



Was wir vorhersagen, soll auch eintreffen!



## Berichtserstellung mit SAP Analysis for Office //

*Marcel Seelig*



## Zusammenfassung

Analysis for Office ist laut aktueller Roadmap der SAP ein wichtiges BI Tools für Fachbereichsanwender. Diese sind bereits in der MS Office Welt zu Hause und haben mit Analysis for Office eine einfache Möglichkeit, in diese bekannte Umgebung ihre SAP Daten zu integrieren und mit lokalen Daten anzureichern. Des Weiteren können Fachbereiche nun anspruchsvollere Reporting- und Planungsanwendungen eigenständig erstellen. Hierfür stellt die SAP mit Analysis for Office entsprechende Werkzeuge zur Verfügung und bietet dem Endanwender neue Optionen.

Einige dieser Optionen werden nachfolgend beschrieben.

## Überblick AfO

Analysis for Office ist seit November 2017 in der Version 2.5 SP2 erhältlich, welche als Basis dieses Whitepapers dient. Genaue Termine und Veröffentlichungsdaten der verschiedenen Versionen können der SAP Note 15183591 entnommen werden.

Analysis for Office ist als Plugin zur Berichterstellung und Planung für Microsoft Office entwickelt worden. Folglich gibt es Module für MS Excel, MS PowerPoint und MS Word. Hierbei ist anzumerken, dass das Plug-In für Excel bei Weitem das größte Spektrum an Funktionalität bietet. Daher beziehen sich die folgenden Sachverhalte auf besagtes Excel Plug-In von Analysis for Office.

## Anbindung von Datenquellen

Das zentrale Element ist die Data Source. Analysis for Office bietet die Möglichkeit, Daten aus verschiedenen Daten-Quellen anzubinden. Hierbei kann ein Excel Workbook mehrere Datenquellen enthalten, welche zur Datenversorgung mehrerer Objekte dienen. Gesteuert wird die Datenversorgung durch Filter und Variablen.

Mögliche Quellen der Data Source sind im Einzelnen:

- SAP BW System
  - SAP BEx Queries
  - Query Views
  - BW Infoprovider
- SAP HANA Data Sources
- SAP Analytics Cloud Models
- Lokale Daten

Unabhängig von der Art der Verbindung muss der Benutzer die erforderlichen Rechte zur Benutzung der Daten besitzen, um diese herstellen zu können.

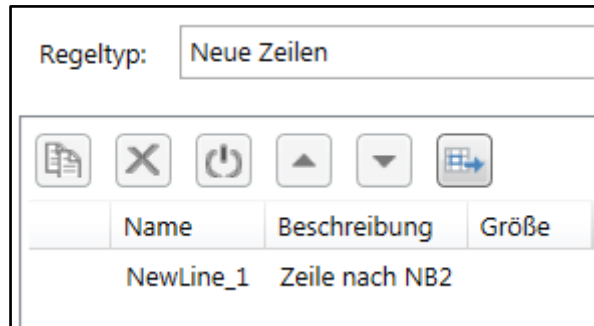
## Designregeln

Seit Version 2.0 gibt es die „Table Design“-Option. Manuelle Änderungen und Ergänzungen am Dokument können getrackt und nachvollzogen oder rückgängig gemacht werden. Zusätzliche Zeilen und Spalten und sogar eigene Formeln können hinzugefügt werden, wobei in früheren Versionen eigene Formeln beim

---

<sup>1</sup> <https://launchpad.support.sap.com/#/notes/1518359/E>

Auffrischen von Queries automatisch zurückgesetzt wurden. Seit der Version 2.5 hat der Benutzer die Wahl, die Designregeln zu behalten oder zu verwerfen.



**Abbildung 1:** Beispiel für Designregel

Die Reihenfolge der Regeln wird im Designregelfeld bestimmt. Durch die oberhalb angeordneten Buttons können einzelne Regeln sowohl verschoben, und somit neu priorisiert, als auch gelöscht oder vorübergehend deaktiviert und bei Bedarf wieder aktiviert werden.

Die Designregeln lassen sich auch mittels der Analysis for Office API nutzen. So können weitere Zeilen oder Spalten durch VBA-Anweisungen eingefügt werden. Die dazu notwendigen Funktionen sind im weiteren Verlauf dieses Whitepapers aufgeführt.

## Grouping und Linking Dimensionen

Mit Hilfe der Funktionen „Grouping“ und „Linking“ ist es möglich, mehrere Datenquellen, die ein gemeinsames Merkmal besitzen, zu verknüpfen. Dabei wird zwischen Verbinden (Combine) und Gruppieren (Group) unterschieden. So ist es beispielsweise mittels Verbinden möglich, in einer Tabelle eine bestimmte Merkmalsausprägung zu filtern. Die verknüpfte Tabelle wird daraufhin ebenfalls gefiltert, unabhängig davon, ob das zuzufilternde Merkmal auch in der zweiten Tabelle vorhanden ist. Änderungen im Filter wirken sich bei Verbindungen immer auf beide Tabellen aus.

Die Funktionalität „Gruppieren“ ermöglicht es, eine zweite Tabelle an die führende Tabelle anzuhängen. Somit werden die Daten aus zwei Datenquellen zu einer Tabelle kombiniert.

## Programmierung in Analysis for Office

Berichte in Analysis for Office können durch programmieretechnische Mittel weiter verbessert und individualisiert werden. AfO bietet die Möglichkeit, direkt in jeder Zelle neben den aus Excel bekannten Formeln auch mit speziellen Analyse-Formeln zu arbeiten. Außerdem können Daten, die über Data Sources in generierte Tabellen in die Berichte geladen wurden, auch in Analyse-Formeln umgewandelt werden. Diese Daten stehen somit direkt mit Excel Bordmitteln zur Weiterverarbeitung zur Verfügung. Nachfolgende Übersicht zeigt die aktuell verfügbaren AnalyseFormeln.

- **SAPGetData** → This function returns the measure value for a specific dimension member combination.
- **SAPGetDimensionDynamicFilter** → This function returns the dynamic filter of a dimension.
- **SAPGetDimensionEffectiveFilter** → This function returns the dynamic filter of a dimension.
- **SAPGetDimensionInfo** → This function returns the name of a dimension or the name of an active hierarchy.
- **SAPGetDimensionStaticFilter** → This function returns the static filter of a dimension
- **SAPGetDisplayedMeasures** → This function returns a list of all measures displayed in the analysis as text.
- **SAPGetInfoLabel** → The property names correspond to the info fields
- **SAPGetMeasureFilter** → This function returns a list of filtered measures created with Filter by Measure
- **SAPGetMember** → This function returns the dimension member or attribute.
- **SAPGetSourceInfo** → This function returns an info field value for a data source.
- **SAPGetUniformScaling** → This function returns the unit/scaling information for a specific dimension member combination.
- **SAPGetVariable** → This function returns the value of the specified property for a specific BW variable.
- **SAPGetWorkbookInfo** → This function returns an info field value for the current workbook.
- **SAPListOf** → This function returns a list of available objects for a specific object type.
- **SAPListOfDimensions** → This function returns a list of dimensions.
- **SAPListOfDynamicFilters** → This function returns a list of all dynamic filters of a data source.
- **SAPListOfEffectiveFilters** → This function returns a list of all effective filters of a data source.
- **SAPListOfMeasureFilters** → This function returns a list of all measure filters including BEx conditions.
- **SAPListOfMembers** → This function returns a list of members for a selected dimension or prompt.
- **SAPListOfMessages** → This function includes messages into the sheet.
- **SAPListOfStaticFilters** → This function returns a list of all static filters of a data source.
- **SAPListOfVariables** → This function returns a list of all variables of a data source or planning object.
- **SAPSetData** → With this function you can set planning values.
- **SAPSetFilterComponent** → This function creates a filter component and sets the members selected by the user as a filter.

Eine weitere Möglichkeit durch Programmierung selbst erstellte Berichte anzupassen und auf die eigenen Bedürfnisse zuzuschneiden bieten VBA Makros. In der bekannten VBA Programmierumgebung von Microsoft Excel können die von Analysis zur Verfügung gestellten Funktionen verwendet werden.

Um die Methoden aufzurufen, wird i.d.R. das VBA Kommando `Application.Run` verwendet. Die entsprechende Methode wird daraufhin zusammen mit ihren Parametern übergeben. Als Rückgabewert folgt anschließend das Ergebnis der Ausführung, im einfachsten Fall die erfolgreiche oder fehlerhafte Durchführung. eispielsweise könnte ein Makroaufruf wie folgt aussehen:

```
Dim result as Long
Result = Application.Run(„SAPLogOff“, True )
```

**Abbildung 2:** An- und Abmelden

Mit dieser kurzen VBA Anweisung würde die Verbindung zum BW-System abgebaut und wieder neu aufgebaut werden. Legt man dieses Makro nun auf einen eigenen Button, so besteht ein schneller und einfacher Weg, die Verbindung zum Backend zu beenden oder aufzufrischen.

```
Dim lResult as long

lResult= Application.Run("SAPSetFilter", "DS_1",
"OSOLD_TO__OCOUNTRY", "CA;US;DE", "INPUT_STRING")
```

**Abbildung 3: Filter setzen**

Diese VBA-Anweisung setzt einen Filter. Die Daten der Data Source „DS\_1“ werden in der Dimension „OSOLD\_TO\_\_OCOUNTRY“ auf die 3 Länder „Kanada“, „USA“ und „Deutschland“ eingeschränkt.

Einen Überblick über alle Funktionen bietet die nachfolgende Grafik. Sie enthält neben dem Namen der Methoden auch eine kurze Beschreibung der Funktion.

- **SAPAddMessage** → you can define messages and add them to the standard message dialog.
- **SAPCallMemberSelector** → you can call the standard input help (member selector) from the worksheet
- **SAPExecuteCommand** → method as a generic command to execute the commands (PlanDataSave)
- **SAPExecutePlanningFunction** → you can execute a planning function.
- **SAPExecutePlanningSequence** → you can execute a planning sequence.
- **SAPGetCellInfo** → you can define a command to get information on a crosstab cell
- **SAPGetProperty** → You can use this API method to read properties of a data source or a workbook.
- **SAPLogOff** → You can use this API method to disconnect a workbook from the server
- **SAPLogon** → You can use this API method to trigger a logon to a system for a specified data source.
- **SAPMoveDimension** → With this API method, you can define the position of a dimension in the crosstab, swap the dimension with another one or position a dimension relative to another one.
- **SAPOpenWorkbook** → You can use this API method to open an Analysis workbook.
- **SAPSetFilter** → With this API method, you can define which members of a dimension should be filtered.
- **SAPSetPlanParameter** → With this API method, you can define values for input-ready variables of planning objects.
- **SAPSetRefreshBehaviour** → With this API method, you can use different API methods one after the other without the result set being refreshed and without the message dialog being shown.
- **SAPSetVariable** → With this API method, you can define values for input-ready BW variables (prompts).
- **SAPSuppressMessage** → With this API method, you can suppress messages in the standard message dialog.
- **Table Design API** → Table Design API enables you to edit single elements in a crosstab with API methods:
  - **SAPSetFormat**
  - **SAPInsertLine**
  - **SAPFixLineSize**
  - **SAPListOfDesignRules**
  - **SAPDeleteDesignRule**

## Fazit

Mit Analysis for Office hat SAP einen weiteren wichtigen Schritt in Richtung Office Integration gemacht. Fachbereichsanwender sind bereits mit MS Excel vertraut und haben folglich mit geringem Ausbildungsaufwand viele neue Möglichkeiten, die bestehende Reporting- und Planungswelt zu ergänzen.



## Kontaktieren Sie uns //

mayato GmbH  
Am Borsigturm 9  
13507 Berlin

[info@mayato.com](mailto:info@mayato.com)

+49 / 30 4174 4270 10