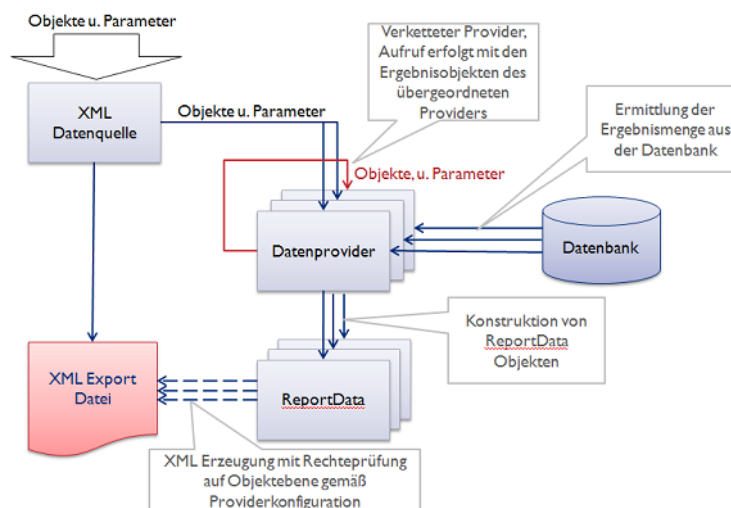


Abb. 2.10: Excel: PowerReports - Funktionsprinzip XML Export

2.4.3 Datenquellen und Datenprovider

PowerReports beziehen ihre Daten aus XML Datenquellen, die im System unter *Administration/Konfiguration* → *Konfiguration* → *XML Datenquellen* konfiguriert werden.

XML Datenquellen können kontextfreie oder kontextbezogene Daten liefern und werden grundsätzlich anhand dieser Eigenschaft in zwei Typen unterschieden:

1. *Kontextbezogene Datenquellen*

Kontextbezogene Datenquellen liefern beispielsweise Daten zu einem Projekt, EC oder einem Artikel. Zur Auswertung kontextbezogener XML Datenquellen wird also immer mindestens ein Objekt benötigt, das den Kontext bildet. Beispielsweise Aufgaben und Offene Punkte zum Projekt. Das Projekt bildet in diesem Beispiel den Kontext. Reports, die auf kontextbezogenen Datenquellen basieren, werden immer über das Kontextmenü des kontextbildenden Objekts aufgerufen. Ein Report 'Projektübersicht', der Detailinformationen zu einem Projekt enthält, wird also im Kontextmenü eines Projektes aufgerufen.

2. Kontextfreie Datenquellen

Kontextfreie Daten sind beispielsweise Abfragen über den gesamten Datenbestand. Zur Auswertung kontextfreier Datenquellen wird kein kontextbildendes Objekt benötigt. Reports, die auf kontextfreien Datenquellen basieren, werden an zentraler Stelle in der OfficeLink-Menüleiste unter *Einstellungen* → *Weitere Einstellungen...* → *Reports* aufgerufen.

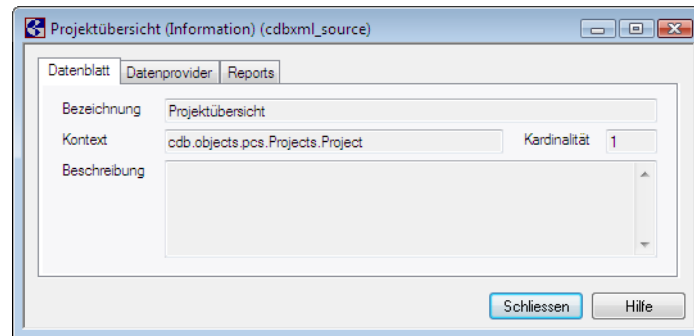


Abb. 2.11: Excel: PowerReports - Datenquelleneigenschaften

Bezeichnung Eindeutiger Name einer XML Datenquelle.

Kontext Legt fest, ob die Datenquelle kontextfrei oder kontextbezogen ist. Für kontextfreie Datenquellen bleibt dieses Feld leer. Für kontextbezogene Datenquellen wird der Objekttyp, in dessen Kontext die Datenquelle ausgewertet werden soll, in Form eines voll qualifizierten Pythonnamens der zugehörigen cdb.objects Klasse angegeben. Beispielsweise `cdb.objects.pcs.Projects.Project` für eine Datenquelle, die im Kontext von Projekten ausgewertet werden kann. Bei der Neuanlage einer Datenquelle stehen hier alle Root-Klassen in der Dropdown-Liste zur Verfügung. Bei der Verwendung von Python-Unterklassen muss der voll qualifizierte Pythonname jedoch manuell eingetragen werden.

Kardinalität Gibt die Kardinalität des Kontextes an. Für kontextfreie Datenquellen wird dieses Feld nicht ausgewertet. Für kontextbezogene Datenquellen legt die Kardinalität fest, ob die Datenquelle mit genau einem kontextbildenden Objekt (Single Select) oder mit einer beliebigen Anzahl kontextbildender Objekte (Multi Select) ausgewertet werden kann. Mögliche Werte sind 1 oder N.

Beschreibung Beschreibung der Datenquelle.

Datenprovider

Die Daten einer XML Datenquelle werden durch zugeordnete Datenprovider geliefert. Eine Datenquelle kann aus einer beliebigen Anzahl von Daten Providern bestehen. Datenprovider sind parametrisierbar und wiederverwendbar.

Datenprovider werden in zwei Gruppen unterschieden:

1. Universelle Datenprovider

Universelle Datenprovider sind vorgefertigte Datenproviderimplementierungen für häufige Anwendungsfälle, die ohne Programmierung in beliebigen Datenquellen verwendet und in deren Kontext konfiguriert werden können. Folgende Typen universeller Datenprovider stehen zur Verfügung:

- *Relationship*

zur Navigation von Beziehungen des cdb.objects Frameworks

- *Rule*
für konfigurierte komplexe Abfragen
- *SimpleQuery*
für einfache interaktive Abfragen
- *GroupBy*
zur Gruppierung der Ergebnismenge anderer Provider

2. Spezialisierte Datenprovider

Spezialisierte Datenprovider sind kundenspezifische Providerimplementierungen für spezielle Anwendungsfälle. Spezialisierte Datenprovider sind ebenfalls in beliebigen Datenquellen wiederverwendbar. Die Implementierung erfolgt durch Ableitung von einer vordefinierten Basisklasse und Implementierung einer Schnittstelle. Spezialisierte Datenprovider sind immer vom Typ `CustomCode`.

Kontextbezogene Datenquellen enthalten automatisch einen speziellen Datenprovider, der das kontextbildende Objekt bzw. die kontextbildenden Objekte liefert. Dieser Datenprovider wird im Folgenden als Kontextprovider bezeichnet. Die folgende Abbildung zeigt den Kontextprovider anhand der XML Datenquelle 'Projektübersicht'. Das Projekt, auf dem ein zugrundeliegender Report aufgerufen wird, ist somit automatisch in den exportierten XML Daten enthalten und kann im Report dargestellt werden.

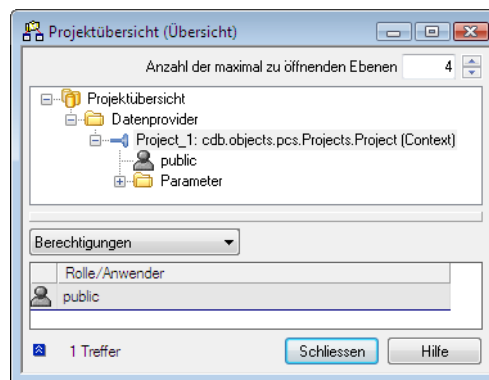


Abb. 2.12: Excel: PowerReports - Kontextprovider eines projektbezogenen Reports

Datenprovider können hierarchisch angeordnet werden. Ein untergeordneter Datenprovider arbeitet dabei immer mit der Ergebnismenge des übergeordneten Datenproviders. Die Ergebnismenge des übergeordneten Providers dient also als Eingabeparameter für den untergeordneten Provider. Auf diese Weise lassen sich beispielsweise die Ergebnisse des übergeordneten Providers mit einem GroupBy Provider gruppieren oder Beziehungen durch das Objektmodell navigieren.

Folgende Abbildung zeigt wieder das Beispiel 'Projektübersicht', erweitert um einen Relationship Provider, der die Dokumente zum Projekt liefert. Der Kontextprovider dient dabei als übergeordneter Provider und stellt damit das Startobjekt (das Projekt) zur Navigation der Beziehung.

Datenprovider Eigenschaften

- *XML Datenquelle*
Fremdschlüssel der zugehörigen XML Datenquelle. Automatisch vorbelegt und nicht änderbar.
- *Typ*
Der Typ des Datenproviders (siehe *Providertypen im Detail* (Seite 14)).
- *Provider*
Abhängig vom Providertyp spezifiziert dieses Feld die Datenherkunft (siehe *Providertypen im Detail* (Seite 14)).

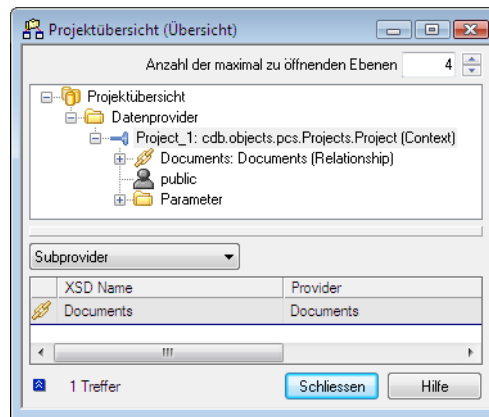


Abb. 2.13: Excel: PowerReports - Relationship Subprovider zum Kontextprovider

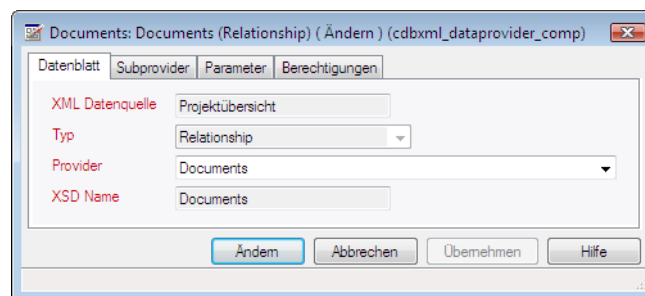


Abb. 2.14: Excel: PowerReports - Datenprovider Eigenschaften

- *XSD Name*

Ein im Kontext der Datenquelle eindeutiger Namensraum. Abhängig vom Providertyp auch automatisch vorbelegt. Nach der Neuanlage eines Providers nicht mehr veränderbar.

Providertypen im Detail

- *Relationship*

Datenprovider vom Typ Relationship liefern Objekte, die durch Navigation einer Beziehung des cdb.objects Frameworks ermittelt werden. Da zur Navigation einer Beziehung immer ein Startobjekt benötigt wird, können Relationship Provider ausschließlich als Subprovider verwendet werden. Relationship Provider können allen Providern zugeordnet werden, die cdb.objects Objekte als Ergebnis liefern.

Im Feld `Provider` wird der Name einer Beziehung eingetragen, die ausgehend von dem Ergebnisobjekt des übergeordneten Providers navigierbar ist. Die jeweils verfügbaren Beziehungen stehen bei der Neuanlage eines Relationship Providers in einem Katalog zur Auswahl. Insofern es sich jedoch bei der Datenquelle des übergeordneten Providers nur um eine Python-Unterklasse handelt, dann werden die verfügbaren Beziehungen nicht automatisch aufgelistet, sondern müssen manuell eingetragen werden.

Durch Verkettung von Relationship Provider können ausgehend von dem Startobjekt Daten aus dem gesamten Objektmodell ermittelt werden. Folgendes Beispiel zeigt die Verwendung von Relationship Provider ausgehend von einem Projekt. Unter anderem werden auch das übergeordnete Projekt und alle zu diesem Projekt gehörigen Dokumente ermittelt.

Die Verwendung eines Relationship Providers an einem übergeordneten Provider, der ein n-stelliges Ergebnis liefert, führt zu einem Multi-Export, d.h. die Beziehung wird für jedes Ergebnisobjekt des übergeordneten Providers einzeln navigiert und gemeinsam mit dem Startobjekt exportiert. Für Datenquellen mit Multi-Export Konfigurationen ist ein spezielles Template-Sheet in der Excel Vorlage erforderlich (siehe *Darstellung komplexer Zeilen bei Multi-Export Datenquellen* (Seite 40)).

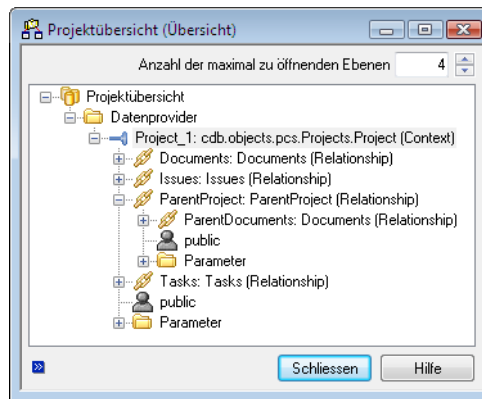


Abb. 2.15: Excel: PowerReports - Beziehungen als Provider

Die Ergebnismenge eines Relationship Providers ist in Abhängigkeit von der Kardinalität der Beziehung entweder ein `ReportData` Objekt oder eine `ReportDataList`.

- *Rule*

Datenprovider vom Typ Rule liefern Objekte durch Auswertung einer konfigurierten Regel, die die Objekte deklarativ beschreibt. Mit konfigurierten Regeln lassen sich sowohl einfache als auch komplexe Abfragen unter Einbeziehung von Eigenschaften referenzierter Objekte ausführen. Detaillierte Informationen zu konfigurierbaren Regeln finden Sie im Administratorhandbuch im Abschnitt 'Konfigurierbare Regeln'.

Rule Provider können ohne übergeordneten Provider oder als Subprovider verwendet werden. In beiden Fällen kann die Regel durch Verwendung von Variablen parametrisiert werden. Die Variablen können durch einen vorgeschalteten Dialog vom Anwender abgefragt werden. Dazu im Folgenden ein Beispiel für einen regelbasierten Provider ohne übergeordneten Provider. Wenn eine Regel mit dem Namen *My Projects: By Category* Projekte mit einer über einen Reportdialog definierten Projektkategorie liefern soll und zu diesem Zweck der Term `category=$(category)` definiert wird, dann muss in dem Reportdialog ein Attribut mit dem Namen `myprojectsbycategory-category` eingebaut werden. Da der Variablenname `$(category)` in diesem Beispiel einem tatsächlichen Objektattributnamen entspricht, kann dieser variable Regelfilter zusätzlich in dem Report selbst per automatisch im XML-Schema erscheinendem Argumentattribut mit einer Zelle verlinkt und aktualisiert werden. Bei Verwendung als Subprovider wird versucht, die Variablen aus den Attributen des Ergebnisobjektes des übergeordneten Providers zu belegen, sofern diese nicht durch einen ggf. vorgeschalteten Dialog definiert sind. Ein Beispiel dazu für einen Term, der für einen regelbasierten Subprovider Aufgaben zu dem übergeordneten Projekt Provider liefern soll, wäre demnach `cdb_project_id=$(cdb_project_id)`.

Im Feld 'Provider' wird bei Rule Providern der Name der zu verwendenden Regel eingetragen.

Die Sortierung der Ergebnismenge kann durch den Parameter *OrderBy* festgelegt werden. Die Definition der Sortierung erfolgt durch kommasetrennte Auflistung der Datenbankattribute, nach denen sortiert werden soll. Für eine absteigende Sortierung wird dem jeweiligen Attribut ein Minuszeichen vorangestellt.

Folgende Abbildung zeigt eine Datenquelle mit zwei Rule Providern zur Ermittlung von zu bearbeitenden Projektaufgaben und unerledigten Offenen Punkten, für die der angemeldete Anwender verantwortlich ist:

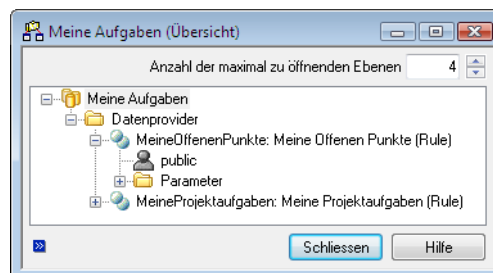


Abb. 2.16: Excel: PowerReports - Provider zur regelbasierten Ermittlung von Aufgaben

Die Definition der Regel zur Ermittlung der Offenen Punkte könnte beispielsweise wie folgt aussehen:

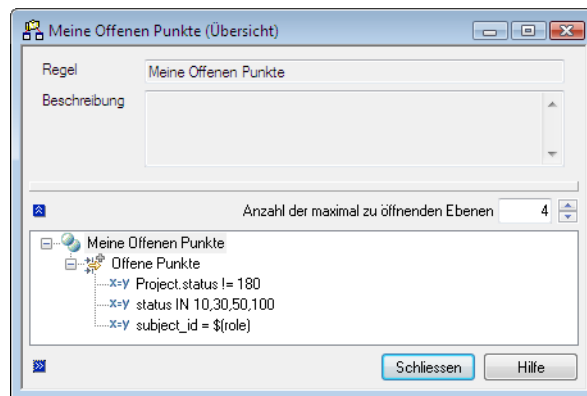


Abb. 2.17: Excel: PowerReports - Konfigurierte Regel zur Ermittlung der Offenen Punkte eines Anwenders

Die Ergebnismenge von Rule Providern ist immer eine *ReportDataList*.

- *SimpleQuery*

Datenprovider vom Typ SimpleQuery liefern Objekte durch Abfragen auf einer Datenbanktabelle, ähnlich einer System-Suche. SimpleQuery Provider werden kontextfrei verwendet, das heißt sie werden nicht als Subprovider verwendet, sondern immer Top Level zur Datenquelle.

Im Feld 'Provider' wird eine *cdb.objects* Klasse eingetragen, die den Typ der zu liefernden Objekte festlegt. Die *cdb.object* Klasse legt damit auch die Datenbanktabelle fest, auf der die Suche ausgeführt wird.

SimpleQuery Provider liefern ohne einschränkende Suchbedingungen den gesamten Datenbestand der jeweiligen Datenbanktabelle. Die zu verwendenden Suchbedingungen werden bei SimpleQuery Providern durch einen vorgeschalteten Reportdialog vom Anwender abgefragt. Dieser Dialog kann für jeden auf einer Datenquelle basierenden Report andere Suchbedingungen zur Auswahl stellen oder auch fest vorgeben. Unabhängig von den verwendeten Suchbedingungen kann die maximale Ergebnismenge mit dem Parameter *MaxRows* beschränkt werden. Wenn diese Menge überschritten wird, dann kommt es zu einer Fehlermeldung und der Report wird nicht generiert.

Die Sortierung der Ergebnismenge kann durch den Parameter *OrderBy* festgelegt werden. Die Definition der Sortierung erfolgt durch kommasetrennte Auflistung der Datenbankattribute, nach denen sortiert werden soll. Für eine absteigende Sortierung wird dem jeweiligen Attribut ein Minuszeichen vorangestellt.

Folgende Abbildung zeigt eine Datenquelle mit einem SimpleQuery Provider für Projekte mit einer definierten Sortierung und eine Begrenzung der Ergebnismenge auf 5000. Die übrigen dargestellten Parameter gelten auch für alle anderen Providertypen und werden in *Provider Parameter* (Seite 18) ausführlich beschrieben.

SimpleQuery Provider liefern immer eine *ReportDataList* als Ergebnis.

- *GroupBy*

GroupBy Provider gruppieren das Ergebnis anderer Provider anhand konfigurierbarer Attribute oder Wertebereiche. Sie werden immer als Subprovider zu Providern verwendet, die als Ergebnis eine *ReportDataList* liefern. Hierbei können für alle numerischen Attribute des übergeordneten Providers Gruppenfunktionen (sum, average, min, max, count) definiert werden. Die Ergebnistabellen von GroupBy Providern sind damit besonders gut zur Erstellung von Diagrammen geeignet.

Die folgende Abbildung zeigt exemplarisch die Gruppierung von Projekten, die durch einen übergeordneten SimpleQuery Provider geliefert werden. Die Definition der Kriterien zur Gruppierung erfolgt durch die Parameter *Group_By1*, *Group_By2* und *Group_By3*. Im einfachsten Fall werden diese Parameter mit Attributnamen belegt, nach denen gruppiert werden soll. Die Definition von Gruppenfunktionen erfolgt durch optionale Parameter *Function1*, *Function2*, usw. Diese werden jeweils mit einem durch Doppelpunkt getrennten Wertepaar aus Gruppenfunktion und Attributname belegt, z.B. mit *count:** oder *average:effort_plan* (die Gruppenfunktion *count* steht hierbei standardmäßig zur Verfügung, d.h. diese muss nicht explizit als

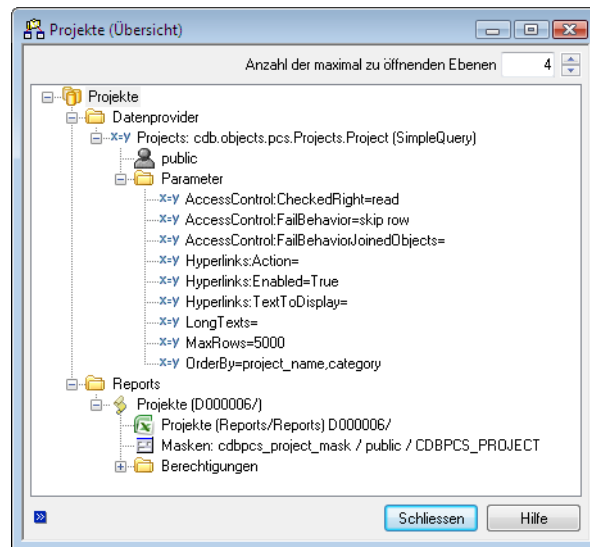


Abb. 2.18: Excel: PowerReports - SimpleQuery Provider

	group_by1	group_by2	group_by	count
2				
3	Projektkierung	IT		6
4	Vertrieb	Prototyping		1
5	Projektkierung	Prototyping		0
6	Organis./Rationa	IT		6
7	Entwicklung	Engineering		4
8	Organis./Rationa	Standardisation		0
9	Forschung	IT		9
10	Organis./Rationa	Engineering		1
11	Entwicklung	IT		6

Abb. 2.19: Excel: PowerReports - Ergebnistabelle eines GroupBy Providers

FunctionX des Providers definiert werden). Durch Hinzufügen mehrerer GroupBy Provider lassen sich die Ergebnisse des übergeordneten Datenproviders gleichzeitig nach unterschiedlichen Kriterien gruppieren. Jeder GroupBy Provider liefert dabei eine eigene Ergebnistabelle.

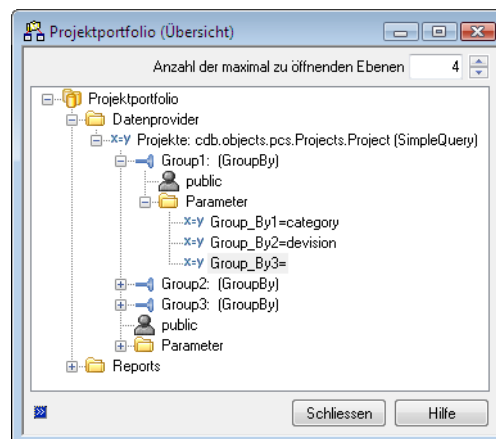


Abb. 2.20: Excel: PowerReports - Gruppierung von Projekten

Die Gruppierungskriterien GroupBy_1, GroupBy_2 und Group_By_3 können in einem vorgeschalteten Reportdialog auch interaktiv vom Anwender abgefragt werden. Anwender können auf diese Weise nach beliebigen Eigenschaften selbst gruppieren. Die Gruppierungskriterien aus dem Dialog haben Vorrang gegenüber den im Register *Parameter* definierten Kriterien.

Zur Gruppierung nach Wertebereichen, beispielsweise Projekte nach Budget, kann statt eines Attributnamen auch eine Liste von konfigurierten Regeln angegeben werden. Die Regeln beschreiben dabei Teilmengen, in die die Ergebnisobjekte des übergeordneten Providers einsortiert werden. Die Regeln werden, wie in der

folgenden Abbildung dargestellt, in Klammern als kommagetrennte Liste angegeben.

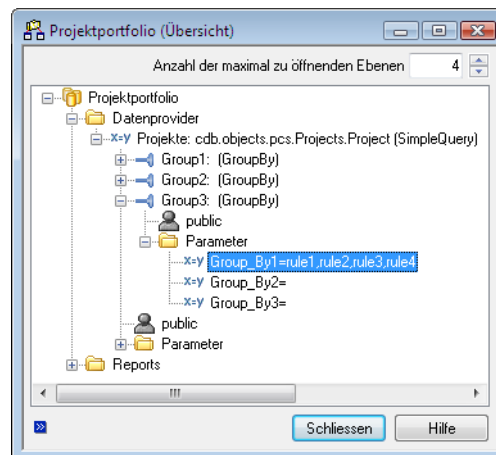


Abb. 2.21: Excel: PowerReports - Gruppierung nach Wertebereichen mit Regeln

Die in dem Beispiel verwendeten Gruppierungsregeln geben Wertebereiche für das Projektbudget an. Die Definition dieser Regeln könnte beispielsweise wie folgt aussehen. Weitere Informationen zu konfigurierbaren Regeln finden Sie im Administratorhandbuch im Abschnitt 'Konfigurierbare Regeln'.

Regel	Beschreibung
rule1	Budget <100000 Report Gruppierungsregel
rule2	Budget 100000-500000 Report Gruppierungsregel
rule3	Budget 500000-1000000 Report Gruppierungsregel
rule4	Budget >=1000000 Report Gruppierungsregel

Abb. 2.22: Excel: PowerReports - Gruppierungsregeln am Beispiel Projekte nach Budget

- *CustomCode*

Datenprovider vom Typ CustomCode können Daten auf beliebige Art und Weise ermitteln und zurückliefern. CustomCode Provider können beispielsweise Abfragen ausführen, berechnete Spalten hinzufügen oder auch Daten aus externen Datenbanken liefern. CustomCode Provider können die Reportdaten aus cdb.objects Objekten oder RecordSets konstruieren. Darüberhinaus können die Reportdaten auch völlig frei konstruiert werden. Die Implementierung von CustomCode Providern und eine ausführliche API Beschreibung finden Sie in *Implementierung eigener Provider* (Seite 21) .

- *Context*

Datenprovider vom Typ Context werden ausschließlich automatisch vom System bei der Neuanlage einer kontextbezogenen Datenquelle angelegt. Eine kontextbezogene Datenquelle enthält immer genau einen Datenprovider vom Typ Kontext. Dieser Provider liefert das Objekt bzw. die Objekte, auf denen der Report aufgerufen wurde. An den Kontextprovider können beispielsweise Subprovider vom Typ Relationship angehängt werden, um referenzierte Objekte in den Report einzubeziehen.

In Abhängigkeit von der Kardinalität der Datenquelle liefert ein Kontextprovider entweder einer Liste von Objekten als *ReportDataList* oder ein einzelnes Objekt als *ReportData* Objekt.

Provider Parameter

Datenprovider sind durch Parameter konfigurierbar. Die verfügbaren Parameter hängen vom jeweiligen Providertyp ab und werden bei der Neuanlage von Providern automatisch mit Standardwerten hinzugefügt. Die in diesem Abschnitt beschriebenen Parameter gelten für alle Provider, die Daten in Form von cdb.objects Objekten liefern. Dies gilt auch für selbst implementierte Provider vom Typ CustomCode.

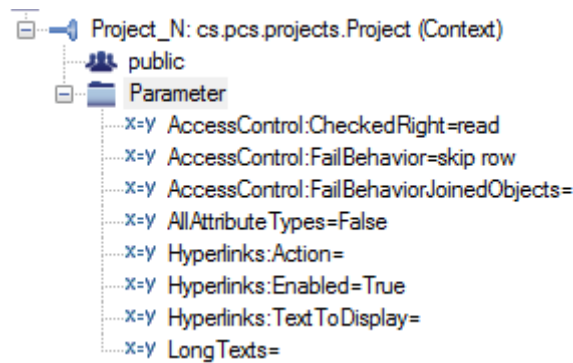


Abb. 2.23: Excel: PowerReports - Datenprovider Standardparameter

Rechteprüfung auf Objektebene Die Rechteprüfung auf Objektebene kann für alle Datenprovider, die cdb.objects-Objekte als Ergebnismenge liefern, mit folgenden Parametern konfiguriert werden.

- *AccessControl:CheckedRight*

Legt das pro Objekt zu prüfende Recht fest. Ist dieser Parameter nicht vorhanden oder leer, ist die Rechteprüfung auf Objektebene deaktiviert. Verwendet werden können alle im Rechtesystem verfügbaren Rechte und Rechtegruppen. Standardmäßig wird bei Neuanlage eines universellen Providers automatisch das read-Recht eingetragen. Bei großen Datenmengen, wie sie beispielsweise bei Providern vom Typ SimpleQuery anfallen können, verschlechtert sich bei aktivierter Rechteprüfung das Laufzeitverhalten. Sofern nicht unbedingt erforderlich, sollte die Rechteprüfung auf Objektebene bei potentiell großen Datenmengen deaktiviert werden.

- *AccessControl:FailBehavior*

Legt das Verhalten bei negativer Rechteprüfung fest. Zur Auswahl stehen zwei Modi:

- *obfuscate*

Die Objektattribute werden vor dem Export unkenntlich gemacht. Die Schlüsselattribute und weitere hinzugefügte Daten, beispielsweise berechnete Spalten, bleiben leserlich.

- *skip row*

Das Objekt und ggf. weitere hinzugefügte Daten werden nicht exportiert.

Hyperlinks Datenprovider können konfigurativ Hyperlinks exportieren, mit denen aus Excel direkt auf die entsprechenden Objekte im System verlinkt werden kann. Die Konfiguration von Hyperlinks erfolgt mit den nachfolgend beschriebenen Parametern:

- *Hyperlinks:Enabled*

Aktiviert Hyperlinks für den Datenprovider. Es wird ein zusätzliches Attribut cdbxml_hyperlink exportiert. Dieses kann in der Excel Datenbindung verwendet werden und verlinkt damit automatisch aus Excel auf das entsprechende System-Objekt. Zur Aktivierung von Hyperlinks ist es hinreichend, diesen Parameter mit 'true' zu belegen. Werden die nachfolgend beschriebenen weiteren Parameter nicht belegt, werden Hyperlinks mit der Aktion CDB_ShowObject (Anzeige des Objekts im Infodialog) generiert, die mit der im Klassenverzeichnis konfigurierten Objektbeschreibung beschriftet werden.

- *Hyperlinks:Action*

Legt die auszuführende System-Operation beim Click auf den Hyperlink fest. Verwendet werden können alle für eine Objektklasse definierten Operationen. Dies umfasst Systemoperationen wie beispielsweise CDB_Modify und CDB_View, PowerScript Operationen und GraphView Operationen wie beispielsweise cdbpcs_project_overview zur Darstellung der Projektübersicht. Wird dieses Feld leer gelassen, wird die Klassenvariable __default_action__ der zugrundeliegenden cdb.objects Klasse ausgewertet. Ist diese ebenfalls nicht definiert, wird die Operation CDB_ShowObject verwendet.

- *Hyperlinks:TextToDisplay*

Legt den anzuzeigenden Text fest. Es kann eine beliebige Zeichenkette für einen konstanten Ausdruck oder ein Attributname der zugrundeliegenden cdb.objects-Klasse angegeben werden. Wird dieses Feld leer gelassen, wird die konfigurierte Objektbeschreibung aus dem System-Klassenverzeichnis verwendet.

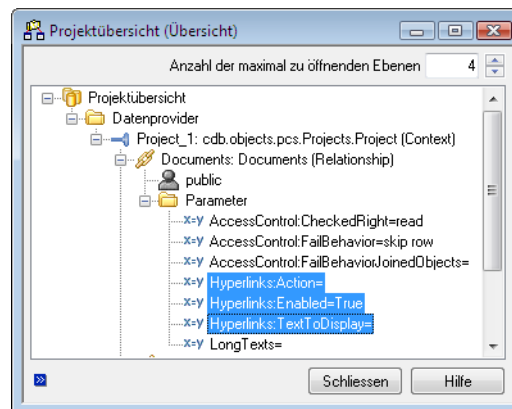


Abb. 2.24: Excel: PowerReports - Hyperlinkkonfiguration und zugehöriges Schemaobjekt

Langtexte Datenprovider können optional die zu einem Objekt gehörigen Langtexte exportieren. Die Angabe der zu exportierenden Langtexte erfolgt aus Performancegründen explizit durch den Parameter *LongTexts*. Die zu exportierenden Langtexte werden durch eine kommasetrennte Aufzählung der Langtextrelationen angegeben. Die eingetragenen Langtexte stehen anschließend in Excel zur Herstellung der Datenbindung unter diesen Namen zur Verfügung.

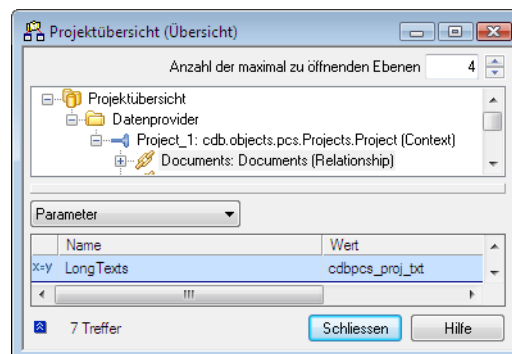


Abb. 2.25: Excel: PowerReports - Langtextkonfiguration und zugehöriges Schemaobjekt

Attributtypen Datenprovider exportieren standardmäßig neben allen normalen Attributen auch immer alle Mapped Attribute. Jedoch muss das Ausleiten weiterer spezieller Attribute, wie das der Joined und Virtual Attribute, für jeden Provider explizit aktiviert werden. Die geschieht, indem man den Providerparameter *AllAttributeTypes* auf *True* setzt.

Wichtig: Das Verwenden von Joined und Virtual Attributen ist aus Gründen der Performance nicht für Provider empfehlenswert, welche viele Objekte liefern müssen.

MaxRowsTruncate Die Ergebnismenge eines Providers kann mit diesem Parameter limitiert werden. Der Standardwert hierfür liegt aufgrund einer Beschränkung in Excel bei 64000. Falls eine Liste beim Füllen einer Tabelle die Tabellenzeilennummer 65535 überschreitet, kommt es zu einer XML-Import Fehlermeldung in Excel, dass keine oder nur einige Daten importiert wurden.

Generelle Zugriffsberechtigungen auf Providerebene

Der Zugriff auf die Daten eines Datenproviders wird durch Zuordnung von berechtigten Rollen und Personen geschützt. Bei der Neuanlage eines Datenproviders wird automatisch die Rolle 'public' eingetragen, so dass auf Providerebene zunächst keine Zugriffsbeschränkungen bestehen.

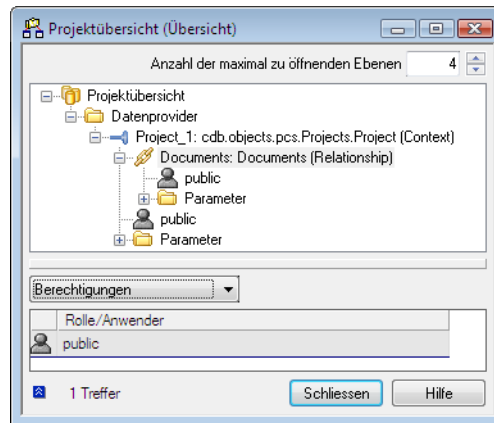


Abb. 2.26: Excel: PowerReports - Generelle Zugriffsberechtigungen auf Providerebene

Zusätzlich zu den Zugriffsbeschränkungen auf Providerebene führen Provider beim Datenexport auch eine Rechteprüfung auf Objektebene anhand des Rechtesystems durch. Die Rechteprüfung auf Objektebene ist für jeden Provider aktivierbar/deaktivierbar. Das geprüfte Recht ist konfigurierbar.

2.4.4 Implementierung eigener Provider

Eigene Datenprovider können in einem beliebigen Python Kundenmodul abgelegt werden. Damit eigene Datenprovider zu einer Datenquelle zugeordnet werden können, müssen die jeweiligen Python Module in der dem jeweiligen Kundenmodul zugehörigen `default.conf` eingetragen werden. Befinden sich die Implementierungen eigener Datenprovider beispielsweise in dem Python Modul `my_namespace/my_module/PowerReportsCodeSamples.py`, so muss die Zeile `my_namespace.my_module.PowerReportsCodeSamples` zur Datei `my_namespace/my_module/default.conf` hinzugefügt werden. Der Eintrag bewirkt, dass die enthaltenen Providerklassen bei der Zuordnung zur Datenquelle zur Auswahl stehen.

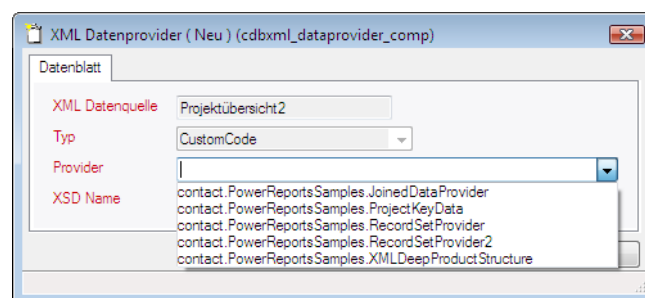


Abb. 2.27: Excel: PowerReports - Auswahlkatalog für eigene Providerimplementierungen

Die Implementierung eigener Datenprovider erfolgt durch Ableitung von der Basisklasse `contact.PowerReports.CustomDataProvider`.

```
from cdb.objects.PowerReports import *
class ProjectsProvider(CustomDataProvider):
    ...
```


Für einen Datenprovider müssen zwei Kardinalitäten angegeben werden. Die Kardinalität der Ergebnismenge und die Kardinalität für den Aufruf. Die Definition dieser Kardinalitäten erfolgt durch die zwei Klassenvariablen `CARD` und `CALL_CARD`.

- **CARD**

Gibt die Ergebniskardinalität eines Datenproviders an. Gültige Werte sind:

- **CARD_1**

Für Datenprovider, die ein einzelnes Datenobjekt als Ergebnismenge liefern.

- **CARD_N**

Für Datenprovider, die Listen als Ergebnismenge liefern.

- **CALL_CARD**

Die Aufrufkardinalität legt u.a. fest, ob ein Datenprovider die Ergebnismenge kontextfrei oder kontextbezogen ermittelt. Kontextbezogene Datenprovider werden immer als Subprovider zu einem übergeordneten Provider zugeordnet und erhalten zur Ermittlung der eigenen Ergebnismenge das Ergebnis des übergeordneten Providers als ersten Parameter.

- **CARD_0**

Für Datenprovider, deren Ergebnismenge kontextfrei ermittelt wird. Beispielsweise ein Datenprovider, der alle laufenden Projekte liefert. Datenprovider mit der Aufrufkardinalität `CARD_0` können nicht als Subprovider zugeordnet werden, sondern werden immer Top-Level zur Datenquelle zugeordnet.

- **CARD_1**

Für Datenprovider, die zur Ermittlung der eigenen Ergebnismenge genau ein Datenobjekt des übergeordneten Providers benötigen. Beispielsweise ein Datenprovider, der zu einem Projekt den Projektfortschritt und weitere Kennzahlen berechnet. Datenprovider mit der Kardinalität `CARD_1` werden immer als Subprovider zu einem übergeordneten Provider zugeordnet. Liefert der übergeordnete Provider als Ergebnis eine Liste, wird der Subprovider mehrfach mit jedem Listenelement einzeln aufgerufen. Diese Kombination aus Ergebniskardinalität des übergeordneten Providers und Aufrufkardinalität des untergeordneten Providers erfordert die Erstellung eines besonderen Template Sheets in der Excel Vorlage. Weitere Informationen zur Erstellung derartiger Vorlagen finden Sie in [Reportvorlagen erstellen](#) (Seite 34) .

- **CARD_N**

Für Datenprovider, die zur Ermittlung der Ergebnismenge eine Ergebnisliste vom übergeordneten Datenprovider erwarten. Beispielsweise ein Provider, der über eine Liste von Projekten, die vom übergeordneten Provider geliefert wird, aggregierte Informationen liefert.

Folgendes Codebeispiel zeigt einen Datenprovider, der eine Liste von Projekten liefern soll und dafür keinen Kontext benötigt.

```
class ProjectsProvider(CustomDataProvider):
    CARD = CARD_N
    CALL_CARD = CARD_0
    ...
```

Im Gegensatz zum vorherigen Beispiel kann der im folgenden Codebeispiel dargestellte Datenprovider ausschließlich als Subprovider verwendet werden, da zur Ergebnisermittlung ein Ergebnisobjekt des übergeordneten Providers benötigt wird. Ein derartiger Provider könnte beispielsweise alle laufenden Projekte zu einem Kunden liefern. Der Kunde ist dabei das Ergebnis des übergeordneten Providers.

```
class ProjectsProvider(CustomDataProvider):
    CARD = CARD_N
    CALL_CARD = CARD_1
    ...
```

Aus den Kardinalitäten ergibt sich, wie Datenprovider miteinander verkettet werden können. Die Kardinalität der Ergebnismenge muss zur Aufrufkardinalität des Subproviders passen.

- *CARD 1* → *CALL_CARD 1*

Der übergeordnete Provider liefert beispielsweise ein Projekt, welches vom Subprovider als Parameter zur Berechnung seiner eigenen Ergebnismenge verwendet wird. Dieser könnte beispielsweise den Projektfortschritt und weitere Kennzahlen zum Projekt berechnen.

- *CARD N* → *CALL_CARD N*

Beispielsweise zur Berechnung aggregierter Daten über die Ergebnisliste des übergeordneten Providers. Der Subprovider kann keine Daten liefern, die sich auf einzelne Listenelemente des übergeordneten Ergebnisses beziehen, da dieser Zusammenhang beim Export und Import verloren geht und in Excel nicht darstellbar ist. In diesem Fall muss eine Multi-Export Konfiguration verwendet werden.

- *CARD N* → *CALL_CARD 1* (Sonderfall Multi-Export)

Der Subprovider wird für jedes Ergebnisobjekt des übergeordneten Providers einzeln aufgerufen und exportiert. Mit einer Multi-Export Konfiguration können geschachtelte Listen und komplexe Zeilen in Excel dargestellt werden. Die Excel Datenbindung für Daten, die via Multi-Export exportiert werden, muss bei der Vorlagengenerierung auf einem speziellen Template Sheet erfolgen. Weitere Informationen zur Erstellung derartiger Vorlagen finden Sie in [Darstellung komplexer Zeilen bei Multi-Export Datenquellen](#) (Seite 40).

Die Implementierung eines Datenproviders besteht im Wesentlichen aus zwei Methoden. Eine Methode zur Definition des XML Schemas und eine Methode zur Ermittlung der Ergebnismenge.

```
class ProjectsProvider(CustomDataProvider):
    CARD = CARD_N
    CALL_CARD = CARD_1
    def getSchema(self):
        ...
    def getData(self, parent_result, source_args, **kwargs):
        ...
```

getSchema(self)

Liefert das XML Schema zur Herstellung der Datenbindung in Excel. Das XML Schema muss zu den durch die Methode `getData` gelieferten Daten passen. Das Schema wird durch die Konstruktion eines Objekts vom Typ `XSDType` beschrieben. Als Konstruktorparameter wird die Kardinalität angegeben, die mit der Ergebniskardinalität des Providers identisch sein muss. Mit der Methode `add_attr` können anschließend zu dem Schemaobjekt beliebig viele Attribute hinzugefügt werden. Der Datentyp wird dabei als zweiter Parameter in Form eines `cdb.sqlapi` Datentypen angegeben. Die hinzugefügten Attribute muss es in der Datenbank nicht zwingend geben, sondern können auch durch den Provider berechnet sein, jedoch dürfen deren Attributnamen keine Großbuchstaben enthalten.

```
class ProjectsProvider(CustomDataProvider):
    CARD = CARD_N
    CALL_CARD = CARD_1
    def getSchema(self):
        schema = XSDType(self.CARD)
        schema.add_attr('cdb_project_id', sqlapi.SQL_CHAR)
        schema.add_attr('project_name', sqlapi.SQL_CHAR)
        schema.add_attr('calculated', sqlapi.SQL_INTEGER)
        ...
        return schema
```

Als zweiter Parameter kann im `XSDType` Konstruktor optional ein Relationsname oder eine `cdb.objects` Klasse angegeben werden. Das Schemaobjekt enthält dann automatisch alle in der Relation enthaltenen Attribute. Im Falle einer `cdb.objects` Klasse sind anschließend zusätzlich alle `MappedAttributes` im Schemaobjekt enthalten.

```
...
def getSchema(self):
    schema = XSDType(self.CARD, 'cdbpcs_project')
    schema.add_attr('calculated', sqlapi.SQL_INTEGER)
```

```
...
return schema
```

Basiert das Schema auf genau einer Datenbankrelation oder `cdb.objects` Klasse, kann `getSchema` auch direkt einen Relationsnamen oder eine `cdb.objects` Klasse zurückliefern. Das XML Schema wird in diesem Fall vollständig automatisch aus dem Datenbankschema erzeugt. Im Falle einer `cdb.objects` Klasse sind wiederum zusätzlich alle `MappedAttributes` im XML Schema enthalten.

```
...
def getSchema(self):
    return "cdbpcs_project"
```

Für zeilenweise kombinierte Daten kann das Schema mit dem Plus-Operator aus den zugrundeliegenden Typen konstruiert werden. Bei namensgleichen Attributen in den beteiligten Relationen muss als dritter Parameter ein Präfix angegeben werden, der die Eindeutigkeit der Attributbezeichner in dem XML Schema sicherstellt. Dieser Präfix ist auch in der Implementierung von `getData` bei der Konstruktion der Ergebnisobjekte zu verwenden.

Folgendes Beispiel zeigt die Konstruktion eines Schemas für Artikel, erweitert um das Schema für die zugehörige Projekte:

```
...
def getSchema(self):
    schema = XSDType(self.CARD, 'teile_stamm')
    schema += XSDType(self.CARD, 'cdbpcs_project', 'Project_')
    ...
    return schema
```

`getData(self, parent_result, source_args, **kwargs)`

Liefert die Reportdaten beim Aufruf eines Reports. Zur Ermittlung der Ergebnismenge werden dem Provider folgende Argumente übergeben:

parent_result Ergebnismenge des übergeordneten Providers. Ist der übergeordnete Provider der Kontextprovider, enthält `parent_result` die Objekte, auf denen der Report aufgerufen wurde. In Abhängigkeit von der Ergebniskardinalität des übergeordneten Providers ist `parent_result` entweder vom Typ `ReportDataList` oder `ReportData`. Für Top-Level Provider ist dieser Parameter immer `None`.

source_args Ein Dictionary mit alle Argumenten aus dem optional vorgeschalteten Reportdialog. Die Schlüssel des Dictionaries entsprechen den Attributnamen der zugrundeliegenden Maskenkonfiguration, jedoch bestehen sie in diesem Moment immer nur aus Kleinbuchstaben, selbst wenn sie in der Maske teilweise mit Grossbuchstaben (im Providernamen) definiert wurden. Die Attributnamen in der Maskenkonfiguration folgen zur Zuordnung der Parameter zu den Daten Providern dem Namensschema `<Providername>-<Attributname>`.

kwargs Alle Argumente aus dem optional vorgeschalteten Reportdialog, die durch die Namenskonvention `<Providername>- <Attributname>` an den Datenprovider adressiert sind. Die Adressierung an den Provider ist aus den Argumentnamen entfernt, so dass die Argumente direkt via Attributname verwendet werden können. Wie oben erwähnt, enthalten die Attributnamen an dieser Stelle stets nur Kleinbuchstaben.

Die Rückgabe der Reportdaten erfolgt in Abhängigkeit von der Ergebniskardinalität (CARD) des Providers. Für die Ergebniskardinalität `CARD_1` muss ein `ReportData` Objekt, für die Ergebniskardinalität `CARD_N` eine `ReportDataList` zurückgeliefert werden. Die Konstruktion dieser Ergebnisobjekte wird im Folgenden beschrieben.

ReportData

Ergebnisobjekt für Datenprovider mit der Ergebniskardinalität `CARD_1`.

Die Konstruktion von `ReportData` Objekten erfolgt immer durch Angabe des konstruierenden Providers als 1.Parameter. Es wird also immer `self` als ersten Parameter übergeben. Folgendes Beispiel konstruiert ein leeres `ReportData` Objekt und gibt dieses als Ergebnis zurück.

```
class SampleProvider (CustomDataProvider):
    CARD = CARD_1
    ...
    def getData(self, parent_result, source_args, **kwargs):
        return ReportData(self)
```

Der Zugriff auf die Elemente eines `ReportData` Objekts entspricht den üblichen Zugriffsmethoden auf Python Dictionaries. Verfügbar sind die Methoden `keys`, `values` und `items`. Das Setzen und Lesen einzelner Attribute erfolgt durch den Indexoperator.

```
def getData(self, parent_result, source_args, **kwargs):
    data = ReportData(self)
    data["cdb_project_id"] = "4711"
    data["project_name"] = "Test project"
    return data
```

Die Konstruktion von `ReportData` Objekten kann direkt aus einem `cdb.objects`-Objekt oder `sqlapi.Record` erfolgen. Hierfür wird das Objekt bzw. der Record als zweiter Parameter an den Konstruktor übergeben.

```
def getData(self, parent_result, source_args, **kwargs):
    project = pcs.Project.ByKeys('P000001')
    data = ReportData(self, project)
    return data
```

Zusätzlich können weitere beliebige Felder hinzugefügt werden.

```
def getData(self, parent_result, source_args, **kwargs):
    project = pcs.Project.ByKeys('P000001')
    data = ReportData(self, project)
    data["calculated_attribute"] = 42
    return data
```

`ReportData` Objekte, die aus `cdb.objects` Objekten konstruiert werden, können auch die zum Objekt gehörigen Langtexte als Ergebnis zurückliefern. Hierfür kann als dritter Parameter eine Liste von Langtextrelationen angegeben werden.

```
def getData(self, parent_result, source_args, **kwargs):
    project = pcs.Project.ByKeys('P000001')
    data = ReportData(self, project, ['cdbpcs_proj_txt'])
    return data
```

Für zeilenweise kombinierte Daten, beispielsweise Artikel mit zugehörigem Projekt, können `ReportData` Objekte mit dem Plus-Operator verknüpft werden. Bei namensgleichen Attributen in den beteiligten Relationen muss als dritter Parameter ein Präfix angegeben werden, der die Eindeutigkeit der Attributbezeichner sicherstellt. Dabei ist derselbe Präfix zu verwenden, der auch bei der Schemaerstellung in der Implementierung von `getSchema` verwendet wurde. Folgendes Beispiel zeigt die Erstellung eines `ReportData` Objektes aus einem Artikel mit anschließender Erweiterung um das zugehörige Projekt.

```
def getData(self, parent_result, source_args, **kwargs):
    item = pdd.Item.ByKeys('4711', '')
    data = ReportData(self, item)
    data += ReportData(self, item.Project, prefix="Project_")
    return data
```

ReportDataList

Ergebnisobjekt für Datenprovider mit der Ergebniskardinalität `CARD_N`.

Die Konstruktion von `ReportDataList` Objekten erfolgt immer durch Angabe des konstruierenden Providers als ersten Parameter. Folgendes Beispiel konstruiert eine leere `ReportDataList` und gibt diese als Ergebnis zurück.

```
class SampleProvider(CustomDataProvider):
    CARD = CARD_N
    ...
    def getData(self):
        return ReportDataList(self)
```

`ReportDataList` Objekte bestehen aus einer Liste von `ReportData` Objekten und bieten zum Zugriff auf Listenelemente die üblichen Zugriffsmethoden für Python Listen: `append`, `count`, `extend`, `index`, `insert`, `pop`, `remove`, `reverse` und `sort`. Darüberhinaus können alle Listenoperatoren verwendet werden. Folgendes Beispiel zeigt, wie mit dem Plus-Operator zwei Listenelemente hinzugefügt werden, die jeweils ein Projekt enthalten.

```
def getData(self, parent_result, source_args, **kwargs):
    data = ReportDataList(self)
    data += ReportData(self, pcs.Project.ByKeys('P000001'))
    data += ReportData(self, pcs.Project.ByKeys('P000002'))
    return data
```

Mit den Plus-Operator kann als verkürzte Schreibweise auch direkt ein `sqlapi.Record` oder `cdb.objects` Objekt hinzugefügt werden, ohne erst selbst ein `ReportData` Objekt zu konstruieren.

```
def getData(self, parent_result, source_args, **kwargs):
    data = ReportDataList(self)
    data += pcs.Project.ByKeys('P000001')
    data += pcs.Project.ByKeys('P000002')
    return data
```

`ReportDataList` Objekte können direkt aus einer Liste von `cdb.objects` Objekten oder einem `sqlapi.RecordSet2` konstruiert werden. Dabei werden automatisch für jedes Listenelement `ReportData` Objekte konstruiert und hinzugefügt.

```
def getData(self, parent_result, source_args, **kwargs):
    projects = pcs.Project.Query("status>0")
    data = ReportDataList(self, projects)
    return data
```

Als dritter Parameter kann bei der Konstruktion aus `cdb.objects` Objekten eine Liste von einzubeziehenden Langtextrelationen angegeben werden.

```
def getData(self, parent_result, source_args, **kwargs):
    projects = pcs.Project.Query("status>0")
    data = ReportDataList(self, projects, ['cdbpcs_proj_txt'])
    return data
```

Bei der Konstruktion von Ergebnislisten ist zu berücksichtigen, dass die Listenelemente nicht heterogen sein dürfen. Es ist also beispielsweise nicht möglich, Projekte und Dokumente gemischt in einer `ReportDataList` zurückzugeben.

getClass(self)

Gibt die `cdb.objects` Klasse der gelieferten Daten zurück, wenn der Provider Daten in Form von `cdb.objects` Objekten liefert. Werden die Daten nicht aus `cdb.objects` Objekten konstruiert, wird diese Methode nicht implementiert. Folgendes Beispiel zeigt die Implementierung für einen Provider, der Projekte als Ergebnis liefert.

```
def getClass():
    return pcs.Projects.Project
```

Liefert ein eigener Provider kombinierte Daten, beispielsweise Artikel mit Projekt, so ist der Typ des Primärobjekts anzugeben. Das Primärobjekt ist das initiale Objekt, dass zur ReportData Konstruktion verwendet wurde.

```
def getData(self, parent_result, source_args, **kwargs):
    item = pdd.Item.ByKeys('4711', '')
    data = ReportData(self, item) # <- Primary object
    data += ReportData(self, item.Project, prefix="Project_")
    return data
def getClass(self):
    return pdd.Item # <- Primary object
```

Die Implementierung von `getClass` ermöglicht die Verwendung von Relationship-Providern als Subprovider an eigenen Providern. Eigene Provider lassen sich somit direkt in das Objektmodell des `cdb.objects` Frameworks integrieren. Liefert beispielsweise ein eigener Provider Projekte, so können alle von Projekten ausgehende Beziehungen als Subprovider an den eigenen Provider angehängt werden. Auf diese Weise können also zu den selbst ermittelten Projekten beispielsweise Offenen Punkte, Aufgaben oder Dokumente ohne weitere Programmierung in die Datenquelle integriert werden.

Darüberhinaus erkennt das Report Framework an dieser Methode `cdb.objects` basierte Provider, so dass bei fehlender Implementierung beispielsweise die in *Provider Parameter* (Seite 18) beschriebenen Parameter nicht automatisch angelegt werden.

getDefaultParameters(self)

Gibt ein Dictionary mit Parameternamen und Defaultwerten zurück, die bei der Neuanlage eines Providers dieses Typs automatisch angelegt werden und anschließend im Register 'Parameter' des Providers angepasst werden können. Die durch diese Methode zurückgelieferten Parameternamen können in der Implementierung des Providers verwendet werden, um den Provider im Verwendungskontext konfigurierbar zu machen. Auf diese Art und Weise lassen sich universelle und konfigurierbare Datenprovider selbst implementieren.

```
def getDefaultParameters():
    return {"My Parameter" : "My Value"}
```

`cdb.objects` basierte Datenprovider erhalten zusätzlich automatisch immer folgende Defaultparameter, sofern die `getClass` Methode implementiert wurde.

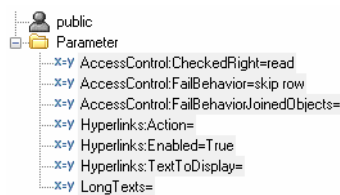


Abb. 2.28: Excel: PowerReports - Defaultparameters

Die Auswertung dieser Parameter erfolgt automatisch und muss für eigene Provider nicht selbst implementiert werden.

Der Zugriff auf eigene Parameter eines Providers erfolgt durch die Methode `getParameter(name)`.

```
def getData(self, parent_result, source_args, **kwargs):
    param = self.getParameter("My Parameter")
    if param == "My Value":
        ...
    else:
        ...
    ...
```

getArgumentDefinitions(self)

Die Ergebnismenge von Datenprovidern kann durch einen vorgeschalteten Reportdialog beeinflusst werden (siehe auch *Reportdialoge erstellen* (Seite 36)). Üblicherweise werden dabei zusätzliche einschränkende Suchbedingungen interaktiv vom Anwender abgefragt. Damit auch zu einem späteren Zeitpunkt nachvollziehbar ist, welche Daten in einem Report dargestellt werden, welche Abfrageparameter also zu dem speziellen Ergebnis geführt haben, ist es häufig sinnvoll, die verwendeten Abfrageparameter in dem Report mit darzustellen.

Zu diesem Zweck kann die Methode *getArgumentDefinitions* implementiert werden. Die Methode gibt ein Dictionary mit Parameternamen und den zugehörigen Datentypen zurück. Die zurückgelieferten Argumentnamen werden automatisch mit dem angegebenen Datentyp in das XML Schema unter dem Schemanamen *Arguments* hinzugefügt und beim Datenexport mit exportiert.

Die Argumentnamen werden dabei automatisch um den Providernamen erweitert, so dass eine eindeutige Zuordnung zum Datenprovider sichergestellt ist (analog zur Dialogkonfiguration wie in *Reportdialoge erstellen* (Seite 36) beschrieben).

Wichtig: Die Attributnamen dürfen keine Großbuchstaben enthalten.

Die Verwendung im Detail am Beispiel eines Reports über Projekte mit einem vorgeschalteten Dialog zur optionalen Einschränkung auf eine bestimmte Kategorie (der Dialog enthält das Attribut *projekte-category*, ermöglicht also die Eingabe einer Kategorie, die von dem Datenprovider *projekte* ausgewertet wird):

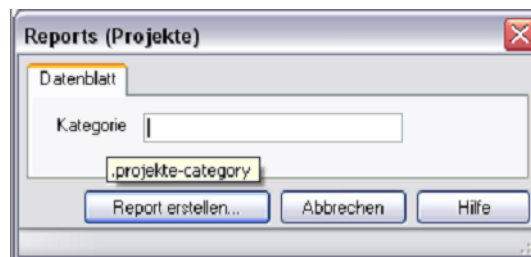


Abb. 2.29: Excel: PowerReports - Reportdialog am Beispiel Abfrage der Projektkategorie

Die im folgenden Codebeispiel dargestellte Implementierung des Datenproviders wertet die abgefragte Kategorie aus und schränkt so die Ergebnismenge ein. Die Methode *getArgumentDefinitions* liefert das Attribut *category* und teilt dem PowerReports Framework damit mit, dass dieses Argument mit exportiert werden soll.

```
class Projekte(CustomDataProvider):
    def getSchema(self):
        return "cdbpcs_project"
    def getData(self, parent_result, source_args, **kwargs):
        category = kwargs["category"] # Parameter from dialog
        cond = ""
        if category:
            cond = "category='%s'" % category
        return ReportDataList(self, pcs.Projects.Project.Query(cond))
    def getArgumentDefinitions(self):
        return {"category" : sqlapi.SQL_CHAR}
```

Aufgrund der obigen Implementierung von *getArgumentDefinitions* ist das in dem Reportdialog verwendete Attribut unter *Arguments* verfügbar und kann in der Datenbindung der Excel Vorlage verwendet werden.

Nach Herstellung der Datenbindung enthalten Reports die verwendete Suchbedingung für die Projektkategorie.

Die Suchbedingung kann direkt in dem Report geändert werden und anschließend die Kopplungsfunktion *Report aktualisieren* aufgerufen werden.

Der Report wird mit den geänderten Suchbedingungen aktualisiert, ohne diesen neu aus dem System aufrufen zu müssen.



Abb. 2.30: Excel: PowerReports - XML Schema für Argumente

	A	B	C	D
1	Kategorie	Entwicklung		
2				
3	cdb project id	cdb status txt	description	category
4	CON-PCS-P000002	in Bearbeitung	Demo-Daten für das Projektmanagem	Entwicklung
5	CON-PCS-P000009	angelegt	Projektvorlage Automotivebereich	Entwicklung
6	CON-PCS-P000010	in Bearbeitung	Projekt Werkzeugentwicklung	Entwicklung
7	CON-PCS-P000011	verworfen	Projektbeispiel Werkzeugentwicklung	Entwicklung
8	CON-PCS-P000012	in Bearbeitung	Automotivebereich	Entwicklung
9	CON-PCS-P000015	in Bearbeitung	Projekt Werkzeugentwicklung	Entwicklung
10	CON-PCS-P000016	in Bearbeitung	Projekt Werkzeugentwicklung	Entwicklung
11	CON-PCS-P000017	in Bearbeitung	Projekt Werkzeugentwicklung	Entwicklung
12	CON-PCS-P000018	verworfen	Projekt	Entwicklung
13	CON-PCS-P000019	angelegt	Projektvorlage Mechatronik	Entwicklung
14				

Abb. 2.31: Excel: PowerReports - Report mit einschränkender Suchbedingung (1)

	A	B	C	D
1	Kategorie	Vertrieb		
2				
3	cdb project id	cdb status txt	description	category
4	CON-PCS-P000004	abgeschlossen	Erstellung der Demo-Daten - englisch	Vertrieb
5				
6				

Abb. 2.32: Excel: PowerReports - Report mit einschränkender Suchbedingung (2)

Hinzufügen weiterer Hyperlinks

Hyperlinks werden abhängig von der Providerkonfiguration automatisch für die durch einen Provider gelieferten Objekte erzeugt. Sollen diese Objekte weitere Links zu referenzierten Objekten enthalten, müssen diese explizit hinzugefügt werden. Dabei ist zu beachten, dass das XSD Schema entsprechend erweitert werden muss. Die Erstellung von Hyperlinks erfolgt mit der Methode `MakeReportURL`. Folgendes Beispiel zeigt das Hinzufügen eines Hyperlinks für das zugeordnete Projekt eines Artikels.

```
def getSchema(self):
    t = XSDType(1, 'teile_stamm')
    t.add_attr("project_hyperlink", sqlapi.SQL_CHAR)
    return t
def getData(self, parent_result, source_args, **kwargs):
    item = pdd.Item.ByKeys('4711', '')
    data = ReportData(self, item)
    data["project_hyperlink"] = MakeReportURL(item.Project)
    return data
```

Im obigen Beispiel wird der Hyperlink mit der Default Aktion `CDB_ShowObject` und der konfigurierten Objektschreibung aus dem Klassenverzeichnis erzeugt.

Als zweiter und dritter Parameter kann der `MakeReportURL` Methode eine beliebige andere Aktion und eine anzuzeigende Beschriftung übergeben werden. Wird als Beschriftung ein Attributbezeichner des Objekts angegeben, wird der Wert dieses Attributes verwendet. Ansonsten wird als Beschriftung die angegebene Zeichenkette konstant verwendet.

```
def getData(self, parent_result, source_args, **kwargs):
    item = pdd.Item.ByKeys('4711', '')
    data = ReportData(self, item)
    data["project_hyperlink"] = MakeReportURL(item.Project, "CDB_Modify", "project_
→name")
    return data
```

Bei kombinierten Daten werden Hyperlinks grundsätzlich nur für das Primärobjekt automatisch erzeugt. Das folgende Beispiel zeigt Reportdaten für einen Artikel, erweitert um die Daten des zugehörigen Projekts. Für das Projektobjekt wird in diesem Fall kein Hyperlink automatisch angelegt. Soll ein Hyperlink auf das Projekt enthalten sein, muss dieser wie zuvor beschrieben manuell hinzugefügt werden.

```
def getData(self, parent_result, source_args, **kwargs):
    item = pdd.Item.ByKeys('4711', '')
    data = ReportData(self, item) # <- Primary object
    data += ReportData(self, item.Project, prefix="Project_")
    data["project_hyperlink"] = MakeReportURL(item.Project)
    return data
```

Hinzufügen von Bilddateien

Soll ein Datenprovider dynamische Bildinhalte liefern (siehe auch *Darstellung dynamischer Bildinhalte* (Seite 39)), dann muss zunächst wie im vorigen Kapitel das XSD Schema entsprechend erweitert werden.

```
def getSchema(self):
    schema = XSDType(CARD_N, 'teile_stamm')
    schema.add_attr('cdbxml_image_preview', sqlapi.SQL_CHAR)
    return schema
```

Das Hinzufügen von sogenannten Imagelinks erfolgt über die Methode `add_image`.

`add_image(self, image=None, attr_name='cdbxml_image', hyperlink='openfile', args)`

image Mögliche Werte:

- CDB_File-Objekt:
 - z.B. `CDB_File.ByKeys(cdb_object_id='...')`
 - Die Datei wird zusammen mit den anderen Report-Daten übergeben.
- Objekt-ID eines CDB_File-Objektes:
 - z.B. `some_preview_file_obj.cdb_object_id`
 - Die Datei wird zusammen mit den anderen Report-Daten übergeben.
- Einfacher Pfad:
 - z.B. `"C:\temp\bild.jpg"`
 - Die Datei wird zusammen mit den anderen Report-Daten übergeben.
- Absoluter URI Pfad:
 - z.B. `"file:///C:\temp\bild.jpg"`
 - Die Datei wird *nicht* den anderen Report-Daten übergeben, sondern später direkt bei der Reportgenerierung von der Kopplung importiert. Dabei ist natürlich zu beachten, dass die jeweilige Maschine, welche den Report zusammenstellt, Zugriff auf diesen (z.B. Netzwerk-)Pfad hat.
- HTTP Pfad:
 - z.B. `"http://local.myserver.de/bild.jpg"`
 - Die Datei wird *nicht* den anderen Report-Daten übergeben, sondern später direkt bei der Reportgenerierung von der Kopplung importiert. Dabei ist natürlich wiederum zu beachten, dass die jeweilige Maschine, welche den Report zusammenstellt, Zugriff auf diesen Internet-Pfad hat.

attr_name Das Attribut mit welchem zuvor an anderer Stelle das XSD Schema erweitert wurde (siehe *add_attr*).

hyperlink Standardmäßig wird jedes Bild im PowerReport mit einem Hyperlink verknüpft, welcher direkt auf die lokale Bilddatei zeigt. Dadurch sieht man das Bild in der Regel beim Anklicken im Internet-Explorer im (vergrößerten) Originalzustand.

Folgendes Beispiel zeigt das Hinzufügen einer Bilddatei.

```
def getData(self, parent_result, source_args, **kwargs):
    item = pdd.Item.ByKeys('4711', '')
    data = ReportData(self, item)
    filename = _getPreviewImageFile(item) # some custom method for retrieving_
    →regarding preview images
    data.add_image(filename, 'cdbxml_image_preview')
    return data
```

Ein weiteres Beispiel zeigt, wie man zusätzlich eigene Hyperlinks einbauen könnte.

```
def getData(self, parent_result, source_args, **kwargs):
    item = pdd.Item.ByKeys('4711', '')
    data = ReportData(self, item)
    filename = _getPreviewImageFile(item) # some custom method for retrieving_
    →regarding preview images
    hyperlink = item.MakeURL(None, plain=2)
    data.add_image(filename, 'cdbxml_image_preview', hyperlink)
    return data
```

Wichtig: Beim Aktualisieren eines PowerReports werden Bilddateien *nicht* mit aktualisiert.

Hinzufügen von Hierarchieinformationen bei hierarchischen Daten

Hierarchische Daten, wie beispielsweise Produktstrukturen oder Aufgabenstrukturen, können in PowerReports entsprechend dargestellt werden. Hierarchieebenen sind in PowerReports auf- und zu klappbar und können für eine bessere Übersicht farblich codiert und eingerückt werden.



14	CON-PROE-M000003	Verbindungsplatte
15	CON-PROE-M000001	Verbindungsblech
16	CON-PROE-M000002	Niet
17	CON-PROE-M000013	Griff_links
18	CON-PROE-M000006	Griffschale_links
19	CON-PROE-M000007	Kreuzschlitz-Schraubendreher
20	CON-PROE-M000008	Kapselheber
21	CON-PROE-M000021	Achse
22	CON-PROE-M000009	Sage
23	CON-PROE-M000009	Sage

Abb. 2.33: Excel: PowerReports - Hierarchische Darstellung am Beispiel Produktstruktur

Zur Darstellung hierarchischer Daten benötigt die Excel Kopplung Informationen über die Hierarchietiefe. Diese Information wird der Kopplung durch ein weiteres Attribut mit dem festen Namen `cdbxml_level` mitgeteilt.

Folgender Code zeigt am Beispiel Produktstruktur, wie `cdbxml_level` zum Schema hinzugefügt wird und beim Datenexport gesetzt werden kann. Bitte beachten Sie, dass die in dem Beispiel verwendete Methode `getComponents2` zur Ermittlung der Produktstruktur nicht Bestandteil einer System-Standarddistribution ist. Der Code kann daher nicht ohne weitere Anpassungen für einen eigenen Provider kopiert werden.

```
def getSchema(self):
    t = XSDType(N, "einzelteile")
    t.add_attr("cdbxml_level", sqlapi.SQL_INTEGER)
    return t

def getData(self, parent_result, source_args, **kwargs):
    item = parent_result.getObject()
    result = ReportDataList(self)
    # Custom method getComponents2 provides product structure tuples containing
    # item and depth
    for lev, obj in item.getComponents2():
        d = ReportData(self, obj)
        d["cdbxml_level"] = lev
        result.append(d)
    return result
```

Die Beschreibung zur Erstellung von Excelvorlagen mit Darstellung hierarchischer Daten finden Sie in *Darstellung hierarchischer Daten* (Seite 37).

2.4.5 Aktivierung von PowerReports

Kontextbezogene Reports müssen für jede Objektklasse, für die PowerReports verwendet werden sollen, aktiviert werden. Außerdem muss für jede zu aktivierende Objektklasse eine minimale Konfiguration zur Reportauswahl im jeweiligen Kontextmenü durchgeführt werden.

Die Aktivierung erfolgt in zwei Schritten:

1. Erweiterung der entsprechenden `cdb.objects` Klasse durch Ableitung
2. Aktivierung der 'Reports' Operation für das Kontextmenü

cdb.objects Klasse

Am Beispiel Projekte:

```
from cdb.objects.pcs.Projects import Project
from cdb.objects import PowerReports
class MyProject(Project, :guilabel:`PowerReports.WithPowerReports` ):
...
```

Falls für die zu aktivierende Objektklasse noch keine cdb.objects basierte Komponente existiert, muss diese zusätzlich in der Komponentenkonfiguration zu der ggf. schon existierenden server_<relation> Komponente angelegt werden. Am Beispiel Projekte:

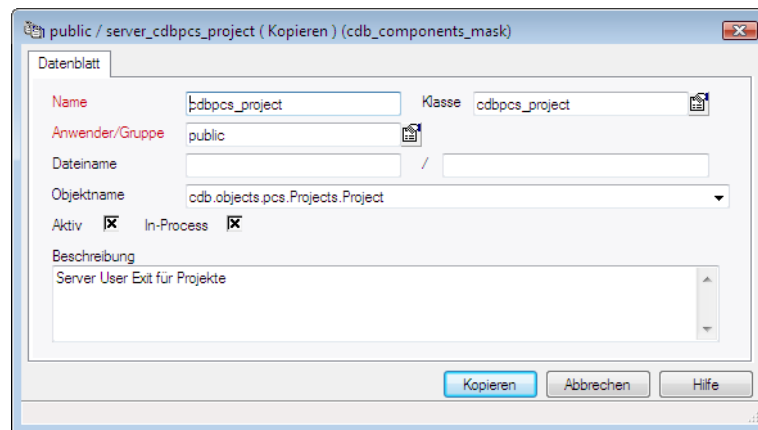


Abb. 2.34: Excel: PowerReports - cdb.objects basierte Komponente für Projekte

Sollen die Reports für mehrere Switch-Klassen mit unterschiedlichen Klassennamen aber gleichen Typs eingerichtet werden, also beispielsweise für verschiedene Projektklassen, muss pro Switch-Klassenname eine Komponente angelegt werden. Alle Komponenten müssen dabei dieselbe cdb.objects-Klasse im Feld Objektname verwenden.

Im Register 'Aktive Aufrufe' muss die Aktion cdbxml_excel_report mit dem Aufrufzeitpunkt * aktiviert werden. *Es dürfen auf keinen Fall alle Aktionen mit * aktiviert werden!* Dies würde in bestehenden Installationen zur unerwünschten Aktivierung zusätzlichen User Exit Codes führen, der zusätzlich zum existierenden server_<relation> User Exit ausgeführt werden würde.

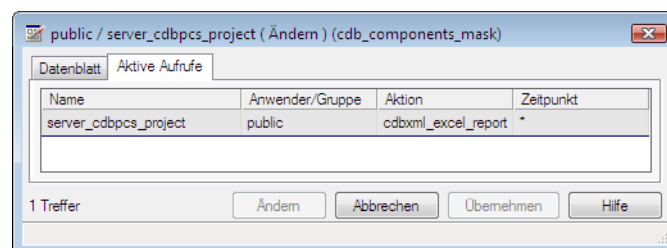


Abb. 2.35: Excel: PowerReports - Aktivierung der cdbxml_excel_report Operation

'Reports' Operation

Die Aktivierung der Operation erfolgt im Register 'Konfiguration' der Operation cdbxml_excel_report durch Hinzufügen eines weiteren Eintrages für die zu aktivierende Klasse. Standardmäßig ist mindestens ein Eintrag für Projekte vorhanden, der als Kopiervorlage verwendet werden kann.

Die Aktivierung von Reports am Beispiel für Dokumente ist damit abgeschlossen und es können beliebige Reports im Kontext von Dokumenten sowohl Single-Select als auch Multi-Select im Kontextmenü aufgerufen werden.

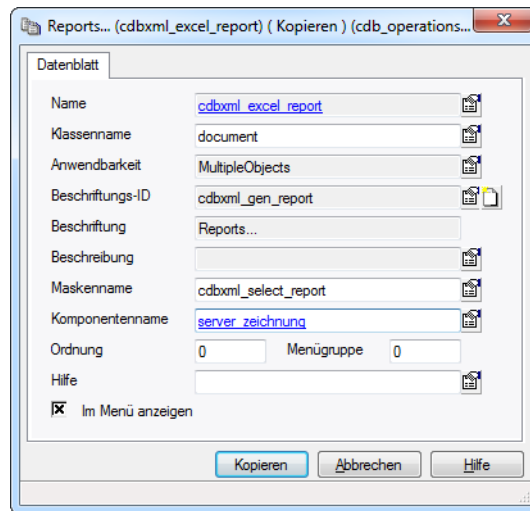


Abb. 2.36: Excel: PowerReports - Operationskonfiguration am Beispiel Dokumente



Abb. 2.37: Excel: PowerReports - Kontextmenüoperation zum Aufruf von Reports

Nach Auswahl der Kontextmenüfunktion 'Reports...' öffnet sich der Dialog zur Auswahl des auszuführenden Reports. Der Reportauswahlkatalog listet automatisch alle für den jeweiligen Anwender verfügbaren Reports. Dabei werden bei Multi-Select automatisch die Single-Select Reports ausgeblendet (siehe [Ausführen von PowerReports](#) (Seite 45)).

2.4.6 Reportvorlagen erstellen

Die Erstellung neuer Reports erfolgt mit Unterstützung der Excel Kopplung in drei Schritten:

1. Datenquelle auswählen

Die Auswahl der Datenquelle erfolgt mit der Kopplungsfunktion [*Datenquelle auswählen*] .

Es wird ein Dialog zur Auswahl einer im System konfigurierten Datenquelle angezeigt.

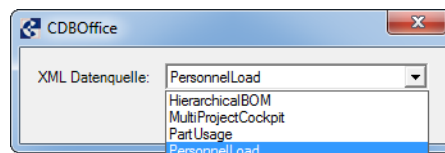


Abb. 2.38: Excel: PowerReports - Auswahldialog für Datenquellen

2. Datenbindung herstellen und Layout gestalten

Nach Auswahl der Datenquelle wird das zugehörige XSD Schema geladen und angezeigt. Das XSD Schema wird zur Herstellung der Datenbindung verwendet. Die Datenbindung wird via Drag&Drop von Schemaelementen an die gewünschte Position in das Excel Sheet hergestellt. Dabei können einzelne Attribute aber auch vollständige Objekte auf einmal positioniert werden.

3. Dokument als Reportvorlage speichern

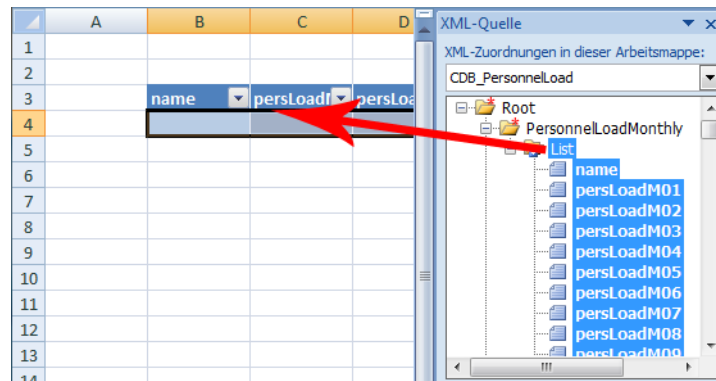


Abb. 2.39: Excel: PowerReports - Datenbindung via Drag&Drop herstellen

Abschließend muss der neu erstellte Report als Reportvorlage zur Datenquelle gespeichert werden. Dies erfolgt durch die Kopplungsfunktion [*Vorlage anlegen*] .

Reportvorlagen werden im System direkt als Dateien an den jeweiligen Report gehängt. Dabei können mehrere sprachspezifische Vorlagen für ein und den gleichen Report erstellt werden (siehe *Erstellung sprachspezifischer Reports* (Seite 44)).

Bemerkung: Da Reportvorlagen nicht als eigene Dokumente angelegt werden sondern direkt an die Reports geheftet, können die Reportvorlagen nicht im Nachhinein über die [*Laden*] Funktion der Kopplung zum erneuten Bearbeiten geöffnet werden. Zum erneuten Bearbeiten können sie entweder im Client (z.B. in der Projektübersicht) oder nach der Ausführung eines Reports über die [*Vorlage bearbeiten*] Funktion geöffnet werden.

Mit Abschluss der Dateienneuanlage wird die Reportvorlage automatisch der Datenquelle zugeordnet. Der Report ist anschließend sofort in Abhängigkeit von der Datenquelle entweder kontextfrei oder im Kontextmenü eines entsprechenden Objektes für den Reportersteller aufrufbar.

Alle zu einer Datenquelle erstellten Reports sind in dem Ordner Reports einer Datenquelle gelistet. Dort können die Berechtigungen, ein ggf. vorzuschaltender Dialog und der Reporttitel angepasst werden. Abbildung 32 zeigt exemplarisch den Report Projektportfolio zur gleichnamigen Datenquelle. Zu einer Datenquelle können beliebig viele Reports unterschiedlicher Ausprägung und in jeweils unterschiedlichen Sprachversionen erstellt werden.

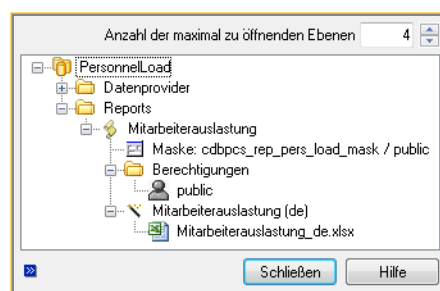


Abb. 2.40: Excel: PowerReports - Report zur Datenquelle

Der Reporttitel und ein ggf. vorzuschaltender Dialog können im Datenblatt der Vorlagenzuordnung zur Datenquelle konfiguriert werden. Das Register *Berechtigungen* steuert die Sichtbarkeit von Reports im Reportauswahldialog. Der Reportersteller wird bei der Neuanlage automatisch durch eine entsprechende Zuordnung berechtigt. Andere Anwender können den Report erst dann aufrufen, wenn diese hier zusätzlich durch eine Rollenzuordnung oder durch direkte Personenzuordnung berechtigt werden.

Desweiteren können in dem Datenblatt noch diverse Einstellungen zur Art der Reporterzeugung und dessen Darstellung vorgenommen werden (siehe *Auswahlkonfiguration* (Seite 46)).

Reportdialoge erstellen

Einem Report kann optional ein Dialog vorgeschaltet werden, der Parameter vom Anwender abfragt, die beim Aufruf eines Reports an die Datenprovider weitergegeben werden und dort die Datenermittlung beeinflussen können. Dies ist immer dann sinnvoll, wenn der Anwender das Reportergebnis durch eigene Eingaben beeinflussen können soll. Beispielsweise zur Eingabe eines zu betrachtenden Zeitraumes oder anderer filternder Bedingungen in Verbindung mit SimpleQuery Datenprovidern.

Abb. 2.41: Excel: PowerReports - Reportdialog zur Abfrage von Projekten

Die Parameter werden durch Namenskonvention für die Attributnamen der Maskenfelder an Datenprovider adressiert: <Providervname>-<Attributname>, wobei die Attributnamen jedoch keine Großbuchstaben enthalten dürfen. Das Feld Relation der Maskenfeldkonfiguration bleibt leer.

Name	Anwender/Gruppe	Relation	Attribut	Index
cdbpcs_project_report_mask	public		projekte-division	90
cdbpcs_project_report_mask	public		projekte-cost_center	100
cdbpcs_project_report_mask	public		projekte-cdb_status_txt	110
cdbpcs_project_report_mask	public		projekte-category	432

Abb. 2.42: Excel: PowerReports - Adressierung der Dialogfelder an Datenprovider als Parameter

Die Möglichkeiten zur Parametrisierung der universellen Standardprovider durch Dialoge finden Sie in *Providertypen im Detail* (Seite 14) zum jeweiligen Providertyp.

Bemerkung: Wenn Felder in Reportdialogen geändert, gelöscht oder hinzugefügt werden, ist darauf zu achten, dass in den Reportvorlagen, welche den jeweiligen Dialog verwenden, das XML Schema aktualisiert werden muss.

Fernerhin kann jeder Dialog um die Möglichkeit erweitert werden, sämtliche Vorbelegungen zu speichern, wodurch bei erneuten Reportausführungen sich eventuell wiederholende Eingaben vermieden werden können. Um dies zu erreichen, muss derjenige Dialog um 3 Elemente erweitert werden: die 2 (versteckten, aber den jeweiligen Report identifizierenden) Textfelder `cdbxml_source_name` und `cdbxml_report_title`; 1 Button welcher bei Betätigung die Komponente `cdbxml_set_defaults` auslöst (siehe *Excel: PowerReports - Speichern der Eingabewerte von Reportdialogen* (Seite 37)).

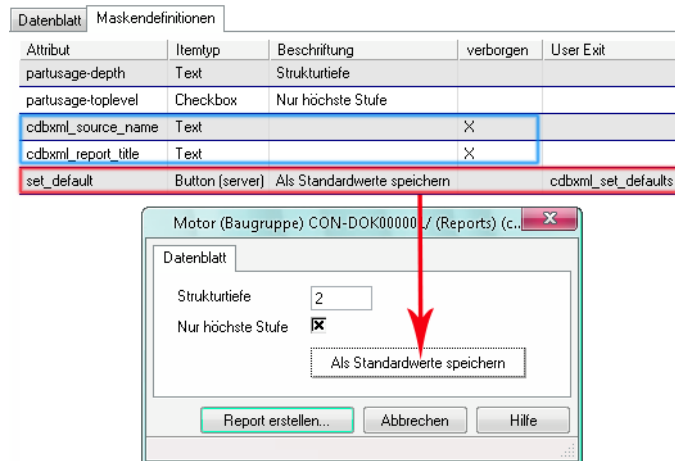


Abb. 2.43: Excel: PowerReports - Speichern der Eingabewerte von Reportdialogen

Darstellung hierarchischer Daten

In PowerReports können hierarchisch organisierte Daten entsprechend dargestellt werden. Typische Beispiele für hierarchische Daten sind Produktstrukturen oder Aufgabenstrukturen.

Voraussetzung für die hierarchische Darstellung ist, dass der Datenprovider die Hierarchieinformation in dem Attribut `cdxml_level` bereitstellt. Ist diese Voraussetzung erfüllt, können die Hierarchieebenen im PowerReport auf- und zu geklappt und für eine bessere Übersicht farblich codiert und eingerückt werden.

Zur Erstellung einer hierarchisch organisierten Liste muss das Attribut `cdxml_level` in der Liste enthalten sein. Idealerweise wird dieses Attribut am Ende des Listekopfes angefügt und mit Abschluss der Layoutarbeiten durch Setzen der Spaltenbreite auf 0 unsichtbar gesetzt. Ist das Attribut `cdxml_level` in der Liste enthalten, wird die Ergebnistabelle automatisch, wie in folgender Abbildung am linken Rand dargestellt, mit Excel Funktionalität gegliedert.

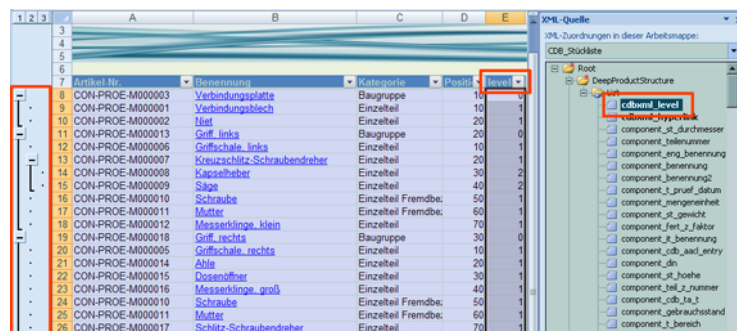


Abb. 2.44: Excel: PowerReports - Hierarchische Darstellung am Beispiel Produktstruktur

Diese Gliederung kann zusätzlich durch Einrückung einer beliebigen Spalte und durch farbliche Codierung hervor gehoben werden. Hierfür muss jeweils ein spezieller Ausdruck in die Spaltenbeschriftung des `cdxml_level` Attributes aufgenommen werden. Die Ausdrücke müssen dabei durch einen Doppelpunkt voneinander getrennt werden. Einrückung einer Spalte:

```
indent(column)
```

Farbliche Codierung:

```
colorize(<rgb_start_color>), (<rgb_offset>)
```

column gibt den Spaltenindex an, in welchem die Einrückungen ausgeführt werden.

`rgb_start_color` gibt dabei die Startfarbe als RGB Farbcodierung für die Hierarchieebene 0 an. Für die darunterliegenden Ebenen wird die Zeilenfarbe durch addieren des `rgb_offsets` ermittelt.

Excel: PowerReports - Hierarchische Darstellung mit Einrückung der 1. Spalte und Farbcodierung

Nur Einrückung der 1 Spalte:

```
level:indent(1)
```

Einrücken der 1.Spalte + blaue Farbcodierung:

```
level:colorize(230,230,255),(20,15,12):indent(1)
```

Der Ausdruck `level:colorize(230,230,255),(20,15,12):indent(1)` ergibt beispielsweise folgende Darstellung.

13	Artikel.Nr.	Benennung	Kategorie	Positi
14	CON-PROE-M000003	Verbindungsplatte	Baugruppe	10
15	CON-PROE-M000001	Verbindungsblech	Einzelteil	10
16	CON-PROE-M000002	Niet	Einzelteil	20
17	CON-PROE-M000013	Griff_links	Baugruppe	20
18	CON-PROE-M000006	Griffschale_links	Einzelteil	10
19	CON-PROE-M000007	Kreuzschlitz-Schraubendreher	Einzelteil	20
20	CON-PROE-M000008	Kapselheber	Einzelteil	30
21	CON-PROE-M000009	Säge	Einzelteil	40
22	CON-PROE-M000010	Schraube	Einzelteil Fremdbe	50
23	CON-PROE-M000011	Mutter	Einzelteil Fremdbe	60
24	CON-PROE-M000012	Messerklinge_klein	Einzelteil	70
25	CON-PROE-M000018	Griff_rechts	Baugruppe	30
26	CON-PROE-M000005	Griffschale_rechts	Einzelteil	10
27	CON-PROE-M000014	Ahle	Einzelteil	20
28	CON-PROE-M000015	Dosenöffner	Einzelteil	30
29	CON-PROE-M000016	Messerklinge_groß	Einzelteil	40
30	CON-PROE-M000010	Schraube	Einzelteil Fremdbe	50
31	CON-PROE-M000011	Mutter	Einzelteil Fremdbe	60
32	CON-PROE-M000017	Schlitz-Schraubendreher	Einzelteil	70

Abb. 2.45: Excel: PowerReports - Hierarchische Darstellung mit Einrückung der 1. Spalte und Farbcodierung

Bemerkung: Standardmäßig werden in Excel Gruppierungen mit der Einstellung ‘Hauptzeilen unter Detaildaten’ angezeigt. In der Regel wird sich der Anwender diese jedoch wie in obigem Beispiel über den Detaildaten wünschen. Daher sollte darauf geachtet werden, dass bei der Erstellung der PowerReport-Vorlage diese Einstellung deaktiviert wird (*Daten* → *Gliederung* → *Einstellungen*).

13	Artikel.Nr.		13	Artikel.Nr.
14	CON-PROE-M000003		14	CON-PROE-M000003
15	CON-PROE-M000001		15	CON-PROE-M000001
16	CON-PROE-M000002		16	CON-PROE-M000002
17	CON-PROE-M000013		17	CON-PROE-M000013

Abb. 2.46: Excel: PowerReports - Hierarchische Darstellung - Gruppierungseinstellungen

Darstellung dynamischer Spaltentitel in Listen

In PowerReports ist es möglich die Spaltentitel von Listen, welche durch Provider gefüllt werden, dynamisch mit Werten zu belegen. Als Beispiel dient der folgende PowerReport:

Dieser PowerReport zur Anzeige der Mitarbeiterauslastung zeigt in einer Liste einen durch den Benutzer vor der Generierung angegebenen Zeitraum. Im Nachhinein kann in dem rot markierten Feld der Zeitraum geändert und der PowerReport aktualisiert werden. Die Spaltentitel sind somit abhängig vom Zeitraum, da sie die jeweiligen Kalenderwochen anzeigen. Realisiert wird dies durch Hinzufügen eines weiteren Providers (‘PersonnelLoadWeekly_Label’), welcher bei jedem Erstellen/Aktualisieren eines PowerReports die jeweils aktuellen Spaltentitel liefert. Damit die Excel-Kopplung diese Werte finden, auslesen und in die Liste übertragen kann, müssen sie in die Zeile direkt über der Liste gelegt werden. Im Abschluss muss diese Zeile in der Vorlage versteckt werden.

		KW 37 2009	KW 38 2009	KW 39 2009	KW 40 2009	KW 41 2009	KW 42 2009	KW 43 2009	KW 44 2009	KW 45 2009	KW 46 2009	KW 47 2009	KW 48 2009
Mitarbeiter													
10	Bertrand, Adrian	40,0	40,0	40,0	40,0	40,0	40,0	40,0	40,0	40,0	40,0	40,0	40,0
11	Cordes, Paul	30,0	30,0	30,0	30,0	30,0	30,0	30,0	30,0	30,0	30,0	30,0	30,0
12	Enver, Julia	40,0	40,0	40,0	40,0	40,0	40,0	40,0	40,0	40,0	40,0	40,0	40,0
13	Hanna, Tina	30,0	30,0	30,0	30,0	30,0	30,0	30,0	30,0	30,0	30,0	30,0	30,0

Abb. 2.47: Excel: PowerReports - Darstellung dynamischer Spaltentitel

		KW 37 2009	KW 38 2009	KW 39 2009	KW 40 2009	KW 41 2009	KW 42 2009	KW 43 2009	KW 44 2009	KW 45 2009	KW 46 2009	KW 47 2009	KW 48 2009
Mitarbeiter													
10	Bertrand, Adrian	40,0	40,0	40,0	40,0	40,0	40,0	40,0	40,0	40,0	40,0	40,0	40,0
11	Cordes, Paul	30,0	30,0	30,0	30,0	30,0	30,0	30,0	30,0	30,0	30,0	30,0	30,0
12	Enver, Julia	40,0	40,0	40,0	40,0	40,0	40,0	40,0	40,0	40,0	40,0	40,0	40,0
13	Hanna, Tina	30,0	30,0	30,0	30,0	30,0	30,0	30,0	30,0	30,0	30,0	30,0	30,0

Abb. 2.48: Excel: PowerReports - Darstellung dynamischer Spaltentitel (Layout)

Darstellung dynamischer Bildinhalte

In PowerReports hat man die Möglichkeit, sowohl einfache Felder wie auch Listenfelder mit Image-Dateien (in allen von Excel unterstützten Formaten) zu füllen. Welche Bilder man bereitstellen kann und wie man diese bereitstellt, wird im Abschnitt *Hinzufügen von Bilddateien* (Seite 30) beschrieben. Damit die bei der Reportgenerierung dynamisch geladenen Bilddateien korrekt eingefügt werden bzw. eine gewünschte einheitliche Größe oder Position innerhalb der Zelle bekommen, muss in der Reportvorlage ein entsprechendes Musterbild eingefügt werden. Damit dieses Bild von der Kopplung als Muster erkannt wird, muss entweder der Bildname mit "cdb_image" oder der Attributname, welcher dem zugehörigen Feld über den Datenprovider zugewiesen wurde, mit "cdbxml_image" beginnen. Des weiteren können bestimmte Formatierungsregeln in der Alternativtextbox im Eigenschaftendialog des jeweiligen Bildes festgelegt werden.

Wichtig: Das Vorlagebild muss sich mit seiner linken oberen Ecke eindeutig innerhalb der jeweiligen Zelle befinden, ansonsten verknüpft Excel das Bild mit einer anderen Zelle und die dynamischen Bilder werden u.U. nicht eingefügt.

In bestimmten Fällen kommt es bei dynamischen Bilderlisten zu unerwarteten Ergebnissen (z.B. verzerrte oder unsichtbare Bilder). Dies betrifft zur Zeit vornehmlich Excel 2010. Vermeiden lässt sich dies in der Regel dadurch, dass die Bildvorlage so in der Zelle platziert wird, dass sie (bzw. die später eingefügten Bilder) nicht über die Zellgrenze hinausragen. Insofern bei einem PowerReport ein anschließendes Filtern oder Sortieren von Listen gewünscht ist, welche dynamische Bilder enthalten, müssen die betroffenen Bildvorlagen innerhalb dieser Liste mit der Eigenschaft "Von Zellposition und -größe abhängig" formatiert werden. Andernfalls kann es bei einer filter- oder sortierbedingten Listenzeilenbewegung zu falschen Bildergebnissen kommen.

Wichtig: Dynamische Bildinhalte werden zur Zeit in komplexen Reports nicht unterstützt.

Formatierungsmöglichkeiten:

- *KeepWidth* : Die Breite des Musterbildes wird auf den dynamischen Bildinhalt übertragen. Diese Variable ist intern standardmäßig aktiviert, da in der Regel ein Einhalten der Spaltenbreite erwünscht ist.

- *KeepHeight* : Die Höhe des Musterbildes wird auf den dynamischen Bildinhalt übertragen.
- *MinBorderHeight* : Der Mindestabstand zwischen Bildhöhe und Zellhöhe. Ein Wert von “1” garantiert z.B., dass beim Füllen von ‘zu hochformatigen’ Bildern die jeweilige Zeilenhöhe vergrößert und dadurch ein Überschneiden von Bildinhalten vermieden wird. Höhere Werte wie “20” oder “30” sorgen für einen optischen Rand zwischen Bild und Zelle.
- *MinBorderWidth* : Der Mindestabstand zwischen Bildbreite und Zellbreite.
- *CenterWidth* : Der dynamische Bildinhalt wird innerhalb der Zelle horizontal zentriert.
- *CenterHeight* : Der dynamische Bildinhalt wird innerhalb der Zelle vertikal zentriert.

Wichtig: Der gesamte Formatierungstext muss mit dem Präfix “cdb:” beginnen.

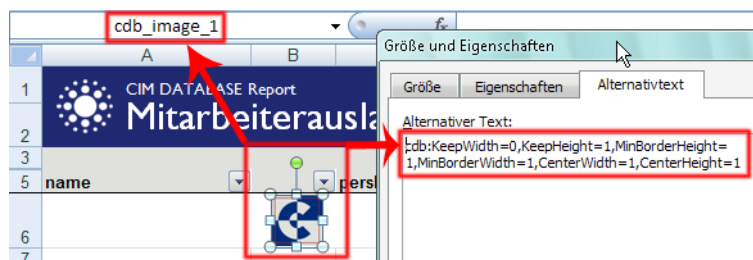


Abb. 2.49: Excel: PowerReports - Dynamische Bildinhalte (Musterbild)

- Formatierungsbeispiel 1: “*cdb:KeepHeight=1,CenterWidth=1*”

	A	B	C	D	E
1	CIM DATABASE Report				
2	Mitarbeiterauslastung				
3					
5	Mitarbeiter		03 2011	04 2011	05 2011
6	Bertrand, Adrian		168,0	168,0	176,0
7	Cordes, Paul		126,0	126,0	132,0
8	Enver, Julia		168,0	168,0	176,0
9	Hegge, Tina		126,0	126,0	132,0

Abb. 2.50: Excel: PowerReports - Dynamische Bildinhalte (Beispiel 1)

- Formatierungsbeispiel 2: “*cdb:MinBorderHeight=1,KeepWidth=1*”

Darstellung komplexer Zeilen bei Multi-Export Datenquellen

Mit Microsoft Excel ist es nicht möglich, Listen bestehend aus komplexeren Zeilen zu erstellen. Eine Listezeile besteht immer nur aus genau einer Zeile Excel Zellen. Häufig wird jedoch die Darstellung komplexerer Daten in Listenform benötigt. Die folgende Darstellung einer Artikelliste mit jeweils einer Liste der zugeordneten Dokumente ist beispielsweise allein mit Excel Listen nicht realisierbar. Stattdessen müssen derartige Listen durch aneinanderfügen eines sich wiederholenden Layouts erstellt werden. In folgender Abbildung ist ein solches komplexes Listenelement rot gekennzeichnet. Dieses Element wiederholt sich beliebig häufig und bildet dadurch eine komplexe Liste.

Das Layout und die Datenbindung für ein komplexes Listenelement muss auf einem speziellen Template Sheet mit dem fest vorgegebenen Namen `cdb_template` erfolgen. Auf dem Template Sheet wird genau eine komplexe Zeile modelliert.

	A	B	C	D	E
1	CIM DATABASE Report				
2	Mitarbeiterauslastung				
3					
4					
5	Mitarbeiter				
6	Bertrand, Adrian		40,0	40,0	40,0
7	Cordes, Paul		30,0	30,0	30,0
8	Enver, Julia		40,0	40,0	40,0

Abb. 2.51: Excel: PowerReports - Dynamische Bildinhalte (Beispiel 2)

79	Proj. Multitool - Net	CON-PROE-M000002-001		in Arbeit	CAD Dokumente	3D-Einzelteil	CON-PCS-P00 Engineering
80							
81	Artikel	CON-PROE-M000003	Projekt	Mult. Werkzeug		EC	
82	Index		Status	in Bearbeitung		Status	
83	Benennung	Verbindungsplatte	Kategorie	Entwicklung		Kategorie	
84	Dokumente zum Artikel						
85	Titel	[> Dokument.Nr. > Index	Status	Hauptkategorie	Kategorie	Projekt.Nr. > Abteilung	
86	Dokument	CON-DOK000001-49	in Arbeit	Projektmanagement	Projektplan	CON-PCS-P00 Engineering	
87	Prozessschrittbericht	CON-DOK000001-50	in Änderung	Projektmanagement	Statusbericht	CON-PCS-P00 Engineering	
88	Änderungsantrag - FR LA	CON-DOK000001-52	freigegeben	Qualität	Änderungsantrag	CON-PCS-P00 Prototyping	
89	Checkliste - Q2 Paperroll	CON-DOK000001-58	freigegeben	Qualität	Checkliste	CON-PCS-P00 Engineering	
90	Prozessschritt - Beispiel	CON-DOK000001-68	freigegeben	Qualität	Proz.Vorschlagsplan	CON-PCS-P00 Test	
91	Wissensmanagement - IT	CON-DOK000001-78	freigegeben	Wissen	Organisationswissen	CON-PCS-P00 Engineering	
92	Wissensmanagement - IT	CON-DOK000001-81	freigegeben	Wissen	Organisationswissen	CON-PCS-P00 Engineering	
93	Wissensmanagement - IT	CON-DOK000001-95	freigegeben	Wissen	Organisationswissen	CON-PCS-P00 Engineering	
94	Wissensmanagement - IT	CON-DOK000001-95	freigegeben	Wissen	Organisationswissen	CON-PCS-P00 Engineering	
95	Proj. Multitool - Gesamt	CON-PROE-M000003-001	in Arbeit	CAD Dokumente	3D-Gruppe	CON-PCS-P00 Engineering	
96							
97	Artikel	CON-PROE-M000004	Projekt	Mult. Werkzeug		EC	
98	Index		Status	in Bearbeitung		Status	
99	Benennung	Klammer	Kategorie	Entwicklung		Kategorie	
100	Dokumente zum Artikel						
101	Titel	[> Dokument.Nr. > Index	Status	Hauptkategorie	Kategorie	Projekt.Nr. > Abteilung	
102	Werkzeuganweisung	CON-DOK000001-2-1	in Arbeit	Produkt Teil	Film	CON-PCS-P00 Engineering	
103	Zusatzteil - Bauteile	CON-DOK000001-44	in Arbeit	Projektmanagement	Pflichtentwurf	CON-PCS-P00 Engineering	
104	Zusatzteil - Mutter	CON-DOK000001-47	freigegeben	Projektmanagement	Pflichtentwurf	CON-PCS-P00 Engineering	
105	Änderungsantrag - User	CON-DOK000001-1-1	in Prüfung	Qualität	Änderungsantrag	CON-PCS-P00 Engineering	
106	Proj. Multitool - Einzelteil	CON-PROE-M000004-001	in Arbeit	CAD Dokumente	3D-Einzelteil	CON-PCS-P00 IT	
107							
108	Artikel	CON-PROE-M000005	Projekt	Mult. Werkzeug		EC	
109	Index		Status	in Bearbeitung		Status	
110	Benennung	Griffschale, rechts	Kategorie	Entwicklung		Kategorie	
111	Dokumente zum Artikel						
112	Titel	[> Dokument.Nr. > Index	Status	Hauptkategorie	Kategorie	Projekt.Nr. > Abteilung	
113	Proj. Multitool - Griffschale	CON-PROE-M000005-001	in Arbeit	CAD Dokumente	3D-Einzelteil	CON-PCS-P00 Engineering	
114							
115	Artikel	CON-PROE-M000006	Projekt	Mult. Werkzeug		EC	
116	Index		Status	in Bearbeitung		Status	
117	Benennung	Griffschale, links	Kategorie	Entwicklung		Kategorie	
118	Dokumente zum Artikel						
119	Titel	[> Dokument.Nr. > Index	Status	Hauptkategorie	Kategorie	Projekt.Nr. > Abteilung	
120	Proj. Multitool - Griffschale	CON-PROE-M000006-001	in Arbeit	CAD Dokumente	3D-Einzelteil	CON-PCS-P00 Engineering	
121							
122	Artikel	CON-PROE-M000007	Projekt	Mult. Werkzeug		EC	
123	Index		Status	in Bearbeitung		Status	
124	Benennung	Griffschale, links	Kategorie	Entwicklung		Kategorie	
125	Dokumente zum Artikel						
126	Titel	[> Dokument.Nr. > Index	Status	Hauptkategorie	Kategorie	Projekt.Nr. > Abteilung	
127	Proj. Multitool - Griffschale	CON-PROE-M000007-001	in Arbeit	CAD Dokumente	3D-Einzelteil	CON-PCS-P00 Engineering	
128							

Abb. 2.52: Excel: PowerReports - Komplexe Zeilen am Beispiel Artikelliste

Bei der Reportausführung werden die komplexen Zeilen einer Datenquelle in einzelne XML Dateien exportiert. Diese werden durch die Excel Kopplung anschließend einzeln in das Template Sheet importiert. Zwischen den Importvorgängen wird das Ergebnis vom Template Sheet in das Zielsheet kopiert.

	A	B	C	D	E
1	Tabelle1\$A6				
2					
3	Artikel	CON-PROE-M000022		Projekt	
4	Index			Status	
5	Benennung	Multitool		Kategorie	
6	Dokumente zum Artikel				
7	<input type="text" value="Titel"/>	<input type="text" value="Dokument-Nr."/>	<input type="text" value="Index"/>	<input type="text" value="Status"/>	<input type="text" value="Hauptkategorie"/>
8					
9					
10					
11					
12					

Abb. 2.53: Excel: PowerReports - Definition einer komplexen Zeile

In der oberen linken Zelle des Template Sheets kann das Zielsheet und eine Zielzelle angegeben werden. Diese Zieladresse beschreibt das Ziel für die obere linke Zelle der 1. zu kopierenden komplexen Zeile. Alle weiteren Zeilen werden von dort ausgehend fortlaufend aneinandergehängt. Wird keine Zieladresse angegeben, wird als Startzelle die obere linke Zelle des ersten im Workbook enthaltenen Sheets verwendet.

Das PowerReports Framework erkennt Daten, die als komplexe Zeilen exportiert und importiert werden müssen, automatisch anhand der Ergebnis- und Aufrufkardinalitäten verknüpfter Datenprovider.

Eine Multi-Export Konfiguration liegt immer dann vor, wenn der übergeordnete Provider als Ergebnis eine Liste liefert (Ergebniskardinalität N) und ein zugeordneter Subprovider die Aufrufkardinalität 1 hat, der Subprovider also Daten liefert, die sich auf einzelne Listenelemente des übergeordneten Providers beziehen.

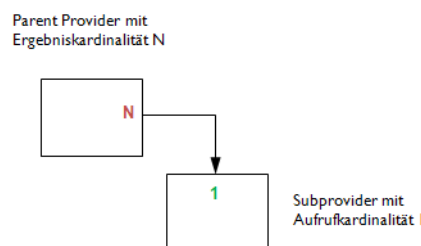


Abb. 2.54: Excel: PowerReports - Schema Multi-Export

Liefert beispielsweise ein SimpleQuery Provider eine Liste von Artikeln und ein Relationship Provider liefert als zugeordneter Subprovider die zugeordneten Dokumente, so liegt eine Konfiguration vor, die ausschließlich als komplexe Liste in Excel darstellbar ist.

Ein zugeordneter GroupBy Provider würde hingegen keine Multi-Export Konfiguration darstellen, da GroupBy Provider mit dem gesamten Ergebnis des übergeordneten Providers aufgerufen werden, also die Aufrufkardinalität N haben. GroupBy Provider liefern keine Ergebnisse, die sich auf einzelne Listenelemente des Parent Providers beziehen sondern aggregierte Informationen, die sich ausschließlich auf die gesamte Liste beziehen.

Darstellung von Langtexten

Für Zeilen, in welchen Zellen ausschließlich oder teilweise mit Langtexten gefüllt werden, muss die Eigenschaft *Zeilenumbruch* der kompletten Zeile gesetzt sein, damit die Kopplung diese erkennen und deren Zeilenhöhe automatisch anpassen kann.

Falls Langtexte in Zellen angezeigt werden sollen, welche aus mehreren Verbundzellen bestehen, dann lässt sich die korrekte Darstellung nur über einen Umweg erreichen:

1. Die eigentliche Verknüpfung mit dem Langtext-Attribut der Datenquelle darf nicht mit der gewünschten Ziel-Zelle erfolgen, sondern mit einer am Ende *versteckten Zelle*, welche sich auf der *gleichen Zeile* befindet.

2. Damit die Langtexte nicht abgeschnitten oder mit anfänglichen Leerzeilen dargestellt werden, muss diese Helfer-Zelle absolut die *gleiche Breite* besitzen.
3. Die Eigenschaft “*Zeilenumbruch*” muss für die komplette Zeile gesetzt sein.

Nachbearbeitung fertiger PowerReports durch VBA

Manchmal möchte man programmtechnisch über VBA und das Excel-Object-Model bestimmte Dinge erledigen, nachdem ein Report von der Kopplung fertig gestellt wurde (z.B. um gefüllte Diagramme korrekt zu formatieren, oder um etwas zu fokussieren). Zu diesem Zweck sucht die Kopplung in der Excel-Mappe eine VBA-Funktion mit einem bestimmten Namen. Wird diese gefunden, wird sie jeweils nach Fertigstellung/Aktualisierung eines Reports aufgerufen.

Hinzufügen einer VBA-Funktion zur automatisierten Nachbearbeitung fertiggestellter PowerReports

1. In Excel den VBA-Editor öffnen.
2. In der PowerReport-Vorlage ein Modul einfügen.
3. Eine Funktion mit dem Namen ‘CDBReportDone’ einbauen:

```
Public Sub CDBReportDone()  
    ...  
End Sub
```

Bemerkung: Falls die Sicherheitseinstellungen in Office so eingestellt sind, dass nicht digital signierte Makros automatisch deaktiviert werden, müssen die Report-Vorlagen mit einem Zertifikat signiert werden, bevor sie im System gespeichert werden. Diejenigen mit dem System gelieferten Standard-PowerReports, welche VBA-Code zur Nachbearbeitung enthalten, sind bereits mit einem für die Contact GmbH ausgestellt Zertifikat und unter Verwendung eines Zeitstempel-Servers signiert. Um eigene Report-Vorlagen oder geänderte Standard-Report-Vorlagen mit einem eigenen Zertifikat und unter Verwendung eines Zeitstempel-Servers zu signieren, müssen manuell folgende Einträge in der Registry gesetzt werden, da sich der Zertifikats-Zeitstempel-Server nicht über die Office-Oberfläche konfigurieren lässt:

```
Windows Registry Editor Version 5.00  
  
[HKEY_CURRENT_USER\Software\Microsoft\VBA\Security]  
"TimeStampURL"="http://timestamp.some-ca.com/scripts/timestamp.dll"  
"TimeStampRetryCount"=dword:00000005  
"TimeStampRetryDelay"=dword:00000005
```

Falls das digitale Signieren ohne Zeitstempel erfolgt, erlöscht die Gültigkeit der Signatur nach Ablauf des Zertifikates. Bei Verwendung eines Zeitstempels ist die Signatur zeitlich unbegrenzt gültig.

Erstellung persönlicher Reports

Anwender haben die Möglichkeit, das Layout und die Datenbindung eines aufgerufenen Reports direkt zu modifizieren und als persönlichen Report abzuspeichern. Die Erstellung eines persönlichen Reports kann also ad-hoc erfolgen, wenn dem Anwender das Reportergebnis nicht gefällt oder für seine Zwecke unpassend ist.

Die Speicherung des modifizierten Reports als persönlichen Report erfolgt mit der Excel Kopplungsfunktion *Reports* → *Reportvorlage anlegen*. Der modifizierte Report wird durch Aufruf dieser Funktion als neue Reportvorlage zur zugrundeliegenden Datenquelle angelegt und für den ausführenden Anwender berechtigt. Andere Anwender haben keinen Zugriff auf diesen Report, könnten aber durch den Administrator durch Rollen- und Personenzuordnungen nachträglich berechtigt werden.

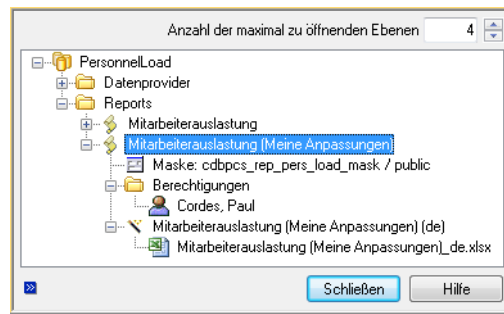


Abb. 2.55: Excel: PowerReports - Persönlicher Report am Beispiel Mitarbeiterauslastung

Bei der Erstellung eines persönlichen Reports wird der ggf. vorgeschaltete Reportdialog vom ursprünglichen Report übernommen. Die Änderung des Reportdialogs durch Anwender ist nicht vorgesehen.

Erstellung sprachspezifischer Reports

Der Gestalter eines PowerReports hat die Möglichkeit eine bereits bestehende Reportvorlage für eine bestimmte Sprache anzupassen und anschliessend für den gleichen übergeordneten Report bereitzustellen, so dass die Anwender, welche diesen Report im Nachhinein ausführen, zwischen verschiedenen Sprachen wählen können. Falls man eine bereits angelegte Reportvorlage zum Bearbeiten öffnet und man hinterher [*Vorlage anlegen*] ausführt, bietet die Kopplung in einem Zwischendialog an, für bereits bestehende Reports eine sprachspezifische Reportvorlage anzulegen.



Abb. 2.56: Excel: PowerReports - Report zur Datenquelle

Im darauf folgenden Dialog muss nur noch die Sprache und ein eindeutiger Titel gewählt werden.

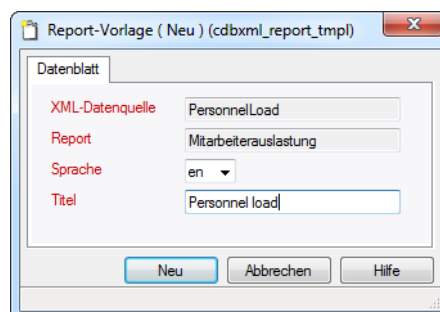


Abb. 2.57: Excel: PowerReports - Report zur Datenquelle

Besondere Hinweise

- Bei der Verknüpfung zwischen Listenattributen eines Providerschemas und den Excel-Zellen, d.h. wenn durch die Verknüpfung in dem Blatt Tabellenspalten entstehen, dann ist besonders darauf zu achten, dass die jeweilige Tabelle als Ganzes in logischer Reihenfolge von links nach rechts aufgebaut wird. Wenn im Nachhinein aus dieser Tabelle wieder Spalten gelöscht oder eingefügt werden, kann es unter Umständen zu Fehlern oder sogar Abbrüchen bei der Generierung der Reports kommen, da die Spalten dann eventuell nicht mehr eine gesamte Tabelle bilden, sondern mehrere nebeneinander stehende Tabellen. Ob jetzt eine Tabelle aus mehrerer Tabellen besteht, kann man an den kleinen Dreiecken am rechten unteren Rand einer jeden Tabelle erkennen. Vor allem seit Excel 2010 stößt man diesbezüglich oft auf Probleme, welche sich nur dadurch lösen lassen, dass Tabellen entfernt und von neuem aufgebaut werden.
- Das Füllen von Daten in Verbundzellen hinein kann zu unerwartetem Fehlverhalten führen. In solchen Fällen kann meist anstelle der Option “Zellen verbinden” auch einfach die Option “Textausrichtung > Horizontal > Über Auswahl zentrieren” verwendet werden.
- Für mit Datumsformaten oder benutzerdefinierten Zahlenformaten vorkonfigurierte Zellen darf nicht gleichzeitig die Option “Zeilenumbruch” gesetzt sein, da dies zu einem Verlust der Formatierung führen kann.
- Bei der Verknüpfung zwischen einem Schemaattribut und einer Zelle formatiert Excel die Zelle automatisch mit der zum jeweiligen Attributwertetyp passenden Zahlenkategorie. Da jetzt z.B. Zellen, welche später mit Gleitkommazahlen gefüllt werden sollen, von Excel automatisch der Zahlenkategorie ‘Standard’ zugewiesen werden, werden eventuelle Nachkommastellen nur dargestellt, wenn diese nicht gleich Null sind. Wenn ein fixes Nachkommastellenformat (0,00) erwünscht ist, muss man die Zahlenkategorie nach der Verknüpfung in ‘Zahl’ ändern.
- Falls in irgendeinem Sheet gleich mehrere dynamisch gefüllte Tabellen übereinander verwendet werden, dann darf eine untere Tabelle niemals breiter sein als die Tabelle(n) über ihr. Sie darf also höchstens genauso breit oder natürlich auch schmaler sein, da es sonst zu internen Problemen kommt, in Folge dessen die Reportgenerierung abbricht. Falls untere Tabellen also vom Design her doch breiter sein müssen, dann müssen die höheren Tabellen z.B. durch unsichtbare Dummy-Spalten verbreitert werden.

2.4.7 Ausführen von PowerReports

Auswahldialog

Nach Auswahl einer Report-Kontextmenüfunktion öffnet sich der Dialog zur Auswahl des auszuführenden Reports. Der Reportauswahlkatalog listet automatisch alle für den jeweiligen Anwender verfügbaren Reports. Dabei werden bei Multi-Select automatisch die Single-Select Reports ausgeblendet.

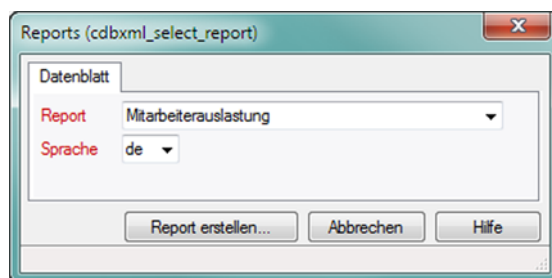


Abb. 2.58: Excel: PowerReports - Reportauswahldialog

Die Comboboxen zur Auswahl des Reports und der Sprache werden automatisch mit der jeweiligen Anmeldesprache vorbelegt. Falls in der Anmeldesprache keine Reports zur Verfügung stehen, wird auf die (konfigurierbaren) Standardsprachen ausgewichen. Falls für letztere Sprachen auch keine Reports vorhanden sind, wird der Dialog mit der nächst zur Verfügung stehenden Sprache vorbelegt. Wie man für einen Report Vorlagen in zusätzlichen Sprachen anlegt wird im Kapitel *Erstellung sprachspezifischer Reports* (Seite 44) erklärt.

Ausführungstypen und -formate

Reporterzeugung

- Client
 - Mögliche Reportformate: *Excel*
 - Bedingung: Auf dem Arbeitsplatzrechner muss die Kopplung installiert sein.
- Server (asynchron)
 - Mögliche Reportformate: *Excel, PDF, Excel & PDF*
 - Die Kopplung muss nicht auf dem Arbeitsplatzrechner installiert werden.
 - Die fertigen PowerReports werden dem Benutzer standardmäßig als Email zugesandt.
- Server (synchron)
 - Mögliche Reportformate: *Excel, PDF, E-Link*
 - Die Kopplung muss nicht auf dem Arbeitsplatzrechner installiert werden.
 - Die fertigen PowerReports werden nach Fertigstellung direkt auf dem Arbeitsplatzrechner zur Ansicht geöffnet.

Bemerkung: Die serverseitige Erzeugung ist nur möglich, wenn ihr CDB-Administrator die jeweiligen Dienste installiert und aktiviert hat.

Ab CDB 9.8 *SP1* können alle fertiggestellten PowerReports in einem eigenen Reiter innerhalb des PC-Clients (als eLink-Panel) geöffnet werden. Innerhalb der Titelleiste befindet sich ein Button mit einem Excel-Symbol, welcher den gezeigten Report bei Bedarf in Excel öffnet. Falls der Report über einen zusätzlichen Dialog verfügt, ist noch ein zweiter Button mit einem Filter-Symbol zu sehen, welcher den jeweiligen Report nach dem Einblenden des Zwischendialoges erneut ausführt.

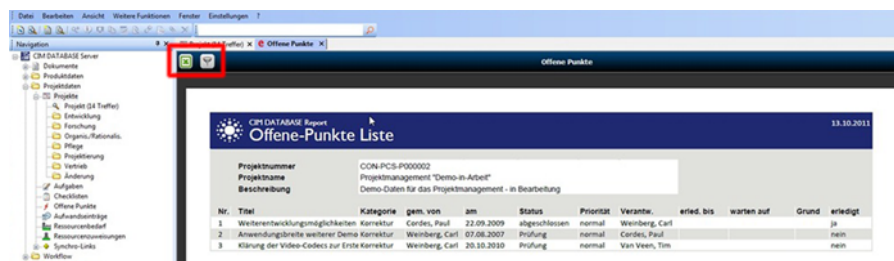


Abb. 2.59: Excel: PowerReports - Reportauswahl als eLink-Panel

Bemerkung: Damit die Reportdarstellung als eLink-Panel auf dem Arbeitsplatz-Rechner funktioniert, muss dort Adobe Acrobat installiert sein.

Bemerkung: Über den CIM Database/WEB-Client können PowerReports nur mit dem Ausführungstyp *Server (asynchron)* generiert werden.

Auswahlkonfiguration

Ab CDB 9.8 *SP1* lässt sich für jeden Report getrennt konfigurieren, ob ein PowerReport z.B. immer als eLink-Panel Panel direkt im PC-Client angezeigt oder aber in Excel geöffnet werden soll, oder ob er aus Performance-

gründen erst bei Fertigstellung im Excel- und PDF-Format als E-Mail gesendet werden soll.

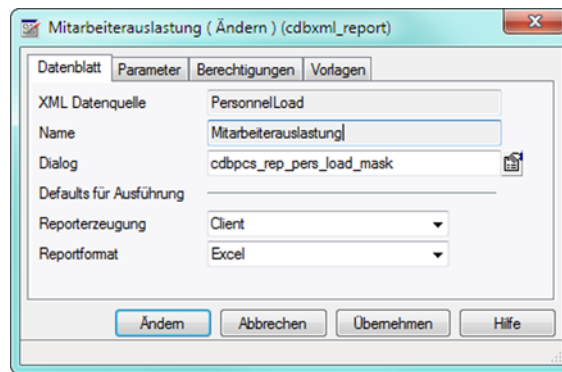


Abb. 2.60: Excel: PowerReports - Reportauswahlkonfiguration

Bemerkung: *Reporterzeugung* und *Reportformat* können auch anschließend noch im Reportauswahldialog geändert werden, jedoch sind diese Felder dort standardmäßig ausgeblendet.

3.1 Einführung

- *Outlook-Oberfläche:*

Die Outlook-Anwendung wird durch die Installation um die OfficeLink-Menü- und Symbolleiste erweitert.



Abb. 3.1: Outlook: OfficeLink-Symbolleiste

- *Email/Anhang neu anlegen:*

Die System Microsoft Office Integration bietet für Outlook die Möglichkeit, einzelne Nachrichten direkt aus Outlook heraus in Form einer MSG -Datei als Dokument im System zu archivieren. MSG -Dateien repräsentieren in Outlook ein vollständiges Nachrichtenobjekt inklusive sämtlicher Anlagen (Attachments). Die Erweiterung bietet dem Benutzer weiterhin die Möglichkeit, gegebenenfalls enthaltene Anlagen der jeweiligen (selektierten) Nachrichten separat im System zu archivieren und/oder die Nachricht selbst. Der Benutzer bekommt dabei - in Form eines Dialoges - die Option, auszuwählen, welche Elemente separat archiviert werden sollen. Zusätzlich kann bei jedem kompletten Archivierungsvorgang entschieden werden, ob separat archivierte Anlagen im Vorfeld aus der Nachricht (MSG -Datei) extrahiert werden sollen, bevor diese selbst im System archiviert wird. Weiterhin wird im Rahmen des Archivierungsvorgangs dem Benutzer die Option geboten, ob im Anschluss an eine erfolgreiche Archivierung die Nachrichtendatei automatisiert in Outlook zu löschen ist.

Für jede einzelne zu archivierende Datei (sowohl Nachricht als auch Nachrichtenanlagen) wird in CIM Database/WIN die Neuanlagemaske aufgeblendet. Bei der Neuanlagemaske für die Nachricht besteht die Möglichkeit, durch Konfiguration vom System einzelne Felder mit Informationen (wie beispielsweise *Betreff* oder *Absender*) der zugrunde liegenden Nachricht vorzubelegen.

Bemerkung: Falls Anhänge nicht als Dokumente angelegt, sondern in CIM Database/WIN als Dateien an die Mail-Dokumente gehängt werden, wird die Neuanlagemaske für diese Dateien automatisch ausgefüllt und übersprungen.

Bemerkung: Die Zuordnungskonfiguration der Metadaten einer Nachricht und den jeweiligen System-Attributen einer Neuanlagemaske nimmt der System-Administrator vor.

- *Dokument anhängen:*

Das Add-In bietet darüber hinaus die Möglichkeit beim Verfassen einer neuen Nachricht Dokumente aus dem System direkt der neuen Nachricht als Anlage anzuhängen. Dieser Vorgang wird über eine Symbolleiste des Nachrichtenfensters initiiert.

- *Einstellungen:*

Auf die Einstellungsmöglichkeiten im Optionsdialog wird im OfficeLink Administrationsbuch eingegangen. Wenden Sie sich bei Fragen an ihren System-Administrator.

3.2 E-Mails und Anlagen

3.2.1 Nachrichten neu im System anlegen

Die Neuanlage einer Nachricht aus Outlook im System geschieht allgemein in folgenden Schritten:

1. Wählen Sie die zu archivierenden Nachrichten aus indem Sie sie markieren.

Bemerkung: Die [*Email/Anhang neu anlegen*] -Schaltfläche wird aktiviert, wenn mindestens eine Nachricht markiert ist. Es können also auch mehrere Elemente gleichzeitig markiert und angelegt werden.

2. Klicken Sie auf die aktivierte Schaltfläche [*Email/Anhang neu anlegen*] .

Es wird nach einer bereits laufenden CIM Database/WIN-Instanz gesucht. Sollte kein CIM Database/WIN gestartet sein, so wird automatisch versucht, eine neue Programminstanz zu öffnen.

Bemerkung: Es wird grundsätzlich empfohlen, im Vorfeld bereits eine Instanz von CIM Database/WIN zu starten, bevor mit Archivierungsvorgängen begonnen wird.

3. Durch das Drücken der [*Email/Anhang neu anlegen*] -Schaltfläche erscheint ein *Auswahldialog*. Dieser Dialog bietet Optionen an, anhand derer die selektierten Nachrichten archiviert werden sollen. Im Email-Knoten sind nochmals Absender, Betreff, Datum und Uhrzeit mit angegeben. Wenn die zu archivierende Nachricht Anlagen enthält, dann werden diese in der Tabelle des Dialoges untergeordnet mitaufgelistet.

Wichtig: Nachrichten die keine Anlagen enthalten, werden nicht im Auswahldialog aufgelistet. Sie werden gemäß der *Standardaktion für Emails* von der System-Integration gehandhabt (siehe Einstellungsdialog). Falls keine der selektierten Nachrichten Anhänge enthält, dann wird der Auswahldialog übersprungen.

4. Dieser Dialog bittet die Möglichkeit sich zu entscheiden, ob die Anlagen einer Nachricht aus der Nachricht extrahiert, d.h. separat im System neu angelegt werden sollen.

Der Benutzer hat die Möglichkeit, die Elemente in der Dialogtabelle zu selektieren. Dabei ist es möglich, ein oder mehrere Elemente auszuwählen. Ebenso können die sich am oberen Rand des Dialogs befindlichen Steuerelemente [*Alle Emails auswählen*] und [*Alle Anhänge auswählen*] benutzt werden, falls der Benutzer eine Aktion für jeweils alle Emails oder Anhänge auswählen/ändern möchte. Geändert werden die Aktionen entweder durch einfaches Klicken mit der Maus auf das Aktions-Symbol, oder durch das Kontextmenü, welches erscheint sobald man mit der rechten Maustaste auf eine Auswahlmarkierung klickt.

Bemerkung: Das Extrahieren/Löschen der Anlagen wirkt sich ausschließlich auf die im System zu archivierende Nachrichtenkopie aus. Die originale Nachricht in Outlook bleibt dabei unverändert, es sei denn, der Benutzer wählt für die jeweilige Nachricht die Option, sie nach dem Archivierungsvorgang automatisch in Outlook zu löschen.

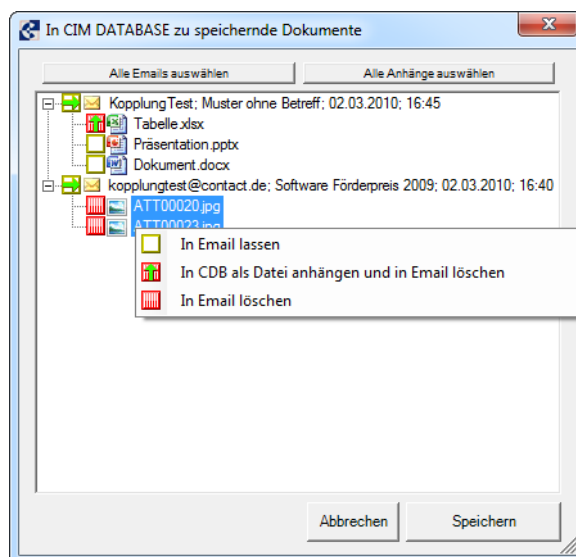


Abb. 3.2: Outlook: Nachrichten mit Anlagen im Archivierungsauswahldialog

Bemerkung: Falls für Emails oder Anhänge immer die gleiche Aktion ausgeführt wird, kann dies für ein schnelleres Arbeiten im Einstellungsdialog vorkonfiguriert werden. Dort kann der Auswahldialog zudem gänzlich deaktiviert werden (siehe OfficeLink Administrationshandbuch). In HTML-Emails verwendete Bilder werden unabhängig von den Einstellungen mit der Standardaktion "In Email lassen" vorbelegt.

Bemerkung: Grau dargestellte Anhänge stellen in die Email eingebettete Objekte dar. Für diese Anhänge kann ausschließlich die Aktion "In Email lassen" verwendet werden. Eine Auswahl der Aktion ist nicht möglich. Beispiel: Bilder in einer Email im Rich Text Format (RTF).

5. Nach der getroffenen Auswahl, wird der Dialog über die Schaltfläche [*Speichern*] bestätigt. Dann erscheint für jede zu archivierende Datei (Nachricht/Anlage) die Dokumentneuanlagemaske im System. Die einzelnen Felder der Neuanlagemaske sind entsprechend der Konfiguration vorbelegt. Die ursprünglichen Masken-Standardeinstellungen werden dabei mit den konfigurierten Werten überschrieben. Jede Maske ist anschließend durch den Benutzer, mit den jeweils notwendigen Parametern ergänzend, zu attributieren und zu bestätigen. Falls Anhänge nicht als Dokumente angelegt, sondern in CIM Database/WIN als Dateien an die Mail-Dokumente gehängt werden, wird die Neuanlagemaske für diese Dateien automatisch ausgefüllt und übersprungen.

Bemerkung: Je nach getroffenen Einstellungen werden gegebenenfalls die Anlagen aus der zu archivierenden Nachricht extrahiert und/oder die Originalnachricht in Outlook abschließend automatisch gelöscht.

3.2.2 System-Dokumente als Nachrichtenanlage anhängen

Das Anhängen von System-Dokumenten an eine zu verfassende Nachricht aus Outlook heraus geschieht allgemein in folgenden Schritten:

1. Erstellen Sie eine neue E-Mail-Nachricht über das Outlook Hauptmenü oder die entsprechende Schaltfläche der Standard-Symbolleiste.
2. Durch Drücken dieser Schaltfläche wird eine System-Dokumentenrecherche initiiert und es wird die Dokumentensuchmaske aufgeblendet. Hier parametrisieren Sie die Suchmaske wie gewohnt mit entsprechen-

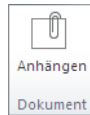


Abb. 3.3: Outlook: Steuerelement 'Anhängen'

den Suchkriterien. Nach der Bestätigung der Maske erhalten Sie einen Dialog mit der resultierenden Dokumenten-Trefferliste.

Wählen Sie ein oder mehrere Dokumente aus der Trefferliste, welche Sie versenden wollen.

Da Dokumente mit mehreren Dateien behaftet sein können, kommt es gegebenenfalls zu einem weiteren Auswahldialog. Das Auswahlverfahren bei Dokumenten mit zusätzlich anhaftenden Dateien kann in dem Einstellungsdialog weiter automatisiert werden (siehe OfficeLink Administrationshandbuch).

Bemerkung: Für die ausgewählten System-Dokumente wird das Export-Recht geprüft.

3.2.3 Nachrichten aus der Dokumentenexport-Funktion von CIM Database/WIN

Bei der System-Operation 'Dokumente als Mail exportieren' wird neben den selektierten Dokumenten auch eine cdbinfo-Datei im XML-Format ausgeleitet, welche als zusätzliche Anlage der zu versendenden Nachricht mitgegeben wird. Nachrichten, die ursprünglich auf diesem Wege versendet wurden können ebenso archiviert werden. Dabei erkennt die Kopplung anhand der cdbinfo-Datei, welche Anlagen aus dem System stammen und bietet diese nicht erneut zur Archivierung an. Nachdem die ausgehende Nachricht archiviert wurde, werden die Referenzen zwischen der Nachricht und den bereits im System befindlichen Anlagen geschrieben.

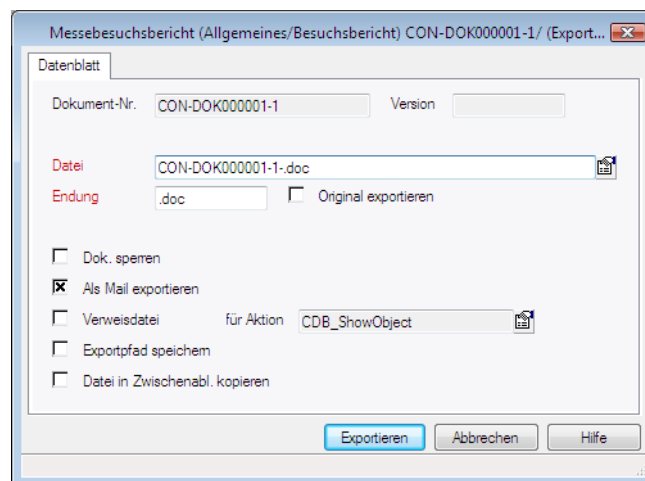


Abb. 3.4: Outlook: 'Als Mail exportieren' in CIM Database/WIN

PowerPoint

4.1 Einführung

- *PowerPoint-Oberfläche:*

Die PowerPoint-Anwendung wird durch die Installation um die OfficeLink-Symboleiste erweitert.

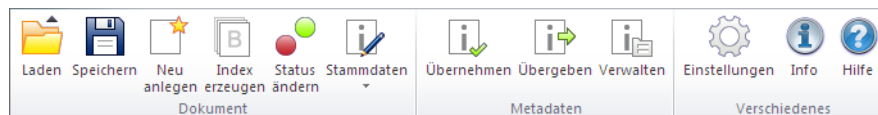


Abb. 4.1: PowerPoint: OfficeLink-Menüleiste

- *Standardoperationen:*

Die System Microsoft Office Integration bietet für PowerPoint die Möglichkeit, Dokumente im System zu suchen, zur Bearbeitung zu öffnen und daraufhin Änderungen in der Datenbank zu speichern. Weiterhin können Dokument-Status oder -Eigenschaften geändert, oder direkt aus PowerPoint heraus neue Dokumente im System angelegt werden.

- *Datenabgleich:*

Die Kopplung bietet die Möglichkeit des Datenabgleichs mit dem System. Voraussetzung dafür ist die Verwendung von vorbereiteten Dokumentvorlagen, die entsprechende System-Variablen enthalten (siehe [Vorlagen zum Datenabgleich erstellen](#) (Seite 54)). Der Datenabgleich wird immer für das aktive Dokument durchgeführt. Das aktive Dokument muss aus dem System zur Bearbeitung geöffnet worden sein.

- *Einstellungen:*

Auf die Einstellungsmöglichkeiten im Optionsdialog wird im OfficeLink Administrationsbuch eingegangen. Wenden Sie sich bei Fragen an ihren System-Administrator.

4.2 Dokumente bearbeiten

4.2.1 Dokumentensuche im System

Die Schaltfläche [*Laden*] öffnet die Dokumentensuchmaske im System. Die vorgegebene Suchbedingung schränkt die Suche automatisch auf PowerPoint Dokumente ein. Nach Eingabe weiterer Suchkriterien und Ausführen der Suche öffnet sich die Ergebnistrefferverliste mit Vorschaufenster. Durch Auswahl eines oder mehrerer Dokumente werden diese zur Bearbeitung geöffnet.

4.2.2 Im System speichern

Die Schaltfläche [*Speichern*] speichert die Änderungen am aktiven Dokument direkt im System.

4.2.3 Neuanlage im System

Die Schaltfläche [*Neu anlegen*] führt für das aktive Dokument eine Dokumentenneuanlage im System durch. Dabei öffnet sich zunächst die Maske zur Dokumentenneuanlage. Nach dem Ausfüllen und Bestätigen der Neuanlagemaske wird das Dokument in das System übertragen und abschließend aus dem System neu zur Bearbeitung geöffnet. Das Originaldokument wird bei erfolgreichem Verlauf der Neuanlage geschlossen. Im Fehlerfall, beispielsweise aufgrund fehlender Berechtigungen im System, bleibt das Dokument geöffnet.

Bemerkung: Da mit Microsoft PowerPoint mehrere Dateiformate vom System unterstützt werden, wird dort zunächst das Speicherformat ermittelt, bevor die Neuanlagemaske der System-Dokumentenverwaltung geöffnet wird: *Als Standardformat benutzen:* Durch Setzen dieser Checkbox wird das ausgewählte Format als Standard-speicherformat festgelegt. Bei der zukünftigen Neuanlage von Dokumenten wird der Auswahldialog nicht mehr angezeigt und stattdessen das Standardspeicherformat verwendet. Das Standardspeicherformat kann jederzeit im Einstellungsdialog geändert oder entfernt werden. *PowerPoint Open XML Dateiformat und Makros:* In dem PowerPoint Open XML Dateiformat .pptx können keine Makros gespeichert werden. Beim Versuch Dokumente mit Makros in diesem Format zu speichern, wird eine entsprechende Meldung angezeigt.

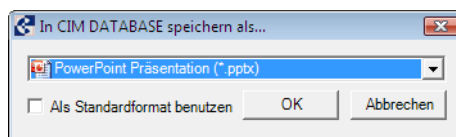


Abb. 4.2: Microsoft PowerPoint: Speicherformatauswahldialog

4.2.4 Index erzeugen im System

Die Schaltfläche [*Index erzeugen*] erzeugt für das aktive Dokument einen neuen Index im System. Dabei öffnet sich zunächst die Maske zur Indexerzeugung. Nach dem Ausfüllen und Bestätigen der Indexmaske werden die aktuellen Änderungen in dem neuen Dokumentenindex im System gesichert.

4.2.5 Dokumentenstatus ändern

Die Schaltfläche [*Status ändern*] führt für das aktive Dokument eine System-Statusänderungsoperation durch.

Bemerkung: Da der Anwender durch die Statusänderung ggf. die Rechte im System verliert, das Dokument erneut zu speichern, speichert die Kopplung das Dokument automatisch, bevor die Statusänderungsoperation aufgerufen wird. Dabei wird ggf. eine entsprechende Meldung angezeigt, die bestätigt werden muss.

4.2.6 Stammdatenänderungsdialog im System öffnen

Die Schaltfläche [*Stammdaten*] öffnet im System den Dialog zum Ändern der Stammdaten für das aktive Dokument. Diese Schaltfläche ist geteilt. Über den unteren Teil kann im System statt des Änderungsdialoges ein Informationsdialog für das aktive Dokument geöffnet werden.

4.3 Daten abgleichen

4.3.1 Metadatenübernahme aus dem System

Durch Betätigung der Schaltfläche [*Übernehmen*] werden die Werte aller System-Variablen mit lesendem Zugriff mit den entsprechenden Werten der zugehörigen System-Attribute aktualisiert. Vom Anwender im Dokument geänderte System- Variablen mit lesendem Zugriff werden dabei überschrieben. Falls ein Fehler im Ausdruck einer Variablen vorliegt, das referenzierte Objekt nicht existiert oder die Beziehung nicht eindeutig ist, wird der Wert auf ein Leerzeichen gesetzt.

4.3.2 Metadatensynchronisation mit dem System

Durch Betätigung der Schaltfläche [*Übergeben*] werden die Werte aller Variablen mit schreibenden Zugriff auf Änderungen überprüft. Wurden Werte im Dokument verändert, werden die betroffenen Objekte nacheinander im System geöffnet. Die Änderungen werden in den Änderungsdialogen vorgeblendet und müssen für jedes Objekt bestätigt werden. Nach Abschluss der Datenübernahme in das System wird eine Metadatenübernahme für System-Variablen mit lesendem Zugriff durchgeführt.

4.3.3 Vorlagen zum Datenabgleich erstellen

Dieser Abschnitt beschreibt die Erstellung von Dokumentvorlagen mit den System- Variablen zum Datenabgleich.

- Abgleich von Dokumentmetadaten (z.B. Titel, Dokument-Nr. ...)
- Abgleich von 1:1 referenzierten Objekten (z.B. Projektname des zugeordneten Projekts)

Formatbeschreibung für System-Variablen

System-Variablen in PowerPoint Dokumenten haben folgendes Format:
 cdb.<Mode>.<Beziehung>.<Attribut>.<Kardinalität>

Mode Der Mode gibt an, ob eine Variable aus dem System ausschließlich gelesen oder auch nach Änderung im Dokument im System aktualisiert werden darf. Gültige Werte für den Modus sind:

- *r*

Nur Datenübernahme aus dem System möglich. Der übernommene Wert kann im Dokument verändert werden, jedoch nicht an das System übergeben werden. Bei erneuter Datenübernahme wird der geänderte Wert im Dokument wieder durch den System-Wert ersetzt.

- *w*

Nur Datenübergabe in das System möglich. Durch diesen Modus kann sichergestellt werden, dass ein Dokumentfeld durch eine Datenübernahme nicht verändert werden kann, eine Aktualisierung im System aber möglich ist (sinnvoll bei Feldern, die durch das Dokument berechnet werden). Dieser Modus kann nicht bei 1:N Beziehungen verwendet werden.

- *rw*

Datenübernahme aus dem System und Aktualisierung im System nach Änderung im Dokument möglich. Dieser Modus kann nicht bei 1:N Beziehungen verwendet werden.

- **s*

Wenn oben genannten Modewerten ein *s* hinzugefügt wird (d.h. *rs*, *ws* oder *rws*), dann gelten zwar die gleichen Regeln wie bei den jeweiligen beschriebenen Werten ohne *s*, jedoch laufen die Lese- bzw. Schreibaktionen vollständig serverseitig ab. Dabei kann optional das Feld *Parameter* verwendet werden. U.U. bedarf es allerdings für die Verwendung der serverseitigen Bearbeitung von Variablen vorbereitend administratives Customizing, welches im OfficeLink Administrationshandbuch erläutert wird.

Beziehung Die Beziehung gibt ausgehend vom Dokument den Pfad zu den durch die Variable repräsentierten Daten im System an. Als Beziehung können folgende Ausdrücke verwendet werden:

- *this*

Der konstante Ausdruck 'this' wird verwendet, um Attribute des Dokuments selbst anzugeben.

- *Name einer im System konfigurierten Beziehung*

Über eine Beziehung können Attribute von referenzierten Objekten übernommen und abgeglichen werden.

- *BY_ZNUM_ZIDX_FROM_<table_name>*

Verwendung einer impliziten Beziehung über die Attribute z_nummer und z_index. Hierzu wird der Ausdruck BY_ZNUM_ZIDX_FROM_<table_name> angegeben. table_name gibt dabei die Datenbanktabelle an, aus der ein Objekt über z_nummer und z_index implizit referenziert wird. Die angegebene Datenbanktabelle muss die Attribute z_nummer und z_index enthalten und die Kombination aus z_nummer und z_index muss eindeutig sein.

Attribut Das Attribut spezifiziert ein Attribut eines durch die Beziehung referenzierten System-Objekts.

Sachmerkmale als Attribute von Artikeln:

Wird eine Artikelbeziehung als Beziehung angegeben, können auch die Sachmerkmale des Artikels als Attribut angegeben werden. Dies kann qualifiziert über die Merkmalkennung oder unqualifiziert mittels Zählnummer erfolgen:

- *Sachmerkmale qualifiziert:*

`__SML_<mm_mk>`

PowerPoint: Datenabgleich - Sachmerkmale qualifiziert

Sachmerkmal mit der Kennung 'L' des zugeordneten Artikels

```
``cdb.r.cdb_doc_to_part.__SML_L``
```

- *Sachmerkmale unqualifiziert:*

`__SMLN<nummer>` für Merkmalskennungen

`__SMLV<nummer>` für Merkmalswerte

Merkmalkennung und der zugehörige Wert müssen dabei dieselbe Nummer haben.

PowerPoint: Datenabgleich - Sachmerkmale unqualifiziert

Kennung und Wert des erstes Sachmerkmals des zugeordneten Artikels

```
``cdb.r.cdb_doc_to_part.__SMLN1``
``cdb.r.cdb_doc_to_part.__SMLV1``
```

Wichtig: SML Attribute können nicht schreibend synchronisiert werden!

Kardinalität Gibt die Kardinalität der Beziehung an. Wenn keine Kardinalität angegeben wurde, wird von der Kardinalität 1 ausgegangen. Bei der Erstellung neuer Dokumentvorlagen sollte die Kardinalität explizit angegeben werden. Folgende Werte können für die Kardinalität verwendet werden:

- *1*

Für Beziehungen vom Typ *this*, *BY_ZNUM_ZIDX_FROM_<table_name>* und für 1-stellige homogene Beziehungen.

Parameter Über dieses Feld können bei der serverseitigen Bearbeitung von Variablen zusätzliche für die jeweilige Variable spezifische konstante Werte übergeben werden. Das können beispielsweise Identifikatoren sein, welche das Verlinken von Ergebnissen aus 1:N Beziehungen mit einzelnen Dokumentfeldern statt mit dynamischen Listen ermöglichen. Beispielsweise wird der folgenden Variable der Parameter *PRIMARY* hinzugefügt, welches einem zuvor speziell dafür bereitgestelltem Userexit übermittelt, dass man den Ursprungsnamen der Primärdatei des Dokuments haben möchte.

```
cdb.rs.document2cdb_file.cdbf_original_name.N.string.PRIMARY
```

Definition von System-Variablen

Die Definition von System-Variablen erfolgt mit einem Assistenten, der über die Schaltfläche [*Verwalten*] der OfficeLink-Symbolleiste aufgerufen wird. Damit die Schaltfläche verfügbar ist, muss sie zuvor in dem Einstellungsdialog aktiviert worden sein. Der Dialog bietet auf der linken Seite die Möglichkeit neue Variablen im System-Format anzulegen (siehe *Formatbeschreibung für System-Variablen* (Seite 54)). Auf der rechten Seite werden alle im Dokument enthaltenen System-Variablen angezeigt. Durch Markierung einer Variablen wird gleichzeitig das entsprechende Textfeld in der PowerPoint Dokument markiert. Durch Betätigung der Schaltfläche [*Variable löschen*] kann die markierte Variable aus dem Dokument entfernt werden.

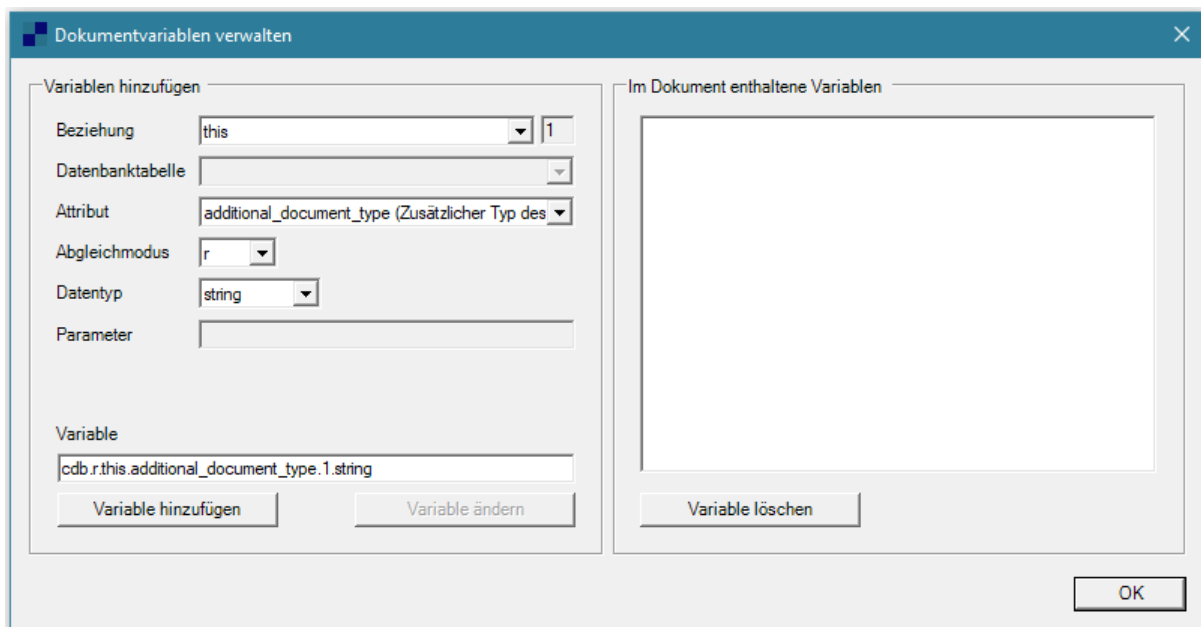


Abb. 4.3: PowerPoint: Datenabgleich - Dialog zur Erstellung von System-Variablen

- *Beziehung:*

Die Beziehung gibt ausgehend vom Dokument den Pfad zu einem Attribut eines oder mehrerer referenzierter System-Objekte an. Der Katalog bietet alle im System konfigurierten Dokumentenbeziehungen an, bei denen das Dokument das Startobjekt darstellt. Für Attribute des Dokuments selbst, muss in diesem Feld *this* eingetragen werden.

Zur Verwendung einer impliziten Beziehung, die über die Schlüsselattribute Dokumentennummer und Index hergestellt wird, muss in diesem Feld *BY_ZNUM_ZIDX_FROM_* ausgewählt werden und im Feld Datenbanktabelle eine Relation als Datenquelle ausgewählt werden.

- *Datenbanktabelle:*

Eine Datenbanktabelle muss nur dann ausgewählt werden, wenn im Feld Beziehung der spezielle Eintrag *BY_ZNUM_ZIDX_FROM_* für implizite Beziehungen ausgewählt wurde. In der ausgewählten Relation muss die Kombination der Attribute *z_nummer* und *z_index* eindeutig sein.

- *Attribut:*

Das Attribut spezifiziert ein Attribut eines durch die Beziehung referenzierten System-Objekts.

- *Abgleichmodus:*

Der Modus gibt an, ob eine Variable aus dem System ausschließlich gelesen oder auch nach Änderung im Dokument im System aktualisiert werden darf:

r: nur lesend

w: nur schreibend

rw: lesender und schreibender Zugriff im System

rs: nur serverseitig lesend

ws: nur serverseitig schreibend

rhs: serverseitig lesender und schreibender Zugriff im System

- *Parameter:*

Optional nur für das serverseitige Bearbeiten von Variablen verwendeter Wert.

- *System Variable:*

In diesem Feld wird der aus den anderen Feldern erzeugte Ausdruck für die zu erstellenden System-Variable angezeigt. Das manuelle Ändern dieses Ausdrucks ist nur zur Angabe von Sachmerkmalen bei Artikelbeziehungen erforderlich.

Beim Hinzufügen der Variablen durch Betätigung der Schaltfläche [*Variable hinzufügen*] wird der in dem Feld System-Variable angezeigte Ausdruck überprüft und bei Ungültigkeit eine entsprechende Fehlermeldung angezeigt. Ansonsten wird die Variable als neue System-Variable in dem aktiven Dokument angelegt und mit dem Wert <Attribut> als Platzhalter initialisiert.

Wichtig: Die System-Variablen können nur in einfachen PowerPoint 'Formen' platziert werden. Desweiteren kann nur jeweils eine System-Variable pro PowerPoint Form definiert werden. Es erscheint eine Fehlermeldung, falls versucht wird eine neue Variable zu definieren, ohne dass eine einfache Form aktiviert ist, oder falls die zur Zeit aktivierte Form schon eine Variable enthält.

Ändern von System-Variablen

Die Änderung der Textfelder, welche System-Variablen beinhalten, erfolgt durch Mausklick in das zu ändernde Feld. Der Cursor befindet sich dann innerhalb des Feldes und ermöglicht eine freie Bearbeitung.

Verwendung von System-Variablen im Dokument

System-Variablen sollten im Dokument für den Anwender kenntlich gemacht werden, um versehentliches Löschen oder Überschreiben zu vermeiden. Der Schutz von Dokumentenvariablen ist nur bedingt möglich. Eine Möglichkeit ist die Erstellung von Formularen.

Bemerkung: Beim Datenabgleich von Variablen in das System werden Attributänderungen auch dann vorgeblendet, wenn das Feld laut Maskenkonfiguration durch den Anwender nicht interaktiv änderbar oder nur via Katalogauswahl änderbar wäre. Bei der Verwendung von System-Variablen mit schreibenden Zugriff ist dies zu berücksichtigen. Gegebenenfalls ist die Gültigkeit der Werte durch eine entsprechende Prüfung im User Exit abzusichern.

Automatisches Aktualisieren von Metadaten

Die Metadaten aller System-Variablen mit lesendem Zugriff werden beim Laden eines Dokuments aus CDB automatisch übernommen. Die Bedingungen dafür sind, dass das Dokument:

1. im Client im “Bearbeiten” Modus geöffnet wurde (oder direkt aus Office heraus über “Laden”)
2. eine “Zeichnung” ist (d.h. die Eigenschaft “vorlage_kz” hat)
3. keine Vorlage ist (d.h. “vorlage_kz=0”)

5.1 Einführung

- *Project-Oberfläche:*

Die Project-Anwendung wird durch die Installation um die OfficeLink-Symbolleiste erweitert.



Abb. 5.1: Project: OfficeLink-Menüleiste

- *Standardoperationen:*

Die System Microsoft Office Integration bietet für Project die Möglichkeit, Dokumente im System zu suchen, zur Bearbeitung zu öffnen und daraufhin Änderungen in der Datenbank zu speichern. Weiterhin können Dokument-Status oder -Eigenschaften geändert, oder direkt aus Project heraus neue Dokumente im System angelegt werden.

- *Projekt-Synchronisation:*

Die System Integration für Microsoft Office ermöglicht es, aus Microsoft Project heraus Projekte zu bearbeiten, welche vom System PCS verwaltet werden. Dabei werden Projektdaten teilweise bidirektional abgeglichen.

- *Einstellungen:*

Auf die Einstellungsmöglichkeiten im Optionsdialog wird im OfficeLink Administrationsbuch eingegangen. Wenden Sie sich bei Fragen an ihren System-Administrator.

5.2 Dokumente bearbeiten

5.2.1 Dokumentensuche im System

Die Schaltfläche [*Laden*] öffnet die Dokumentensuchmaske im System. Die vorgegebene Suchbedingung schränkt die Suche automatisch auf Project Dokumente ein. Nach Eingabe weiterer Suchkriterien und Ausführen der Suche öffnet sich die Ergebnistrefferliste mit Vorschaufenster. Durch Auswahl eines oder mehrerer Dokumente werden diese zur Bearbeitung geöffnet.

5.2.2 Im System speichern

Die Schaltfläche [*Speichern*] speichert die Änderungen am aktiven Dokument direkt im System.

5.2.3 Neuanlage im System

Die Schaltfläche [*Neu anlegen*] führt für das aktive Dokument eine Dokumentenneuanlage im System durch. Dabei öffnet sich zunächst die Maske zur Dokumentenneuanlage. Nach dem Ausfüllen und Bestätigen der Neuanlagemaske wird das Dokument in das System übertragen und abschließend aus dem System neu zur Bearbeitung geöffnet. Das Originaldokument wird bei erfolgreichem Verlauf der Neuanlage geschlossen. Im Fehlerfall, beispielsweise aufgrund fehlender Berechtigungen im System, bleibt das Dokument geöffnet.

5.2.4 Index erzeugen im System

Die Schaltfläche [*Index erzeugen*] erzeugt für das aktive Dokument einen neuen Index im System. Dabei öffnet sich zunächst die Maske zur Indexerzeugung. Nach dem Ausfüllen und Bestätigen der Indexmaske werden die aktuellen Änderungen in dem neuen Dokumentenindex im System gesichert.

5.2.5 Dokumentenstatus ändern

Die Schaltfläche [*Status ändern*] führt für das aktive Dokument eine System-Statusänderungsoperation durch.

Bemerkung: Da der Anwender durch die Statusänderung ggf. die Rechte im System verliert, das Dokument erneut zu speichern, speichert die Kopplung das Dokument automatisch, bevor die Statusänderungsoperation aufgerufen wird. Dabei wird ggf. eine entsprechende Meldung angezeigt, die bestätigt werden muss.

5.2.6 Stammdatenänderungsdialog im System öffnen

Die Schaltfläche [*Stammdaten*] öffnet im System den Dialog zum Ändern der Stammdaten für das aktive Dokument. Diese Schaltfläche ist geteilt. Über den unteren Teil kann im System statt des Änderungsdialoges ein Informationsdialog für das aktive Dokument geöffnet werden.

5.3 Projekt-Synchronisation

5.3.1 Funktionalität

Die Microsoft Project-Integration ermöglicht die Bearbeitung von Projektplänen, die durch das System PCS verwaltet werden, in Microsoft Project. Dazu wird Microsoft Project um die folgenden Funktionen erweitert:

- Veröffentlichen der Projektstruktur im System ausgehend vom Projektplan
- Aktualisieren bestimmter Aufgabenattribute mit aktuellen Informationen aus dem System
- Anzeige von Projekt- und Aufgabenübersichten aus dem System zu geladenen Projektplänen in Microsoft Project
- Anzeige von Aufgabeninformationsblatt im System von selektierten Aufgaben in Microsoft Project

Beim Veröffentlichen und Aktualisieren von Projektplänen findet ein teilweise bidirektionaler Abgleich von Projektdaten sowie Aufgaben statt. Dazu gehören u.a. Projektname, Projektterminierung, Aufgabenstruktur und -beziehungen sowie Meilensteine, so dass die gemeinsame Teilmenge aller Datenfelder zwischen dem System und Microsoft Project vollständig abgedeckt werden kann.

Wichtig: Es ist zudem erforderlich, weiterführende Hinweise im Projects Anwenderhandbuch nachzuschlagen.

5.3.2 Veröffentlichen von Microsoft Project Projektplänen im System PCS

Durch “Veröffentlichen” kann die im System PCS gespeicherte Projektstruktur durch den Microsoft Project Projektplan aktualisiert werden. Die Änderungen am Projektplan werden dabei in die dafür konfigurierten System-Relationen übertragen, neue Aufgaben werden entsprechend angelegt und gelöschte Aufgaben werden aus der Projektstruktur entfernt. Nach erfolgreichem Importieren bzw. Aktualisieren im System PCS besitzen beide Projektpläne den gleichen Stand.

Bei diesem Datenabgleich wird der geänderte Microsoft Project Projektplan im System gespeichert und zusätzlich im Microsoft Project XML Format ausgeleitet und als XML Datei neben dem Projektplan im System abgelegt. Anschließend wird automatisch im System ein Importvorgang über die soeben erstellte XML Datei ausgeführt. Weitere Hinweise über den sich nun öffnenden VorschauDialog finden Sie im Projects Anwenderhandbuch.

5.3.3 Aktualisieren von Aufgabenattributen aus dem System PCS

Beim “Attribute aktualisieren” wird der Projektplan mit bestimmten im System aktuell gespeicherten Aufgabenattributen aktualisiert. Ein Datentransfer in Richtung System findet hierbei nicht statt. Standardmäßig werden nur einige wenige Attribute aus dem System nach Microsoft Project übertragen:

Tab. 5.1: Aktualisierte Aufgabenattribute

MSP Aufgabenfeld	PCS Aufgabenfeld
Number10	Status
Text12	Referenz-ID
Text14	Verantwortlich
Text15	Status (Text)

Wie die Liste dieser aktualisierten Aufgabenattribute angepasst werden kann, wird im OfficeLink Administrationshandbuch beschrieben.

Wichtig: Es kann kein konsistenter beidseitiger Datenabgleich gewährleistet werden, wenn die Liste durch Aufgabenattribute erweitert wird, welche die Projektstruktur erneut indirekt verändern, wie z.B. Aufgabenstart, -dauer, -ende, Vorgänger, Nachfolger o.ä.

5.3.4 Allgemeine Hinweise

Bemerkung: Falls Sie einen Projektplan das erste Mal mit Hilfe der Microsoft Project-Integration bearbeiten, werden Abgleichsinformationen in der Microsoft Project-Datei gespeichert. Auch ohne manuelle Änderungen fragt daher Microsoft Project beim Beenden, ob Änderungen an der Datei gespeichert werden sollen.

Bemerkung: Projektübergreifende Anordnungsbeziehungen zwischen Aufgaben werden nicht synchronisiert. Sie können ein Projekt mit diesen Beziehungen zwar synchronisieren, jedoch sind sie in MS Project nicht sichtbar und werden beim Abgleich ignoriert. Wird eine Aufgabe in MS Project gelöscht, wird beim Abgleich die Aufgabe mit allen Anordnungsbeziehungen im System gelöscht. Es erscheint keine gesonderte Benachrichtigung.

6.1 Einführung

- *Visio-Oberfläche:*

Die Visio-Anwendung wird durch die Installation um die OfficeLink-Symbolleiste erweitert.

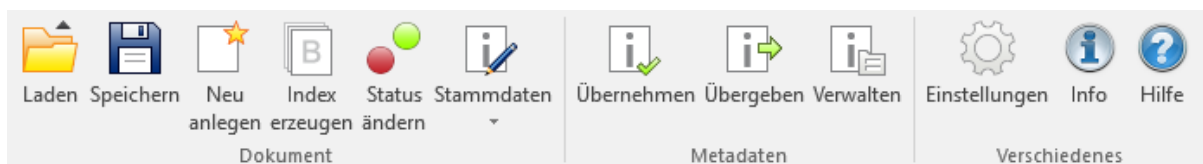


Abb. 6.1: Visio: OfficeLink-Menüleiste

- *Standardoperationen:*

Die System Microsoft Office Integration bietet für Visio die Möglichkeit, Dokumente im System zu suchen, zur Bearbeitung zu öffnen und daraufhin Änderungen in der Datenbank zu speichern. Weiterhin können Dokument-Status oder -Eigenschaften geändert, oder direkt aus Visio heraus neue Dokumente im System angelegt werden.

- *Datenabgleich:*

Die Kopplung bietet die Möglichkeit des Datenabgleichs mit dem System. Voraussetzung dafür ist die Verwendung von vorbereiteten Dokumentvorlagen, die entsprechende System-Variablen enthalten (siehe [Vorlagen zum Datenabgleich erstellen](#) (Seite 64)). Der Datenabgleich wird immer für das aktive Dokument durchgeführt. Das aktive Dokument muss aus dem System zur Bearbeitung geöffnet worden sein.

- *Einstellungen:*

Auf die Einstellungsmöglichkeiten im Optionsdialog wird im OfficeLink Administrationsbuch eingegangen. Wenden Sie sich bei Fragen an ihren System-Administrator.

6.2 Dokumente bearbeiten

6.2.1 Dokumentensuche im System

Die Schaltfläche [*Laden*] öffnet die Dokumentensuchmaske im System. Die vorgegebene Suchbedingung schränkt die Suche automatisch auf Visio Dokumente ein. Nach Eingabe weiterer Suchkriterien und Ausführen der Suche öffnet sich die Ergebnistrefferverliste mit Vorschaufenster. Durch Auswahl eines oder mehrerer Dokumente werden diese zur Bearbeitung geöffnet.

6.2.2 Im System speichern

Die Schaltfläche [*Speichern*] speichert die Änderungen am aktiven Dokument direkt im System.

6.2.3 Neuanlage im System

Die Schaltfläche [*Neu anlegen*] führt für das aktive Dokument eine Dokumentenneuanlage im System durch. Dabei öffnet sich zunächst die Maske zur Dokumentenneuanlage. Nach dem Ausfüllen und Bestätigen der Neuanlagemaske wird das Dokument in das System übertragen und abschließend aus dem System neu zur Bearbeitung geöffnet. Das Originaldokument wird bei erfolgreichem Verlauf der Neuanlage geschlossen. Im Fehlerfall, beispielsweise aufgrund fehlender Berechtigungen im System, bleibt das Dokument geöffnet.

Bemerkung: Da mit *Visio 2013* mehrere Dateiformate vom System unterstützt werden, wird dort zunächst das Speicherformat ermittelt, bevor die Neuanlagemaske der System-Dokumentenverwaltung geöffnet wird: *Als Standardformat benutzen*: Durch Setzen dieser Checkbox wird das ausgewählte Format als Standardspeicherformat festgelegt. Bei der zukünftigen Neuanlage von Dokumenten wird der Auswahldialog nicht mehr angezeigt und stattdessen das Standardspeicherformat verwendet. Das Standardspeicherformat kann jederzeit im Einstellungsdialog geändert oder entfernt werden. *Visio Open XML Dateiformat und Makros*: In dem Visio Dateiformat .vsdx können keine Makros gespeichert werden. Beim Versuch Dokumente mit Makros in diesem Format zu speichern, wird eine entsprechende Meldung angezeigt.

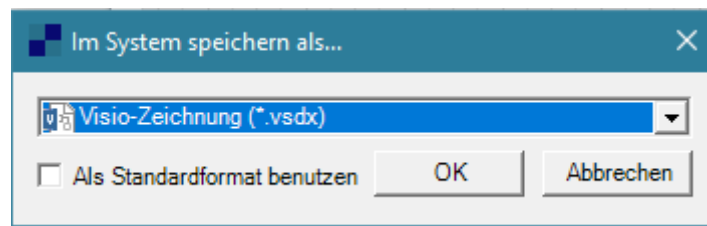


Abb. 6.2: Visio: Speicherformatauswahldialog

6.2.4 Index erzeugen im System

Die Schaltfläche [*Index erzeugen*] erzeugt für das aktive Dokument einen neuen Index im System. Dabei öffnet sich zunächst die Maske zur Indexerzeugung. Nach dem Ausfüllen und Bestätigen der Indexmaske werden die aktuellen Änderungen in dem neuen Dokumentenindex im System gesichert.

6.2.5 Dokumentenstatus ändern

Die Schaltfläche [*Status ändern*] führt für das aktive Dokument eine System-Statusänderungsoperation durch.

Bemerkung: Da der Anwender durch die Statusänderung ggf. die Rechte im System verliert, das Dokument erneut zu speichern, speichert die Kopplung das Dokument automatisch, bevor die Statusänderungsoperation aufgerufen wird. Dabei wird ggf. eine entsprechende Meldung angezeigt, die bestätigt werden muss.

6.2.6 Stammdatenänderungsdialog im System öffnen

Die Schaltfläche [*Stammdaten*] öffnet im System den Dialog zum Ändern der Stammdaten für das aktive Dokument. Diese Schaltfläche geteilt. Über den unteren Teil kann im System statt des Änderungsdialoges ein Informationsdialog für das aktive Dokument geöffnet werden.

6.3 Daten abgleichen

6.3.1 Metadatenübernahme aus dem System

Durch Betätigung der Schaltfläche [*Übernehmen*] werden die Werte aller System-Variablen mit lesendem Zugriff mit den entsprechenden Werten der zugehörigen System-Attribute aktualisiert. Vom Anwender im Dokument geänderte System- Variablen mit lesendem Zugriff werden dabei überschrieben. Falls ein Fehler im Ausdruck einer Variablen vorliegt, das referenzierte Objekt nicht existiert oder die Beziehung nicht eindeutig ist, wird der Wert auf ein Leerzeichen gesetzt.

6.3.2 Metadatensynchronisation mit dem System

Durch Betätigung der Schaltfläche [*Übergeben*] werden die Werte aller Variablen mit schreibenden Zugriff auf Änderungen überprüft. Wurden Werte im Dokument verändert, werden die betroffenen Objekte nacheinander im System geöffnet. Die Änderungen werden in den Änderungsdialogen vorgeblendet und müssen für jedes Objekt bestätigt werden. Nach Abschluss der Datenübernahme in das System wird eine Metadatenübernahme für System-Variablen mit lesendem Zugriff durchgeführt.

6.3.3 Vorlagen zum Datenabgleich erstellen

Dieser Abschnitt beschreibt die Erstellung von Dokumentvorlagen mit den System- Variablen zum Datenabgleich.

- Abgleich von Dokumentmetadaten (z.B. Titel, Dokument-Nr. ...)
- Abgleich von 1:1 referenzierten Objekten (z.B. Projektname des zugeordneten Projekts)

Formatbeschreibung für System-Variablen

System-Variablen in Visio Dokumenten haben folgendes Format: cdb.<Mode>.<Beziehung>.<Attribut>.<Kardinalität>

Mode Der Mode gibt an, ob eine Variable aus dem System ausschließlich gelesen oder auch nach Änderung im Dokument im System aktualisiert werden darf. Gültige Werte für den Modus sind:

- *r*
Nur Datenübernahme aus dem System möglich. Der übernommene Wert kann im Dokument verändert werden, jedoch nicht an das System übergeben werden. Bei erneuter Datenübernahme wird der geänderte Wert im Dokument wieder durch den System-Wert ersetzt.
- *w*
Nur Datenübergabe in das System möglich. Durch diesen Modus kann sichergestellt werden, dass ein Dokumentfeld durch eine Datenübernahme nicht verändert werden kann, eine Aktualisierung im System aber möglich ist (sinnvoll bei Feldern, die durch das Dokument berechnet werden). Dieser Modus kann nicht bei 1:N Beziehungen verwendet werden.
- *rw*
Datenübernahme aus dem System und Aktualisierung im System nach Änderung im Dokument möglich. Dieser Modus kann nicht bei 1:N Beziehungen verwendet werden.

Beziehung Die Beziehung gibt ausgehend vom Dokument den Pfad zu den durch die Variable repräsentierten Daten im System an. Als Beziehung können folgende Ausdrücke verwendet werden:

- *this*
Der konstante Ausdruck 'this' wird verwendet, um Attribute des Dokuments selbst anzugeben.
- *Name einer in CIM DATABASE konfigurierten Beziehung*
Über eine Beziehung können Attribute von referenzierten Objekten übernommen und abgeglichen werden.

- *BY_ZNUM_ZIDX_FROM_<table_name>*

Verwendung einer impliziten Beziehung über die Attribute *z_nummer* und *z_index*. Hierzu wird der Ausdruck *BY_ZNUM_ZIDX_FROM_<table_name>* angegeben. *table_name* gibt dabei die Datenbanktabelle an, aus der ein Objekt über *z_nummer* und *z_index* implizit referenziert wird. Die angegebene Datenbanktabelle muss die Attribute *z_nummer* und *z_index* enthalten und die Kombination aus *z_nummer* und *z_index* muss eindeutig sein.

Attribut Das Attribut spezifiziert ein Attribut eines durch die Beziehung referenzierten System-Objekts.

Sachmerkmale als Attribute von Artikeln:

Wird eine Artikelbeziehung als Beziehung angegeben, können auch die Sachmerkmale des Artikels als Attribut angegeben werden. Dies kann qualifiziert über die Merkmalkennung oder unqualifiziert mittels Zählnummer erfolgen:

- *Sachmerkmale qualifiziert:*

`__SML__<mm_mk>`

Visio: Datenabgleich - Sachmerkmale qualifiziert

Sachmerkmal mit der Kennung 'L' des zugeordneten Artikels

```
` `cdb.r.cdb_doc_to_part.__SML_L` `
```

- *Sachmerkmale unqualifiziert:*

`__SMLN<nummer>` für Merkmalskennungen

`__SMLV<nummer>` für Merkmalswerte

Merkmalkennung und der zugehörige Wert müssen dabei dieselbe Nummer haben.

Visio: Datenabgleich - Sachmerkmale unqualifiziert

Kennung und Wert des erstes Sachmerkmals des zugeordneten Artikels

```
` `cdb.r.cdb_doc_to_part.__SMLN1` `
` `cdb.r.cdb_doc_to_part.__SMLV1` `
```

Wichtig: SML Attribute können nicht schreibend synchronisiert werden!

Kardinalität Gibt die Kardinalität der Beziehung an. Wenn keine Kardinalität angegeben wurde, wird von der Kardinalität 1 ausgegangen. Bei der Erstellung neuer Dokumentvorlagen sollte die Kardinalität explizit angegeben werden. Folgende Werte können für die Kardinalität verwendet werden:

- 1

Für Beziehungen vom Typ *this*, *BY_ZNUM_ZIDX_FROM_<table_name>* und für 1-stellige homogene Beziehungen.

Definition von System-Variablen

Die Definition von System-Variablen erfolgt mit einem Assistenten, der über die Schaltfläche [*Verwalten*] der OfficeLink-Symbolleiste aufgerufen wird. Damit die Schaltfläche verfügbar ist, muss sie zuvor in dem Einstellungsdialog aktiviert worden sein. Der Dialog bietet auf der linken Seite die Möglichkeit neue Variablen im System-Format anzulegen (siehe *Formatbeschreibung für System-Variablen* (Seite 64)). Auf der rechten Seite werden alle im Dokument enthaltenen System-Variablen angezeigt. Durch Markierung einer Variablen wird

gleichzeitig das entsprechende Textfeld in der Visio Dokument markiert. Durch Betätigung der Schaltfläche [*Variable löschen*] kann die markierte Variable aus dem Dokument entfernt werden.

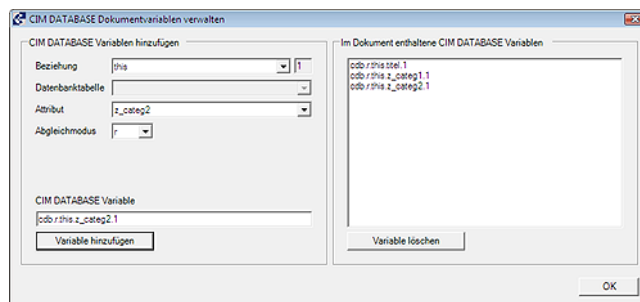


Abb. 6.3: Visio: Datenabgleich - Dialog zur Erstellung von System-Variablen

- **Beziehung:**

Die Beziehung gibt ausgehend vom Dokument den Pfad zu einem Attribut eines oder mehrerer referenzierter System-Objekte an. Der Katalog bietet alle im System konfigurierten Dokumentenbeziehungen an, bei denen das Dokument das Startobjekt darstellt. Für Attribute des Dokuments selbst, muss in diesem Feld `this` eingetragen werden.

Zur Verwendung einer impliziten Beziehung, die über die Schlüsselattribute Dokumentennummer und Index hergestellt wird, muss in diesem Feld `BY_ZNUM_ZIDX_FROM_` ausgewählt werden und im Feld Datenbanktabelle eine Relation als Datenquelle ausgewählt werden.

- **Datenbanktabelle:**

Eine Datenbanktabelle muss nur dann ausgewählt werden, wenn im Feld Beziehung der spezielle Eintrag `BY_ZNUM_ZIDX_FROM_` für implizite Beziehungen ausgewählt wurde. In der ausgewählten Relation muss die Kombination der Attribute `z_nummer` und `z_index` eindeutig sein.

- **Attribut:**

Das Attribut spezifiziert ein Attribut eines durch die Beziehung referenzierten System-Objekts.

- **Abgleichmodus:**

Der Modus gibt an, ob eine Variable aus dem System ausschließlich gelesen oder auch nach Änderung im Dokument im System aktualisiert werden darf:

r: nur lesend

w: nur schreibend

rw: lesender und schreibender Zugriff in CIM DATABASE

- **CIM DATABASE Variable:**

In diesem Feld wird der aus den anderen Feldern erzeugte Ausdruck für die zu erstellenden System-Variable angezeigt. Das manuelle Ändern dieses Ausdrucks ist nur zur Angabe von Sachmerkmalen bei Artikelbeziehungen erforderlich.

Beim Hinzufügen der Variablen durch Betätigung der Schaltfläche [*Variable hinzufügen*] wird der in dem Feld System-Variable angezeigte Ausdruck überprüft und bei Ungültigkeit eine entsprechende Fehlermeldung angezeigt. Ansonsten wird die Variable als neue System-Variable in dem aktiven Dokument angelegt und mit dem Wert `<Attribut>` als Platzhalter initialisiert.

Wichtig: Die System-Variablen können nur in einfachen Visio 'Formen' platziert werden. Desweiteren kann nur jeweils eine System-Variable pro Visio Form definiert werden. Es erscheint eine Fehlermeldung, falls versucht wird eine neue Variable zu definieren, ohne dass eine einfache Form aktiviert ist, oder falls die zur Zeit aktivierte Form schon eine Variable enthält.

Ändern von System-Variablen

Die Änderung der Textfelder, welche System-Variablen beinhalten, erfolgt durch Mausklick in das zu ändernde Feld. Der Cursor befindet sich dann innerhalb des Feldes und ermöglicht eine freie Bearbeitung.

Verwendung von System-Variablen im Dokument

System-Variablen sollten im Dokument für den Anwender kenntlich gemacht werden, um versehentliches Löschen oder Überschreiben zu vermeiden. Der Schutz von Dokumentenvariablen ist nur bedingt möglich. Eine Möglichkeit ist die Erstellung von Formularen.

Bemerkung: Beim Datenabgleich von Variablen in das System werden Attributänderungen auch dann vorgeblendet, wenn das Feld laut Maskenkonfiguration durch den Anwender nicht interaktiv änderbar oder nur via Katalogauswahl änderbar wäre. Bei der Verwendung von System-Variablen mit schreibenden Zugriff ist dies zu berücksichtigen. Gegebenenfalls ist die Gültigkeit der Werte durch eine entsprechende Prüfung im User Exit abzusichern.

Automatisches Aktualisieren von Metadaten

Die Metadaten aller System-Variablen mit lesendem Zugriff werden beim Laden eines Dokuments aus CDB automatisch übernommen. Die Bedingungen dafür sind, dass das Dokument:

1. im Client im "Bearbeiten" Modus geöffnet wurde (oder direkt aus Office heraus über "Laden")
2. eine "Zeichnung" ist (d.h. die Eigenschaft "vorlage_kz" hat)
3. keine Vorlage ist (d.h. "vorlage_kz=0")

7.1 Einführung

- *Word-Oberfläche:*

Die Word-Anwendung wird durch die Installation um die OfficeLink-Symbolleiste erweitert.

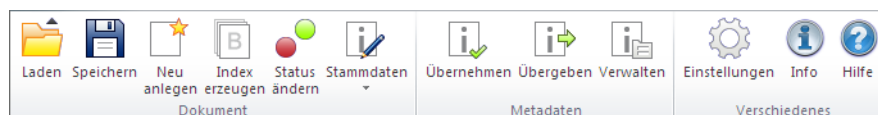


Abb. 7.1: Word: OfficeLink-Menüleiste

- *Standardoperationen:*

Die System Microsoft Office Integration bietet für Word die Möglichkeit, Dokumente im System zu suchen, zur Bearbeitung zu öffnen und daraufhin Änderungen in der Datenbank zu speichern. Weiterhin können Dokument-Status oder -Eigenschaften geändert, oder direkt aus Word heraus neue Dokumente im System angelegt werden.

- *Datenabgleich:*

Die Kopplung bietet die Möglichkeit des Datenabgleichs mit dem System. Voraussetzung dafür ist die Verwendung von vorbereiteten Dokumentvorlagen, die entsprechende System-Variablen enthalten (siehe [Vorlagen zum Datenabgleich erstellen](#) (Seite 70)). Der Datenabgleich wird immer für das aktive Dokument durchgeführt. Das aktive Dokument muss aus dem System zur Bearbeitung geöffnet worden sein.

- *Einstellungen:*

Auf die Einstellungsmöglichkeiten im Optionsdialog wird im OfficeLink Administrationsbuch eingegangen. Wenden Sie sich bei Fragen an ihren System-Administrator.

7.2 Dokumente bearbeiten

7.2.1 Dokumentensuche im System

Die Schaltfläche [*Laden*] öffnet die Dokumentensuchmaske im System. Die vorgegebene Suchbedingung schränkt die Suche automatisch auf Word Dokumente ein. Nach Eingabe weiterer Suchkriterien und Ausführen der Suche öffnet sich die Ergebnistrefferliste mit Vorschaufenster. Durch Auswahl eines oder mehrerer Dokumente werden diese zur Bearbeitung geöffnet.

7.2.2 Im System speichern

Die Schaltfläche [*Speichern*] speichert die Änderungen am aktiven Dokument direkt im System.

7.2.3 Neuanlage im System

Die Schaltfläche [*Neu anlegen*] führt für das aktive Dokument eine Dokumentenneuanlage im System durch. Dabei öffnet sich zunächst die Maske zur Dokumentenneuanlage. Nach dem Ausfüllen und Bestätigen der Neuanlagemaske wird das Dokument in das System übertragen und abschließend aus dem System neu zur Bearbeitung geöffnet. Das Originaldokument wird bei erfolgreichem Verlauf der Neuanlage geschlossen. Im Fehlerfall, beispielsweise aufgrund fehlender Berechtigungen im System, bleibt das Dokument geöffnet.

Bemerkung: Da mit Microsoft Word mehrere Dateiformate von dem System unterstützt werden, wird dort zunächst das Speicherformat ermittelt, bevor die Neuanlagemaske der System-Dokumentenverwaltung geöffnet wird: *Als Standardformat benutzen:* Durch Setzen dieser Checkbox wird das ausgewählte Format als Standard-speicherformat festgelegt. Bei der zukünftigen Neuanlage von Dokumenten wird der Auswahldialog nicht mehr angezeigt und stattdessen das Standardspeicherformat verwendet. Das Standardspeicherformat kann jederzeit im Einstellungsdialog geändert oder entfernt werden. *Word Open XML Dateiformat und Makros:* In dem Word Open XML Dateiformat .docx können keine Makros gespeichert werden. Beim Versuch Dokumente mit Makros in diesem Format zu speichern, wird eine entsprechende Meldung angezeigt.

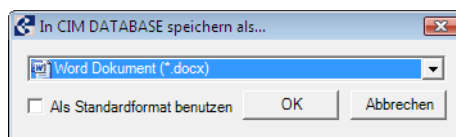


Abb. 7.2: Microsoft Word: Speicherformatauswahldialog

7.2.4 Index erzeugen im System

Die Schaltfläche [*Index erzeugen*] erzeugt für das aktive Dokument einen neuen Index im System. Dabei öffnet sich zunächst die Maske zur Indexerzeugung. Nach dem Ausfüllen und Bestätigen der Indexmaske werden die aktuellen Änderungen in dem neuen Dokumentenindex im System gesichert.

7.2.5 Dokumentenstatus ändern

Die Schaltfläche [*Status ändern*] führt für das aktive Dokument eine System-Statusänderungsoperation durch.

Bemerkung: Da der Anwender durch die Statusänderung ggf. die Rechte im System verliert, das Dokument erneut zu speichern, speichert die Kopplung das Dokument automatisch, bevor die Statusänderungsoperation aufgerufen wird. Dabei wird ggf. eine entsprechende Meldung angezeigt, die bestätigt werden muss.

7.2.6 Stammdatenänderungsdialog im System öffnen

Die Schaltfläche [*Stammdaten*] öffnet im System den Dialog zum Ändern der Stammdaten für das aktive Dokument. Diese Schaltfläche ist geteilt. Über den unteren Teil kann im System statt des Änderungsdialoges ein Informationsdialog für das aktive Dokument geöffnet werden.

7.3 Daten abgleichen

7.3.1 Metadatenübernahme aus dem System

Durch Betätigung der Schaltfläche [*Übernehmen*] werden die Werte aller System-Variablen mit lesendem Zugriff mit den entsprechenden Werten der zugehörigen System-Attribute aktualisiert. Vom Anwender im Dokument geänderte System- Variablen mit lesendem Zugriff werden dabei überschrieben. Falls ein Fehler im Ausdruck einer Variablen vorliegt, das referenzierte Objekt nicht existiert oder die Beziehung nicht eindeutig ist, wird der Wert auf ein Leerzeichen gesetzt.

7.3.2 Metadatensynchronisation mit dem System

Durch Betätigung der Schaltfläche [*Übergeben*] werden die Werte aller Variablen mit schreibenden Zugriff auf Änderungen überprüft. Wurden Werte im Dokument verändert, werden die betroffenen Objekte nacheinander im System geöffnet. Die Änderungen werden in den Änderungsdialogen vorgeblendet und müssen für jedes Objekt bestätigt werden. Nach Abschluss der Datenübernahme in das System wird eine Metadatenübernahme für System-Variablen mit lesendem Zugriff durchgeführt.

7.3.3 Vorlagen zum Datenabgleich erstellen

Dieser Abschnitt beschreibt die Erstellung von Dokumentvorlagen mit System- Variablen zum Datenabgleich.

- Abgleich von Dokumentmetadaten (z.B. Titel, Dokument-Nr. ...)
- Abgleich von 1:1 referenzierten Objekten (z.B. Projektname des zugeordneten Projekts)
- Übernahme von 1:N referenzierten Objekten (z.B. Liste mit Offenen Punkten)

Formatbeschreibung für System-Variablen

System-Variablen in Word Dokumenten haben folgendes Format: cdb.<Mode>.<Beziehung>.<Attribut>.<Kardinalität>

Mode Der Mode gibt an, ob eine Variable aus dem System ausschließlich gelesen oder auch nach Änderung im Dokument im System aktualisiert werden darf. Gültige Werte für den Modus sind:

- *r*

Nur Datenübernahme aus dem System möglich. Der übernommene Wert kann im Dokument verändert werden, jedoch nicht an das System übergeben werden. Bei erneuter Datenübernahme wird der geänderte Wert im Dokument wieder durch den System-Wert ersetzt.

- *w*

Nur Datenübergabe in das System möglich. Durch diesen Modus kann sichergestellt werden, dass ein Dokumentfeld durch eine Datenübernahme nicht verändert werden kann, eine Aktualisierung im System aber möglich ist (sinnvoll bei Feldern, die durch das Dokument berechnet werden). Dieser Modus kann nicht bei 1:N Beziehungen verwendet werden.

- *rw*

Datenübernahme aus dem System und Aktualisierung im System nach Änderung im Dokument möglich. Dieser Modus kann nicht bei 1:N Beziehungen verwendet werden.

- **s*

Wenn oben genannten Modewerten ein *s* hinzugefügt wird (d.h. *rs*, *ws* oder *rws*), dann gelten zwar die gleichen Regeln wie bei den jeweiligen beschriebenen Werten ohne *s*, jedoch laufen die Lese- bzw. Schreibaktionen vollständig serverseitig ab. Dabei kann optional das Feld *Parameter* verwendet werden. U.U. bedarf es allerdings für die Verwendung der serverseitigen Bearbeitung von Variablen vorbereitend administratives Customizing, welches im OfficeLink Administrationshandbuch erläutert wird.

Beziehung Die Beziehung gibt ausgehend vom Dokument den Pfad zu den durch die Variable repräsentierten Daten im System an. Als Beziehung können folgende Ausdrücke verwendet werden:

- *this*

Der konstante Ausdruck 'this' wird verwendet, um Attribute des Dokuments selbst anzugeben.

- *Name einer im System konfigurierten Beziehung*

Über eine Beziehung können Attribute von referenzierten Objekten übernommen und abgeglichen werden.

- *BY_ZNUM_ZIDX_FROM_<table_name>*

Verwendung einer impliziten Beziehung über die Attribute z_nummer und z_index. Hierzu wird der Ausdruck BY_ZNUM_ZIDX_FROM_<table_name> angegeben. table_name gibt dabei die Datenbanktabelle an, aus der ein Objekt über z_nummer und z_index implizit referenziert wird. Die angegebene Datenbanktabelle muss die Attribute z_nummer und z_index enthalten und die Kombination aus z_nummer und z_index muss eindeutig sein.

Attribut Das Attribut spezifiziert ein Attribut eines durch die Beziehung referenzierten System-Objekts.

Sachmerkmale als Attribute von Artikeln:

Wird eine Artikelbeziehung als Beziehung angegeben, können auch die Sachmerkmale des Artikels als Attribut angegeben werden. Dies kann qualifiziert über die Merkmalkennung oder unqualifiziert mittels Zählnummer erfolgen:

- *Sachmerkmale qualifiziert:*

`__SML_<mm_mk>`

Word: Datenabgleich - Sachmerkmale qualifiziert

Sachmerkmal mit der Kennung 'L' des zugeordneten Artikels

```
``cdb.r.cdb_doc_to_part.__SML_L``
```

- *Sachmerkmale unqualifiziert:*

`__SMLN<nummer>` für Merkmalskennungen

`__SMLV<nummer>` für Merkmalswerte

Merkmalkennung und der zugehörige Wert müssen dabei dieselbe Nummer haben.

Word: Datenabgleich - Sachmerkmale unqualifiziert

Kennung und Wert des erstes Sachmerkmals des zugeordneten Artikels

```
``cdb.r.cdb_doc_to_part.__SMLN1``  
``cdb.r.cdb_doc_to_part.__SMLV1``
```

Wichtig: SML Attribute können nicht schreibend synchronisiert werden!

Kardinalität Gibt die Kardinalität der Beziehung an. Wenn keine Kardinalität angegeben wurde, wird von der Kardinalität 1 ausgegangen. Bei der Erstellung neuer Dokumentvorlagen sollte die Kardinalität explizit angegeben werden. Folgende Werte können für die Kardinalität verwendet werden:

- *1*

Für Beziehungen vom Typ *this*, *BY_ZNUM_ZIDX_FROM_<table_name>* und für 1-stellige homogene Beziehungen.

- *N*

Für n-stellige homogene Beziehungen. Variablen, die n-stellige Beziehungen verwenden, müssen spaltenweise in der Kopfzeile einer Liste platziert werden (siehe Verwendung von System-Variablen im Dokument). Beim Abgleich mit dem System wird die Liste auf die zur Darstellung des Ergebnisses benötigte Anzahl Zeilen angepasst. Die erste Zeile nach der Kopfzeile dient dabei als Formatvorlage. Die Sortierung der Liste erfolgt anhand der ersten Spalte.

Word: Datenabgleich - System-Variablen

Zeichnungsnummer des Dokuments:

```
"cdb.r.this.z_nummer"
```

Übernahme und Abgleich der Bemerkung des Dokuments:

```
"cdb.rw.this.z_bemerkung"
```

Übernahme der Artikelnummer des zugeordneten Artikels:

```
"cdb.r.cdb_doc_to_part.teilenummer"
```

Übernahme und Abgleich der Artikelbenennung des zugeordneten Artikels:

```
"cdb.rw.cdb_doc_to_part.benennung"
```

Übernahme der Begründung aus der Änderungshistorie mittels impliziter Beziehung:

```
"cdb.r.BY_ZNUM_ZIDX_FROM_aenderung.begrueend"
```

Parameter Über dieses Feld können bei der serverseitigen Bearbeitung von Variablen zusätzliche für die jeweilige Variable spezifische konstante Werte übergeben werden. Das können beispielsweise Identifikatoren sein, welche das Verlinken von Ergebnissen aus 1:N Beziehungen mit einzelnen Dokumentfeldern statt mit dynamischen Listen ermöglichen. Beispielsweise wird der folgenden Variable der Parameter *PRIMARY* hinzugefügt, welches einem zuvor speziell dafür bereitgestelltem Userexit übermittelt, dass man den Ursprungsnamen der Primärdatei des Dokuments haben möchte.

```
cdb.rs.document2cdb_file.cdbf_original_name.N.string.PRIMARY
```

Definition von System-Variablen

Die Definition von System-Variablen erfolgt mit einem Assistenten, der über die Schaltfläche [*Verwalten*] der OfficeLink-Symbolleiste aufgerufen wird. Damit die Schaltfläche verfügbar ist, muss sie zuvor in dem Einstellungsdialog aktiviert worden sein. Der Dialog bietet auf der linken Seite die Möglichkeit neue Variablen im System-Format anzulegen (siehe *Formatbeschreibung für System-Variablen* (Seite 70)). Auf der rechten Seite werden alle im Dokument enthaltenen System-Variablen angezeigt. Durch Markierung einer Variablen wird gleichzeitig das entsprechende Textfeld in dem Word Dokument markiert. Durch Betätigung der Schaltfläche [*Variable löschen*] kann die markierte Variable aus dem Dokument entfernt werden.

- *Beziehung:*

Die Beziehung gibt ausgehend vom Dokument den Pfad zu einem Attribut eines oder mehrerer referenzierter System-Objekte an. Der Katalog bietet alle im System konfigurierten Dokumentenbeziehungen an, bei denen das Dokument das Startobjekt darstellt. Für Attribute des Dokuments selbst, muss in diesem Feld *this* eingetragen werden.

Zur Verwendung einer impliziten Beziehung, die über die Schlüsselattribute Dokumentennummer und Index hergestellt wird, muss in diesem Feld *BY_ZNUM_ZIDX_FROM_* ausgewählt werden und im Feld *Datenbanktabelle* eine Relation als Datenquelle ausgewählt werden.

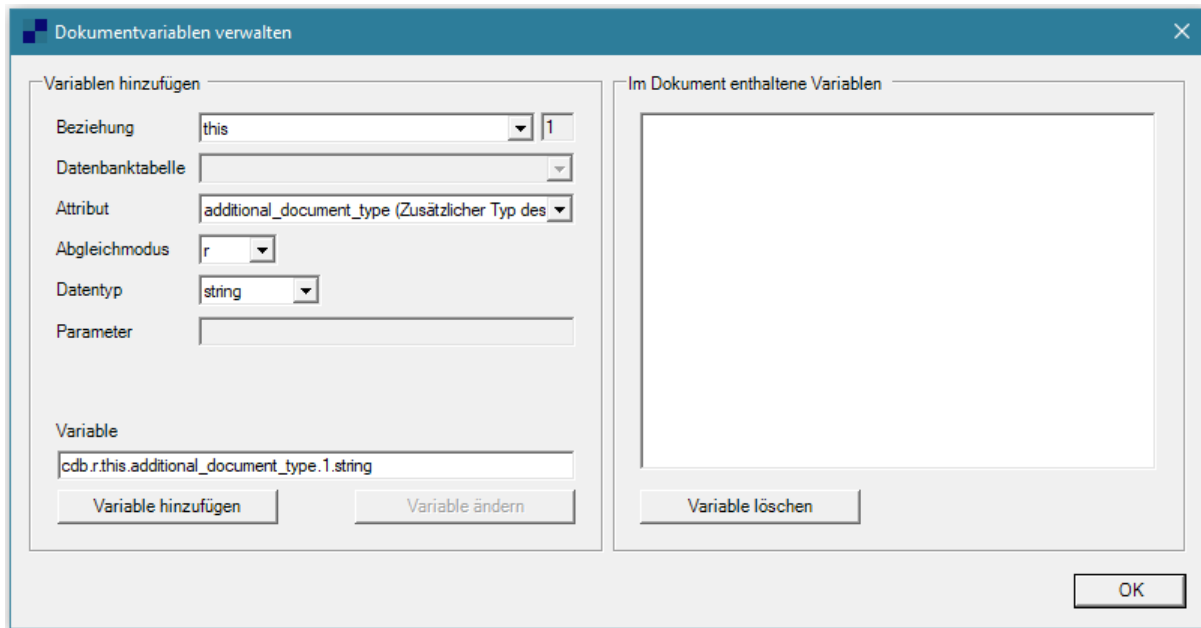


Abb. 7.3: Word: Datenabgleich - Dialog zur Erstellung von System-Variablen

- *Datenbanktabelle:*

Eine Datenbanktabelle muss nur dann ausgewählt werden, wenn im Feld Beziehung der spezielle Eintrag `BY_ZNUM_ZIDX_FROM_` für implizite Beziehungen ausgewählt wurde. In der ausgewählten Relation muss die Kombination der Attribute `z_nummer` und `z_index` eindeutig sein.

- *Attribut:*

Das Attribut spezifiziert ein Attribut eines durch die Beziehung referenzierten System-Objekts.

- *Abgleichmodus:*

Der Modus gibt an, ob eine Variable aus dem System ausschließlich gelesen oder auch nach Änderung im Dokument im System aktualisiert werden darf:

r: nur lesend

w: nur schreibend

rw: lesender und schreibender Zugriff im System

rs: nur serverseitig lesend

ws: nur serverseitig schreibend

rws: serverseitig lesender und schreibender Zugriff im System

- *Parameter:*

Optional nur für das serverseitige Bearbeiten von Variablen verwendeter Wert.

- *System Variable:*

In diesem Feld wird der aus den anderen Feldern erzeugte Ausdruck für die zu erstellenden System-Variable angezeigt. Das manuelle Ändern dieses Ausdrucks ist nur zur Angabe von Sachmerkmalen bei Artikelbeziehungen erforderlich.

Beim Hinzufügen der Variablen durch Betätigung der Schaltfläche [*Variable hinzufügen*] wird der in dem Feld System-Variable angezeigte Ausdruck überprüft und bei Ungültigkeit eine entsprechende Fehlermeldung angezeigt. Ansonsten wird die Variable als neue System-Variable in dem aktiven Dokument angelegt und mit dem Wert `<Attribut>` als Platzhalter initialisiert. Die System-Variablen werden durch Felder vom Typ DocVariable repräsentiert und sind nach dem Einfügen frei im Dokument positionierbar.

Ändern von System-Variablen

Die Änderung der Textfelder, welche System-Variablen beinhalten, erfolgt durch Mausklick in das zu ändernde Feld. Der Cursor befindet sich dann innerhalb des Feldes und ermöglicht eine freie Bearbeitung. Dieser Mechanismus funktioniert nur dann, wenn das Feld aus mindestens zwei Zeichen besteht. Besteht das Feld aus nur einem Zeichen, ist ein Mausklick in das Feld und eine direkte Bearbeitung nicht mehr möglich. Zu diesem Zweck enthält die OfficeLink-Symbolleiste in Word das zusätzliche Buttonuntermenü [*Dokumentvariablenwert ändern*], welches einen speziellen Dialog zum Ändern des Wertes öffnet.

Verwendung von System-Variablen im Dokument

System-Variablen sollten im Dokument für den Anwender kenntlich gemacht werden, um versehentliches Löschen oder Überschreiben zu vermeiden. Der Schutz von Dokumentenvariablen ist nur bedingt möglich. Eine Möglichkeit ist die Erstellung von Formularen.

Bemerkung: Beim Datenabgleich von Variablen in das System werden Attributänderungen auch dann vorgeblendet, wenn das Feld laut Maskenkonfiguration durch den Anwender nicht interaktiv änderbar oder nur via Katalogauswahl änderbar wäre. Bei der Verwendung von System-Variablen mit schreibenden Zugriff ist dies zu berücksichtigen. Gegebenenfalls ist die Gültigkeit der Werte durch eine entsprechende Prüfung im User Exit abzusichern.

System-Variablen, die n-stellige Beziehungen verwenden, müssen unter Word spaltenweise in der Kopfzeile einer Tabelle platziert werden. Beim Abgleich mit dem System wird die Tabelle auf die zur Darstellung des Ergebnisses benötigte Anzahl Zeilen angepasst. Die erste Zeile nach der Kopfzeile dient dabei als Formatvorlage.

Wichtig: Falls n-stellige Beziehungen in der Kopf- oder Fußzeile eines Dokuments eingebaut werden, und dann anschließend die Kopfzeilen in sprechende Texte geändert werden (z.B. von "<z_nummer>" in "Document number"), müssen diese Kopfzeilen unbedingt per CTRL+F11 gesperrt werden. Falls man diese Felder nicht sperrt, werden die sprechenden Text aufgrund einer Eigenart in Word bei einer späteren PDF Konvertierung in ihre ursprünglichen Platzhalternamen geändert (z.B. zurück von "Document number" auf "<z_nummer>"). Bei Bedarf kann die Sperre per CTRL+SHIFT+F11 aufgehoben werden.

Tabellen in Microsoft Office Word für n-stellige Beziehungen:

1. Tabelle mit zwei Zeilen und der gewünschten Anzahl Spalten einfügen
2. Spaltentitel in die erste Zeile eintragen und wie gewünscht formatieren
3. Als Spaltentitel die System-Variablen mit dem Assistenten einfügen (siehe [Definition von System-Variablen](#) (Seite 72)). Alle Variablen müssen dabei dieselbe Beziehung verwenden und von der Kardinalität N sein.

Offene Punkte

<issue_name>	<category>	<priority>

Abb. 7.4: Word: Datenabgleich - Tabelle für n-stellige Beziehungen [leer] (1)

4. System Variablen umbenennen, damit deren Attribute nicht mehr sichtbar sind.

Wichtig: Beim Ändern der Variable darf diese vorher nicht komplett selektiert werden, da dadurch die Variable gelöscht wird. Siehe dazu Bemerkung im Abschnitt "Ändern von System-Variablen".

5. Zweite Zeile formatieren. Diese Zeile dient als Kopiervorlage zum Einfügen weiterer Zeilen beim Datenabgleich.

Offene Punkte

Bezeichnung	Kategorie	Priorität

Abb. 7.5: Word: Datenabgleich - Tabelle für n-stellige Beziehungen [leer] (2)

6. Über den Tabelleneigenschaftendialog einmaligen Titel für die Tabelle definieren. Dies ist nötig, falls in einem Dokument die gleichen Variablen in mehrere Tabellen gleichzeitig ausgeleitet werden sollen, oder falls die Tabelle über eine Seite hinaus in die nächste Seite wachsen könnte.

Bei Durchführung einer Datenübernahme wird die Tabelle auf die benötigte Anzahl Zeilen expandiert und mit der Ergebnismenge der verwendeten Beziehung gefüllt.

Offene Punkte

Bezeichnung	Kategorie	Priorität
Dokumentation aktualisieren	zu vervollständigen	normal
Office 2007 Support	Korrektur	hoch
Service Level erhöhen	offen	niedrig

Abb. 7.6: Word: Datenabgleich - Tabelle für n-stellige Beziehungen [gefüllt]

Felder-Formatierung

Insbesondere bei Datumswerten ist es zuweilen wünschenswert, dass die Werte der abgeglichenen Felder in einem bestimmten Format dargestellt werden. Dies erreichen Sie durch folgende Schritte:

1. Drücken Sie ALT+F9 um die Feldfunktionen statt deren Ergebnisse anzuzeigen
2. Ersetzen Sie den String * *MERGEFORMAT* mit der gewünschten Formatierungsformel (z.B. @ "dd.MM.yyyy")
3. Drücken Sie erneut ALT+F9 um wieder Ergebnisse statt Feldfunktionen anzuzeigen

Automatisches Aktualisieren von Metadaten

Die Metadaten aller System-Variablen mit lesendem Zugriff werden beim Laden eines Dokuments aus CDB automatisch übernommen. Die Bedingungen dafür sind, dass das Dokument:

1. im Client im "Bearbeiten" Modus geöffnet wurde (oder direkt aus Office heraus über "Laden")
2. eine "Zeichnung" ist (d.h. die Eigenschaft "vorlage_kz" hat)
3. keine Vorlage ist (d.h. "vorlage_kz=0")

2.1	Excel: OfficeLink-Menüleiste	2
2.2	Microsoft Excel: Speicherformatauswahldialog	3
2.3	Excel: Datenabgleich - Dialog zur Erstellung von System-Variablen	7
2.4	Excel: Datenabgleich - Zelle auswählen	8
2.5	Excel: Datenabgleich - List/Tabelle für n-stellige Beziehungen [leer]	9
2.6	Excel: Datenabgleich - List/Tabelle für n-stellige Beziehungen [gefüllt]	9
2.7	Excel: PowerReports - Schema	10
2.8	Excel: PowerReports - Architekturübersicht	10
2.9	Excel: PowerReports - Funktionsprinzip Schemaausleitung	11
2.10	Excel: PowerReports - Funktionsprinzip XML Export	11
2.11	Excel: PowerReports - Datenquelleneigenschaften	12
2.12	Excel: PowerReports - Kontextprovider eines projektbezogenen Reports	13
2.13	Excel: PowerReports - Relationship Subprovider zum Kontextprovider	14
2.14	Excel: PowerReports - Datenprovider Eigenschaften	14
2.15	Excel: PowerReports - Beziehungen als Provider	15
2.16	Excel: PowerReports - Provider zur regelbasierten Ermittlung von Aufgaben	15
2.17	Excel: PowerReports - Konfigurierte Regel zur Ermittlung der Offenen Punkte eines Anwenders	16
2.18	Excel: PowerReports - SimpleQuery Provider	17
2.19	Excel: PowerReports - Ergebnistabelle eines GroupBy Providers	17
2.20	Excel: PowerReports - Gruppierung von Projekten	17
2.21	Excel: PowerReports - Gruppierung nach Wertebereichen mit Regeln	18
2.22	Excel: PowerReports - Gruppierungsregeln am Beispiel Projekte nach Budget	18
2.23	Excel: PowerReports - Datenprovider Standardparameter	19
2.24	Excel: PowerReports - Hyperlinkkonfiguration und zugehöriges Schemaobjekt	20
2.25	Excel: PowerReports - Langtextkonfiguration und zugehöriges Schemaobjekt	20
2.26	Excel: PowerReports - Generelle Zugriffsberechtigungen auf Providerebene	21
2.27	Excel: PowerReports - Auswahlkatalog für eigene Providerimplementierungen	21
2.28	Excel: PowerReports - Defaultparameters	27
2.29	Excel: PowerReports - Reportdialog am Beispiel Abfrage der Projektkategorie	28
2.30	Excel: PowerReports - XML Schema für Argumente	29
2.31	Excel: PowerReports - Report mit einschränkender Suchbedingung (1)	29
2.32	Excel: PowerReports - Report mit einschränkender Suchbedingung (2)	29
2.33	Excel: PowerReports - Hierarchische Darstellung am Beispiel Produktstruktur	32
2.34	Excel: PowerReports - cdb.objects basierte Komponente für Projekte	33
2.35	Excel: PowerReports - Aktivierung der cdbxml_excel_report Operation	33
2.36	Excel: PowerReports - Operationskonfiguration am Beispiel Dokumente	34
2.37	Excel: PowerReports - Kontextmenüoperation zum Aufruf von Reports	34
2.38	Excel: PowerReports - Auswahldialog für Datenquellen	34
2.39	Excel: PowerReports - Datenbindung via Drag&Drop herstellen	35
2.40	Excel: PowerReports - Report zur Datenquelle	35

2.41	Excel: PowerReports - Reportdialog zur Abfrage von Projekten	36
2.42	Excel: PowerReports - Adressierung der Dialogfelder an Datenprovider als Parameter	36
2.43	Excel: PowerReports - Speichern der Eingabewerte von Reportdialogen	37
2.44	Excel: PowerReports - Hierarchische Darstellung am Beispiel Produktstruktur	37
2.45	Excel: PowerReports - Hierarchische Darstellung mit Einrückung der 1. Spalte und Farbcodierung	38
2.46	Excel: PowerReports - Hierarchische Darstellung - Gruppierungseinstellungen	38
2.47	Excel: PowerReports - Darstellung dynamischer Spaltentitel	39
2.48	Excel: PowerReports - Darstellung dynamischer Spaltentitel (Layout)	39
2.49	Excel: PowerReports - Dynamische Bildinhalte (Musterbild)	40
2.50	Excel: PowerReports - Dynamische Bildinhalte (Beispiel 1)	40
2.51	Excel: PowerReports - Dynamische Bildinhalte (Beispiel 2)	41
2.52	Excel: PowerReports - Komplexe Zeilen am Beispiel Artikelliste	41
2.53	Excel: PowerReports - Definition einer komplexen Zeile	42
2.54	Excel: PowerReports - Schema Multi-Export	42
2.55	Excel: PowerReports - Persönlicher Report am Beispiel Mitarbeiterauslastung	44
2.56	Excel: PowerReports - Report zur Datenquelle	44
2.57	Excel: PowerReports - Report zur Datenquelle	44
2.58	Excel: PowerReports - Reportauswahldialog	45
2.59	Excel: PowerReports - Reportauswahl als eLink-Panel	46
2.60	Excel: PowerReports - Reportauswahlkonfiguration	47
3.1	Outlook: OfficeLink-Symboleiste	48
3.2	Outlook: Nachrichten mit Anlagen im Archivierungsauswahldialog	50
3.3	Outlook: Steuerelement 'Anhängen'	51
3.4	Outlook: 'Als Mail exportieren' in CIM Database/WIN	51
4.1	PowerPoint: OfficeLink-Menüleiste	52
4.2	Microsoft PowerPoint: Speicherformatauswahldialog	53
4.3	PowerPoint: Datenabgleich - Dialog zur Erstellung von System-Variablen	56
5.1	Project: OfficeLink-Menüleiste	59
6.1	Visio: OfficeLink-Menüleiste	62
6.2	Visio: Speicherformatauswahldialog	63
6.3	Visio: Datenabgleich - Dialog zur Erstellung von System-Variablen	66
7.1	Word: OfficeLink-Menüleiste	68
7.2	Microsoft Word: Speicherformatauswahldialog	69
7.3	Word: Datenabgleich - Dialog zur Erstellung von System-Variablen	73
7.4	Word: Datenabgleich - Tabelle für n-stellige Beziehungen [leer] (1)	74
7.5	Word: Datenabgleich - Tabelle für n-stellige Beziehungen [leer] (2)	75
7.6	Word: Datenabgleich - Tabelle für n-stellige Beziehungen [gefüllt]	75

1.1	Liste der OfficeLink Anwendungsmöglichkeiten	1
5.1	Aktualisierte Aufgabenattribute	61