

Kapitel 7

SAP Lumira Designer

Mit dem SAP Lumira Designer können Sie State-of-the-Art-BI-Applikationen aus einer Vielzahl von Datenquellen erstellen. Speziell konzipiert wurde dieses Werkzeug jedoch für die Verwendung von SAP BW und SAP HANA als Datenquellen. Die entwickelten Applikationen können Sie auf beliebigen Endgeräten verwenden.

7

SAP Lumira 2.x besteht neben SAP Lumira Discovery noch aus dem SAP Lumira Designer. In diesem Kapitel geben wir Ihnen einen vertieften Einblick in den SAP Lumira Designer und erläutern, wie Sie dieses Tool effizient nutzen können. Der SAP Lumira Designer ist der Nachfolger des SAP BusinessObjects Design Studios 1.6, dessen Applikationen Sie auch weiterhin in einem Kompatibilitätsmodus bearbeiten, speichern und ausführen können. Hier ergeben sich jedoch zwei Nachteile: Erstens können Sie keine neuen Funktionalitäten nutzen, die es in SAP Lumira 2.x gibt, und zweitens können Sie diese Applikationen auch nicht in SAP Lumira Discovery nutzen. Wir beginnen mit einem Überblick über die Architektur des SAP Lumira Designers und erläutern zunächst den Aufbau der Entwicklungsumgebung, in die das Werkzeug integriert ist. Nachfolgend stellen wir Ihnen die Komponenten vor, die zur Erstellung und Ausführung einer Applikation zur Verfügung stehen, bevor wir auf die Scripting-Sprache des SAP Lumira Designers eingehen. Danach zeigen wir Ihnen, wie Sie Cascading Style Sheets (CSS) anwenden können, um das Layout Ihrer Lumira-Designer-Applikationen zu verändern.

Der SAP Lumira Designer ist innerhalb der SAP-BI-Suite als Werkzeug positioniert, mit dem Sie Dashboards und BI-Applikationen erstellen können. Lumira-Studio-Applikationen werden mit einem Client erstellt, den Sie unabhängig von Ihrer Installation der SAP BusinessObjects Business Intelligence Platform (BI Platform) installieren müssen. SAP-Lumira-Studio-Applikationen im 2.x-Format (**.lumx**) basieren auf der `SAPUI5.m`-Bibliothek. Damit ist sichergestellt, dass die Applikationen sowohl auf dem Desktop als auch auf mobilen Geräten ohne Probleme dargestellt werden können. Vom SAP Lumira Designer unterstützte Datenquellen sind:

- **Online und offline**
 - SAP BW: InfoProvider, BW Queries und BW Query Views
 - SAP HANA: Calculation und Analytic Views
 - UNX-Universen



Unterstützung von Universen im SAP Lumira Designer

Beachten Sie, dass nicht alle Universentypen unterstützt werden. Eine aktuelle Liste der unterstützten Universen (UNX) können Sie in der Anwendungshilfe unter <http://s-prs.de/v563926> finden.

- Nur offline
 - Dateien
 - Drittanbieter



Mischen von Datenquellen

Das Mischen von Datenquellen funktioniert nur, wenn Sie die Daten offline nutzen. Die Daten aus den verschiedenen Quellen werden zuerst in die In-Memory Data Engine (Velocity) geladen und können dann in den Applikationen genutzt werden

7.1 Architektur

In diesem Abschnitt erhalten Sie zunächst einen Überblick über die Architektur des SAP Lumira Designers. Bei diesem Produkt handelt es sich um ein Eclipse-basiertes Werkzeug.



Integrierte Entwicklungsumgebung Eclipse

Der Funktionsumfang der offenen integrierten Entwicklungsumgebung (SAP Web IDE) *Eclipse* kann mithilfe von Plug-ins um zusätzliche Features erweitert werden. Der Designer des SAP Lumira Designers ist z. B. ein solches Plug-in. SAP nutzt diese Technologie bereits seit einiger Zeit für andere Entwicklungs- und Designwerkzeuge. Wenn ein Kunde bereits Erfahrungen mit einem anderen Eclipse-basierten Werkzeug gesammelt hat, wie z. B. SAP Composite Environment oder SAP HANA Studio, hat die Integration in die offene Entwicklungsumgebung Eclipse für ihn den Vorteil, dass die Einarbeitungszeit sehr viel geringer ist.

Das Eclipse-Plug-in SAP Lumira Designer stellt neben den Services zum Design der Applikationen auch eine Engine zur Generierung der Webapplikationen des SAP Lumira Designers bereit. Durch die in die Entwicklungsumgebung integrierte Engine können Applikationen im WYSIWYG-Modus (*What You See Is What You Get*) erstellt werden. Darüber hinaus können Applikationen im Rahmen der Entwicklung lokal ausgeführt werden.

Abbildung 7.1 zeigt die Arbeitsoberfläche des Eclipse-Plug-ins.

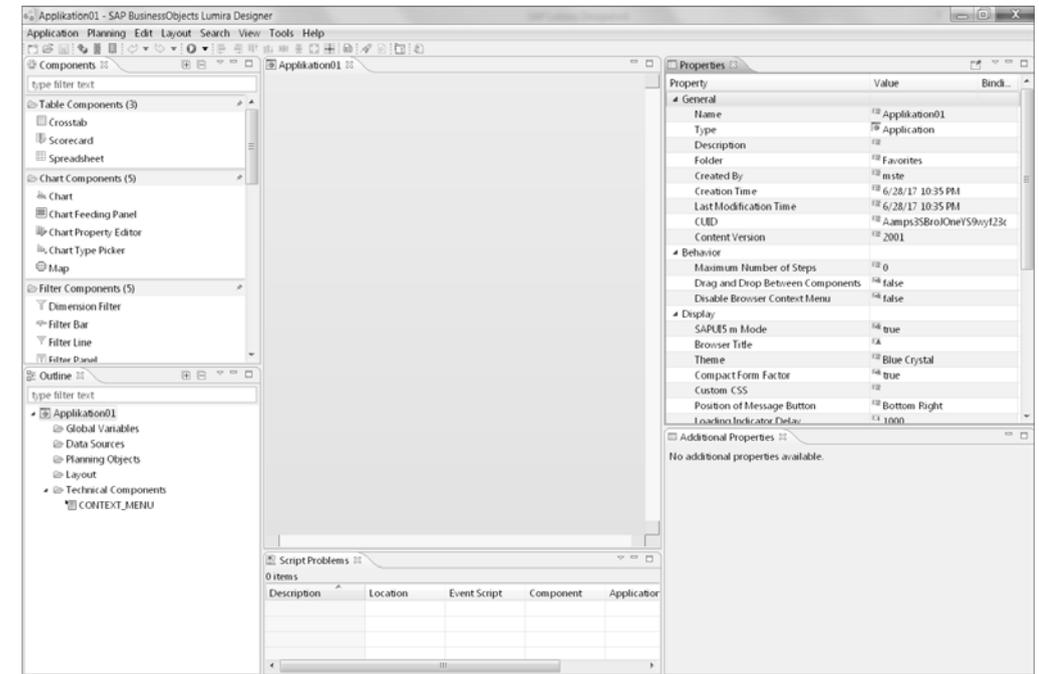


Abbildung 7.1 SAP Lumira Designer in Eclipse

Der SAP Lumira Designer ist ein BI-Werkzeug, das sich auf verschiedenen Plattformen integrieren lässt. Als Plattformen für den SAP Lumira Designer kommen infrage:

- keine Plattform (Local Mode)
- SAP BusinessObjects BI Plattform

Sie können für jede Applikation eine Plattform auswählen, indem Sie im SAP Lumira Designer in der Menüleiste auf den Eintrag **Tools** klicken und dann **Preferences** wählen. Es öffnet sich ein Fenster, in dem Sie im Bereich **General** in einer Auswahlliste die Plattform auswählen können (siehe Abbildung 7.2).

Unabhängig davon, welche Plattform Sie auswählen, stehen Ihnen im SAP Lumira Designer grundsätzlich die gleichen Funktionen zur Verfügung. In den folgenden Abschnitten gehen wir näher auf das Zusammenspiel zwischen dem SAP Lumira Designer und den verschiedenen Plattformen ein.

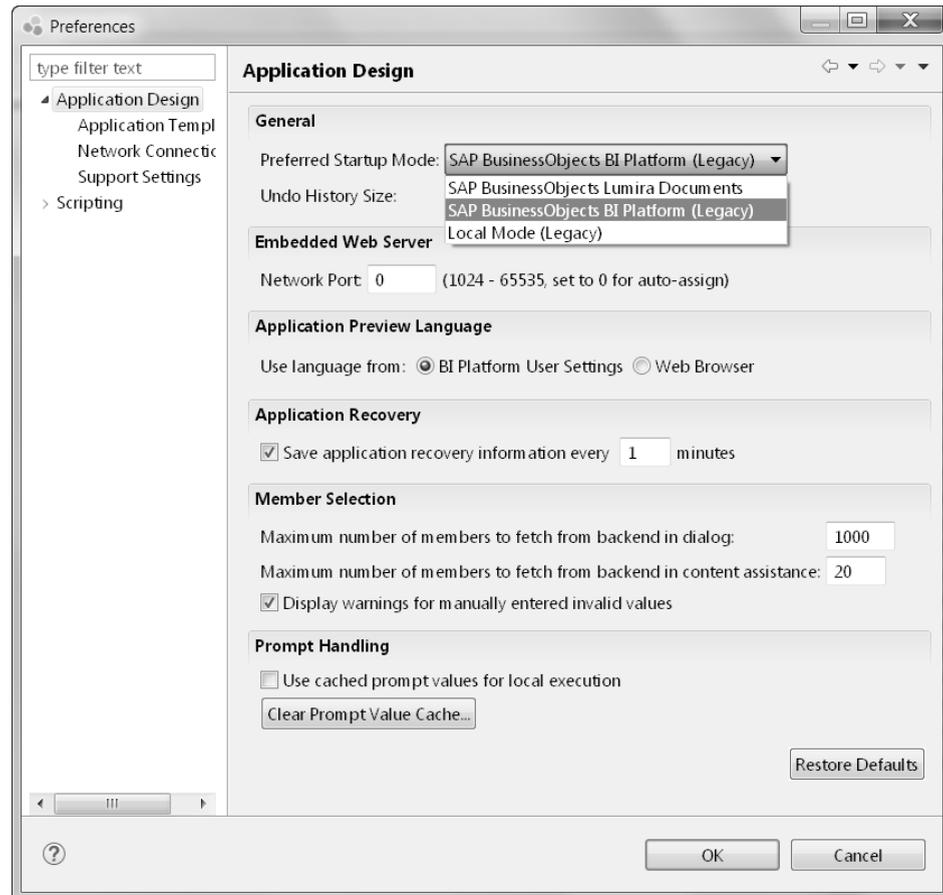


Abbildung 7.2 Auswahl der Plattform für den SAP Lumira Designer

7.1.1 Local Mode

Sie können den SAP Lumira Designer lediglich als Desktop-Version auf Ihrem Rechner verwenden. In diesem Fall werden die SAP-Lumira-Designer-Applikationen auf Ihrem Rechner gespeichert und auch dort angezeigt. Der Local Mode ist nur für sogenannte *Proof-of-Concepts* (PoC) geeignet, d. h., um zu zeigen, dass der SAP Lumira Designer für einen konkreten Anwendungsfall nutzbar ist, oder für Testzwecke, falls Ihnen keine Plattform zur Verfügung steht.

Die Anbindung von BW-Datenquellen erfolgt in diesem Fall über die Datei **saplogon.ini** im Windows-Verzeichnis Ihres PCs unter **%APPDATA%/SAP/Common**. Die Anbindung von SAP-HANA-Datenquellen erfolgt über eine ODBC-Verbindung (Open Database Connectivity) mit SAP HANA. Um eine solche ODBC-Verbindung zu erstellen, müssen Sie auf Ihrem Rechner den SAP HANA Client installieren, um den ent-

sprechenden Treiber einzurichten. Ein Zugriff auf Universen ist im lokalen Modus nicht möglich.

Auch wenn Ihre Lumira-Designer-Applikationen nur auf Ihrem Rechner gespeichert werden, können Sie diese teilen, indem Sie den eingebetteten Webserver verwenden, um den externen Zugriff zu ermöglichen. Um einen externen Zugriff auf Ihre Applikationen zu erlauben, gehen Sie wie folgt vor:

1. Klicken Sie auf die Registerkarte **Tools** und dann auf **Preferences** in der Lumira-Designer-Menüleiste.
2. Wählen Sie **Application Design**.
3. Aktivieren Sie die Checkbox **Allow external access to embedded Web server**.

Anschließend können Sie und andere Anwender über einen Quick-Response-Code (QR-Code) von einem mobilen Gerät aus auf die lokalen Applikationen zugreifen (siehe Abbildung 7.3).

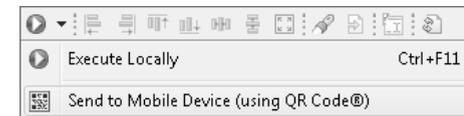


Abbildung 7.3 Export einer lokal gespeicherten Applikation mittels QR-Code

Der externe Zugriff auf Ihre lokal gespeicherten SAP-Lumira-Designer-Applikationen funktioniert, solange sich das Zugriffsgerät (z. B. iPad) im selben Netzwerk befindet und der SAP Lumira Designer auf Ihrem Rechner gestartet ist.

7.1.2 SAP BusinessObjects BI Platform nutzen

Wenn Sie Ihre Lumira-Designer-Applikationen auf der BI Platform speichern, müssen Sie mindestens die SAP BusinessObjects BI Platform 4.0 SPO5 Patch 9 verwenden. Das Transportieren der Applikationen erfolgt in diesem Fall über die Hochstufverwaltung (LCM) der Central Management Console (CMC). Sie können Verbindungen zu SAP BW, SAP HANA und Universen erstellen, und das sogar innerhalb einer einzigen Applikation.

Die Verbindung zu Ihrem BW-System wird in diesem Fall als OLAP-Verbindung auf der BI Platform angelegt. Bild- und CSS-Dateien werden im BI Launchpad hochgeladen und verwaltet. Um sicherzustellen, dass Ihre Bilddateien richtig angebunden werden können, laden Sie die Dateien direkt über die Eigenschaften des SAP Lumira Designers auf die BI Platform, nicht aber aus dem BI Launchpad heraus. Um eine Bilddatei in das BI Launchpad hochzuladen, gehen Sie zunächst zu den Eigenschaften einer Image-Komponente der Kategorie **Basic Components** (siehe Abbildung 7.4). Klicken Sie auf den Wert der Eigenschaft **Image**, erscheint ein Button mit drei Punkten

⋮. Klicken Sie auf diesen Button, um Ihren Rechner nach der Bilddatei zu durchsuchen, die Sie in das BI Launchpad hochladen möchten.

Property	Value	Bindi...
General		
Name	IMAGE_1	
Type	Image	
Vendor	SAP SE	
Visible	true	+
Display		
CSS Class		+
Image		+
Hover Image		+
Click Image		+
Opacity (%)	100	+
Tooltip		+
Display Mode	Stretch	
Background Color		
Events		
On Click		
Layout		
Top Margin	283	+
Left Margin	147	+
Bottom Margin	auto	+
Right Margin	auto	+
Width	100	+
Height	50	+

Abbildung 7.4 Bilddateien in das BI Launchpad hochladen

Als alternative Speicherung der Bilddateien im BI Launchpad können Sie die Bilder direkt auf dem Webserver (z. B. Apache Tomcat) Ihrer BI Plattform speichern. Dies hat den Vorteil, dass stets auf die Dateien zugegriffen werden kann, unabhängig davon, wo Sie Ihre SAP-Lumira-Designer-Applikation gespeichert haben. Beachten Sie, dass SAP-Lumira-Designer-Applikationen nur in SAP BusinessObjects Mobile auf mobilen Geräten verwendet werden können, wenn Sie Ihre Applikationen auf der BI Plattform speichern.

7.1.3 SAP Lumira Documents Mode

Dies ist der neue Standardmodus, der die Erstellung von SAP-Lumira-Dokumenten unterstützt, die beliebig viele Analysenanwendungen, Composites, MIME-Objekte und/oder Offline-Datensätze enthalten können, die mit SAP Lumira Discovery erstellt wurden.

Der SAP Lumira Documents Mode (**Lumira Documents Mode**) ist ein Modus, bei dem Sie immer lokal arbeiten und zusätzlich entscheiden, ob Sie mit der BI Plattform ver-

bunden sein wollen. Es ist immer möglich, Dokumente im Ordner **SAP-Lumira-Dokumente** zu bearbeiten oder zu erstellen, egal ob Sie mit einer BI Plattform verbunden sind oder nicht. Wenn Sie mit der BI Plattform verbunden sind, kann auch jedes auf der Plattform gespeicherte Dokument bearbeitet werden. Technisch, wenn Sie im SAP Lumira Documents Modus arbeiten, bearbeiten Sie immer ein Dokument, das lokal auf Ihrem PC/Laptop gespeichert ist. Dies bedeutet, dass Sie zuerst die Dokumente herunterladen müssen, die bearbeitet werden sollen, wenn Sie mit der BI Plattform verbunden sind.

Getrennt von der BI Plattform arbeiten

Wenn Sie mit der BI Plattform verbunden waren, ist es möglich, die von der BI Plattform heruntergeladenen Dokumente auch zu bearbeiten, wenn Sie von der BI Plattform getrennt sind. Dieser spezielle Bearbeitungsmodus ist jedoch sehr begrenzt, da Sie keine neuen Dokumente erstellen bzw. mit (Online-)Datenquellen arbeiten können.

Wenn Sie im SAP Lumira Documents Mode arbeiten, mit der BI Plattform verbunden sind und Änderungen an einer Anwendung oder einem Composite speichern, wird das Dokument nur lokal abgespeichert. Sie können zu jedem Zeitpunkt entscheiden, ob Sie Ihre lokalen Änderungen auf die Plattform hochladen möchten.

BI-Plattform-Version und lokale Version

Wenn mehrere Benutzer das gleiche Dokument entweder in SAP Lumira Discovery oder im SAP Lumira Designer nach dem Herunterladen von der BI Plattform ändern, kann jeder Benutzer seine Änderungen speichern und das Dokument wieder auf der BI Plattform veröffentlichen. Im SAP Lumira Designer wird Ihnen mitgeteilt, dass das Dokument auf der BI Plattform geändert wurde, wenn Sie versuchen das geänderte Dokument auf die BI Plattform hochzuladen. Wenn Sie beschließen, die lokale Version mit der BI-Plattform-Version zu überschreiben, aber nicht die lokalen Änderungen verlieren möchten, können Sie auch **Abbrechen** oder **Speichern unter** wählen, um das lokal modifizierte Dokument auf der BI Plattform unter einem anderen Namen zu speichern. Danach können Sie die Synchronisation starten, um Ihre lokale Version mit der BI-Plattform-Version zu überschreiben.

7.1.4 Vergleich der SAP-Lumira-Designer-Plattformen

Nachdem wir nun alle verfügbaren Plattformen für den SAP Lumira Designer erläutert haben, stellt Tabelle 7.1 die verschiedenen Optionen noch einmal einander gegenüber.

Plattform	Zentraler Zugriff	Bild- und CSS-Verwaltung	Fähigkeit zum mobilen Zugriff
Local Mode	nur temporär über einen eingebetteten Webserver	lokaler Rechner	temporär über einen eingebetteten Webserver und den Internetbrowser des mobilen Geräts
BI Platform	BI Launchpad	Daten werden im BI Launchpad gespeichert. Das Hochladen erfolgt im SAP Lumira Designer.	SAP BusinessObjects Mobile

Tabelle 7.1 Vergleich der SAP-Lumira-Designer-Plattformen

7.2 Aufbau der Entwicklungsumgebung

Die Entwicklungsumgebung des SAP Lumira Designers starten Sie über den folgenden Pfad im Windows-Startmenü:

SAP Business Intelligence • SAP BusinessObjects Lumira • Lumira Designer

7.2.1 Anmeldevorgang

Abhängig davon, ob Sie die erstellten Dokumente auf der BI Platform oder lokal abspeichern möchten, erscheint während des Starts ein abweichendes Anmeldebild. Sie können diese Einstellung im Menü unter **Tools • Preferences • Application Mode • General • Preferred Startup Mode** ändern. Abbildung 7.5 zeigt exemplarisch die Anmeldung an einer BI Platform.

Geben Sie Benutzernamen und Passwort für die BI Platform ein. Die Webservice-URL folgt dem folgenden Aufbau: `http://<BO-Server>:<Port>/dwsbobje/services/Session`. Im Feld **Authentication** können Sie die Quelle für die Berechtigungsprüfung auswählen. Im Standard ist dies die **Enterprise**-Authentifizierung. Eventuell hat Ihr Systemadministrator aber auch andere Typen für die Berechtigungsprüfung zur Verfügung gestellt, z. B. **Windows Active Directory** und **SAP**.

Nach einem Klick auf **OK** erfolgt die Anmeldung an der BI Platform. Möchten Sie diesen Schritt überspringen und nur lokal arbeiten, können Sie auch auf **Skip** klicken. Analyseapplikationen, die im Local Mode erstellt wurden, müssen Sie nachträglich auf einer der unterstützten Plattformen publizieren, um sie für andere Benutzer zur Verfügung zu stellen.

Auch beim Starten im SAP Lumira Documents Mode wird ein Anmeldedialogfeld angezeigt. Es erlaubt Ihnen, entweder eine Verbindung zu einer BI Platform herzustellen, offline zu arbeiten (dies bedeutet, dass Sie von der BI Platform getrennt sind) oder das Design-Tool zu beenden.



Abbildung 7.5 Anmeldung an der BI Platform im SAP Lumira Designer

Ist die Anwendung SAP Lumira Designer vollständig geladen, wird Ihnen ein Willkommensbild angezeigt. Hier haben Sie verschiedene Optionen dazu, was Sie als Nächstes tun können (siehe Abbildung 7.6):

- Einstieg in die Arbeit mit dem SAP Lumira Designer (**Getting Started**)
- Eine neue Analyseapplikation erstellen (**Create Analysis Application**)
- Eine kurz zuvor bearbeitete Analyseapplikation öffnen (**Recently-Used Analysis Application**)
- Nützliche Links mit weiteren Informationen zum SAP Lumira Designer ansehen (**Useful Links**)

Um eine neue Analyseapplikation zu erstellen, haben Sie zwei Möglichkeiten (siehe Abbildung 7.7):

1. Klicken Sie im Menü auf **Application • New** ❶.
2. Klicken Sie auf den Button **Create Analysis Application** im Willkommensbild ❷.

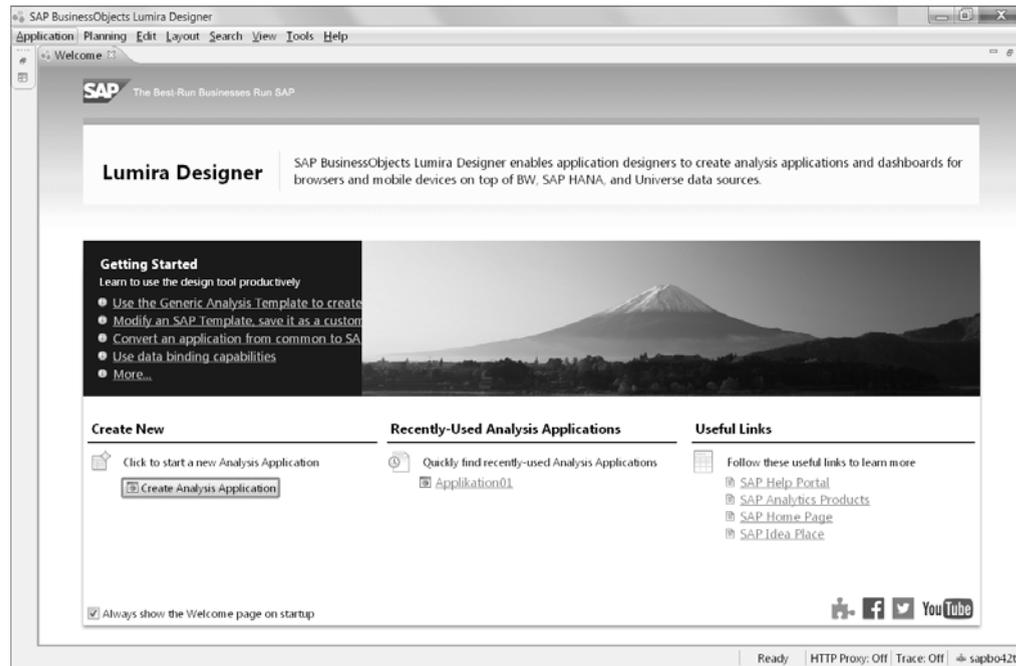


Abbildung 7.6 Willkommensbild im SAP Lumira Designer

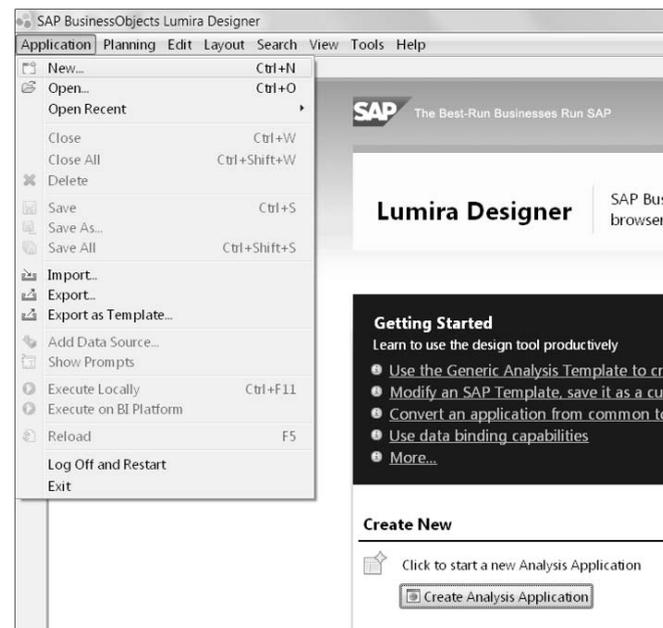


Abbildung 7.7 Neue Analyseapplikation anlegen

Anschließend müssen Sie (sofern Sie sich im ersten Schritt z. B. an der BI Plattform angemeldet haben) einen Ordner (**Folder**) auswählen bzw. anlegen (siehe Abbildung 7.8).

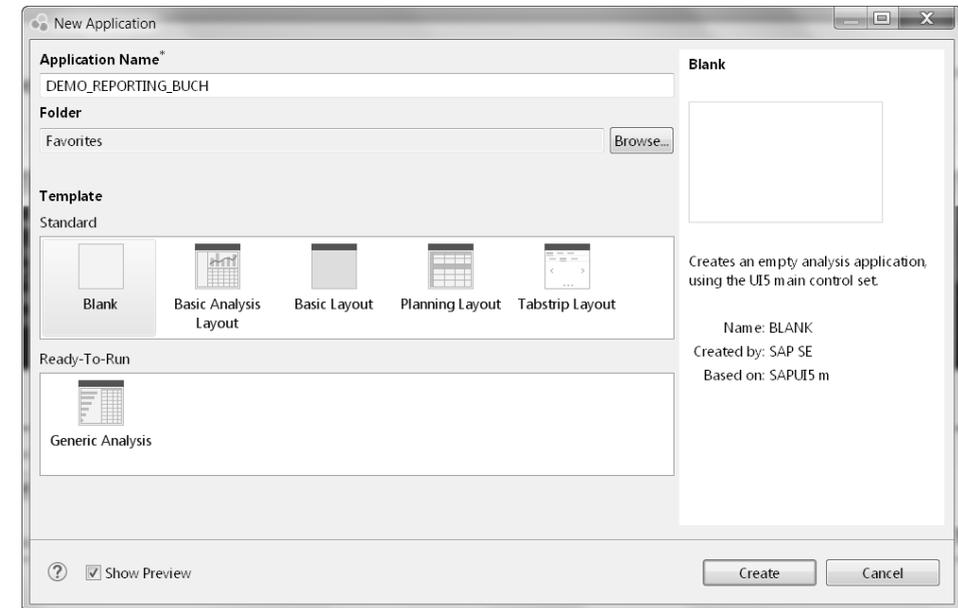


Abbildung 7.8 Einstellungen für die neue Analyseapplikation vornehmen

Zudem müssen Sie für jede Analyseapplikation einen Namen angeben (**Application Name**). Sie können für Ihre Applikation entweder ein Standard-Template oder das Ready-to-Run-Template nutzen:

- **Blank**
leere Analyseapplikation
- **Basic Analysis Layout**
Template mit Tabellen, Grafiken und einem Filterbereich
- **Basic Layout**
Template mit benutzerdefinierten und erweiterbaren CSS-Container-Komponenten, wie z. B. Kopfzeile, Seitenkopf oder Seiteninhalt
- **Planning Layout**
Template mit Tabellen, Grafiken und einem Filterbereich, optimiert für die Daten aus der Planung
- **Tabstrip Layout**
Template beinhaltet einen **LAYOUT CONTAINER** mit **HEADER** und **PAGE**. Innerhalb der **PAGE** beinhaltet es einen **PAGEBOOK CONTAINER** mit mehreren **TABS** auf 2 Leveln

■ Generic Analysis

Template mit Tabellen, Grafiken und einem Filterbereich, bei dem Sie die Datenquelle nach dem Ausführen der Applikation einbinden können.

Durch einen Klick auf **Create** wird das gewählte Template als Grundlage für Ihre Analyseapplikation verwendet. In unserem Beispiel verwenden wir das leere Template, sodass eine leere Analyseapplikation im SAP Lumira Designer angezeigt wird (siehe Abbildung 7.9).

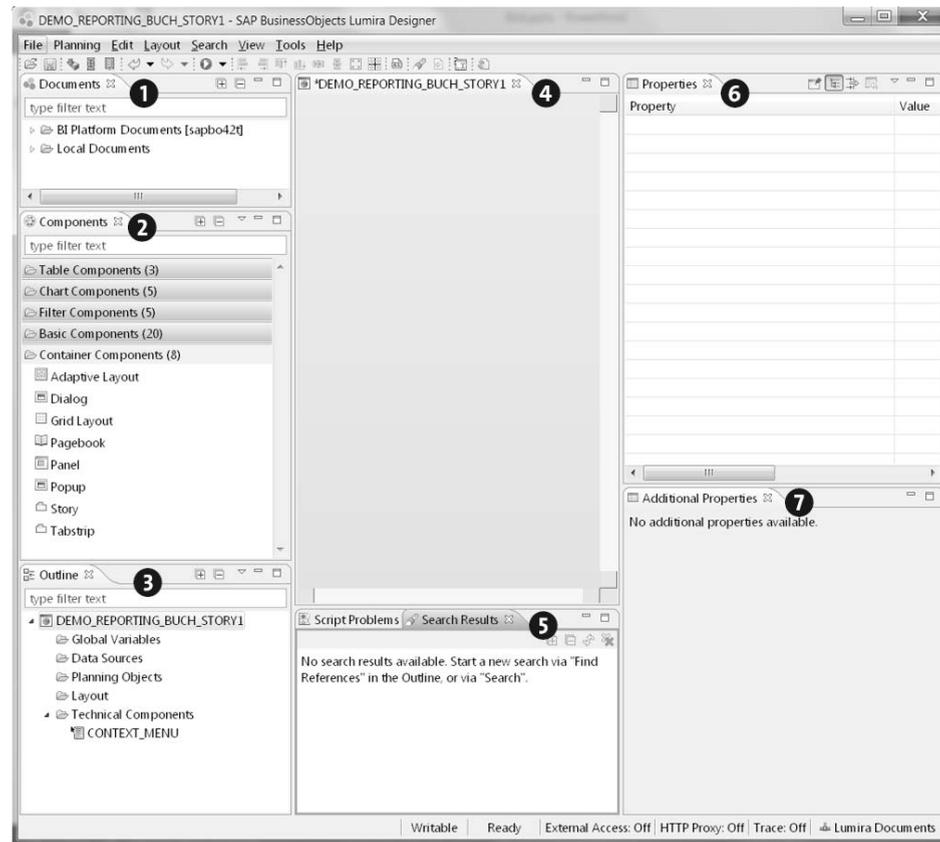


Abbildung 7.9 Leere Analyseapplikation

7.2.2 Die einzelnen Views

Die Entwicklungsumgebung des SAP Lumira Designers setzt sich aus den folgenden Views zusammen:

- ❶ Dokumente (**Documents**)
- ❷ Komponenten (**Components**)
- ❸ Übersicht (**Outline**)

- ❹ Designbereich
- ❺ Fehlermeldungen in den Skripten (**Script Problems**)
- ❻ Eigenschaften (**Properties**)
- ❼ Weitere Eigenschaften (**Additional Properties**)

In den folgenden Abschnitten gehen wir auf diese Views ausführlicher ein.

Dokumente

Der Dokumenten-View steht Ihnen nur im SAP Lumira Documents Mode zur Verfügung. Er ist die zentrale Sicht für das Ändern und Erstellen von Dokumenten. Diese Ansicht hat entweder einen oder zwei Wurzelknoten, von denen einer immer den Ordner **SAP Lumira Documents** und der andere die BI Plattform darstellt. Wenn Sie offline arbeiten, wird der Knoten **BI Platform Documents** nur angezeigt, wenn Sie bereits zuvor mit einer BI Plattform verbunden waren und mindestens ein Dokument heruntergeladen haben. Andernfalls wird der Knoten ausgeblendet.

Anstatt eine neue Applikation anzulegen, müssen Sie hier zuerst ein Dokument (siehe Abbildung 7.10) anlegen oder ein vorhandenes Dokument auswählen. Dieses Dokument dient sozusagen als Container für Ihre Applikation. Wenn Sie bereits ein Dokument in SAP Lumira Discovery erstellt haben, finden Sie hier sowohl alle erstellten Datenquellen als auch Ihre erstellten Stories, inklusive Visualisierungen.

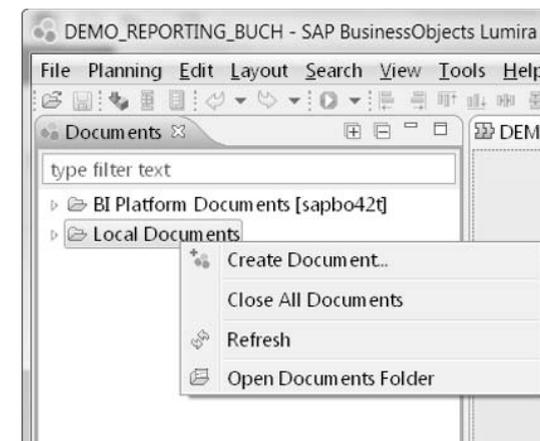


Abbildung 7.10 Dokument im Dokumenten-View erstellen

Folgende Datenquellen können Sie im Dokumenten-View nutzen:

- SAP BW-Systeme aus dem Dialog SAP Logon
- SAP HANA
- Offline-Datenquellen, die von SAP Lumira Discovery erstellt und innerhalb des Dokuments gespeichert wurden



Composites

Eine der wichtigsten Neuerungen im SAP Lumira Designer, die Sie nur über den Dokumenten-View nutzen können, ist das *Composite*. Das Composite erstellen Sie, wie Ihre Applikation, auch im Dokumenten-View in Ihrem Dokument (siehe Abbildung 7.11). Das Composite ist eine neue App im App-Konzept, mit dem Sie wiederverwendbare Inhalte erstellen können, die dann in der Palette des Komponenten-Views angezeigt werden. Im Composite können Sie außer den Planungsobjekten sämtliche Komponenten nutzen, die Sie auch in einer Applikation nutzen können. Sie können hier sowohl komplexe Templates erstellen als auch einzelne Teile wie Kopfzeilen oder Fußzeilen. Somit sind Sie in der Lage, generische und konfigurierbare Komponenten nur mit Bordmitteln, ohne den Weg über das SDK, zu erstellen.

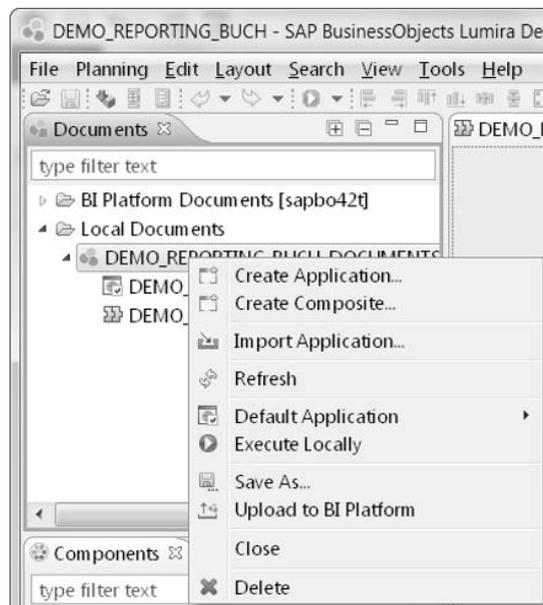


Abbildung 7.11 Composite im Dokumenten-View erstellen

Komponenten

Der *Komponenten-View (Components)* stellt die Komponenten zur Gestaltung von Applikationen zur Verfügung. In Abbildung 7.10 sehen Sie die im SAP Lumira Designer zur Verfügung stehenden Komponenten, die wir uns in Abschnitt 7.6, »Komponenten«, im Detail ansehen. Über den Komponenten-View können Sie die Eigenschaften der einzelnen Komponenten nicht verändern. Er dient ausschließlich zur Auswahl einer Komponente, die anschließend per Drag & Drop in die Applikation eingefügt wird.

Die Komponenten sind nach ihrer Funktion und Verwendungsart in Gruppen gegliedert:

- **Tabellenkomponenten (Table Components)** dienen der Darstellung von Daten in Tabellenform aus einer zugeordneten DataSource. Sie benötigen daher immer eine DataSource, um abgebildet werden zu können.
- **Grafikkomponenten (Chart Components)** dienen der Darstellung von Daten in einer Grafik aus einer zugeordneten DataSource. Sie benötigen daher immer eine DataSource, um abgebildet werden zu können.
- **Filterkomponenten (Filter Components)** werden zur Filterung von Komponenten genutzt, die einer DataSource zugeordnet sind.
- **Basiskomponenten (Basic Components)** können zur Gestaltung von Applikationen verwendet werden. Im Gegensatz zu den Tabellen-, bzw. Grafikkomponenten benötigen diese Komponenten keine DataSource. Über die im SAP Lumira Designer verfügbare Skriptsprache können Sie diesen Basiskomponenten aber Funktionen zuweisen, z. B. für die Filterung von Werten einer DataSource. Die Funktionen können Sie über den bereitgestellten Skripteditor (siehe Abschnitt 7.2.3, »Skripteditor«, und Abschnitt 7.3, »BI Action Language (BIAL)«) bearbeiten.
- **Container-Komponenten (Container Components)** werden zur Formatierung und Positionierung der einzelnen Komponenten einer Applikation verwendet. Auch diese Komponenten benötigen keine Verbindung zu einer DataSource, da sie keine Daten anzeigen.

Übersicht

Der Übersichts-View (**Outline**) gibt einen Überblick über alle Komponenten einer Applikation in Form einer Komponentenhierarchie. Zusätzlich werden die globalen Variablen (**Global Variables**), **Data Sources**, Planungsobjekte (**Planning Objects**), das Layout und die technischen Komponenten (**Technical Components**) einer Applikation aufgeführt. Abbildung 7.12 zeigt eine beispielhafte Übersicht.

Über das Kontextmenü der DataSources können Sie z. B. den *Initial View* der DataSource editieren. Außerdem können Sie über den Übersichts-View jede Komponente einzeln selektieren, was bei vielen verwendeten Komponenten im Designbereich aufwendig sein kann.

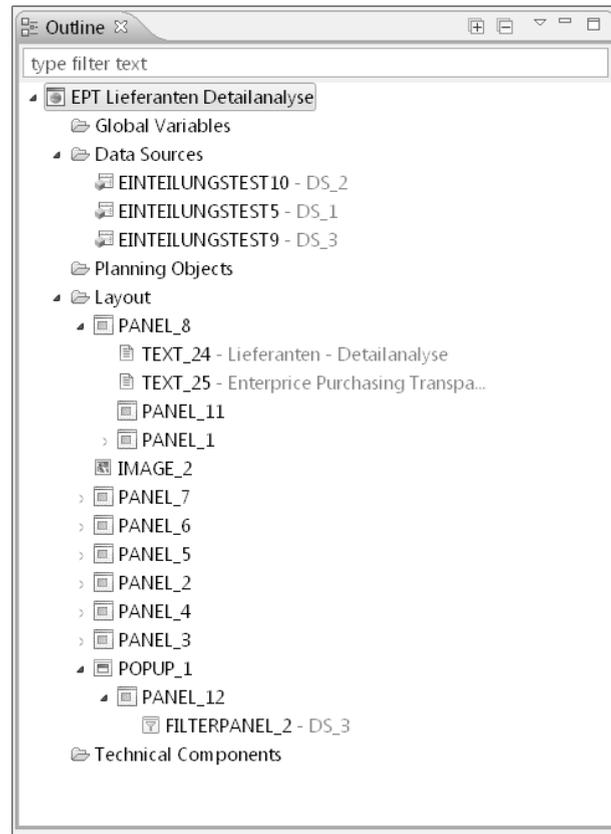


Abbildung 7.12 Übersichts-View im SAP Lumira Designer

Designbereich

Der Designbereich dient zur Positionierung der Komponenten einer Applikation. Abbildung 7.13 zeigt eine Komponente mit variabler Höhe und Breite.

Das Konzept hinter der Definition der Position und der Größe einer Komponente erklären wir anhand der Breite. Um die Position und die Breite zu bestimmen, sind drei Werte notwendig:

- Abstand der Komponente zum linken Rand
- Breite der Komponente
- Abstand der Komponente zum rechten Rand

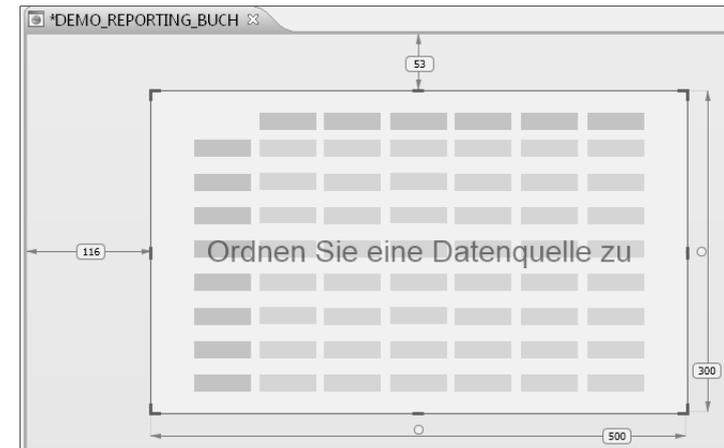


Abbildung 7.13 Komponente im Designbereich

Eine zentrale Layout-Eigenschaft einer Applikation ist, dass die Größe des Ausgabebereichs variabel ist. Daher ist immer einer der drei Werte ein variabler Wert (**Auto**). In Abbildung 7.12 ist dies der rechte Rand der Komponente. Im Designbereich werden jeweils nur die fix festgelegten Werte angezeigt. Um einen dieser fixen Werte in einen variablen Wert umzuändern, genügt es, im Designbereich auf diesen fixen Wert zu klicken.

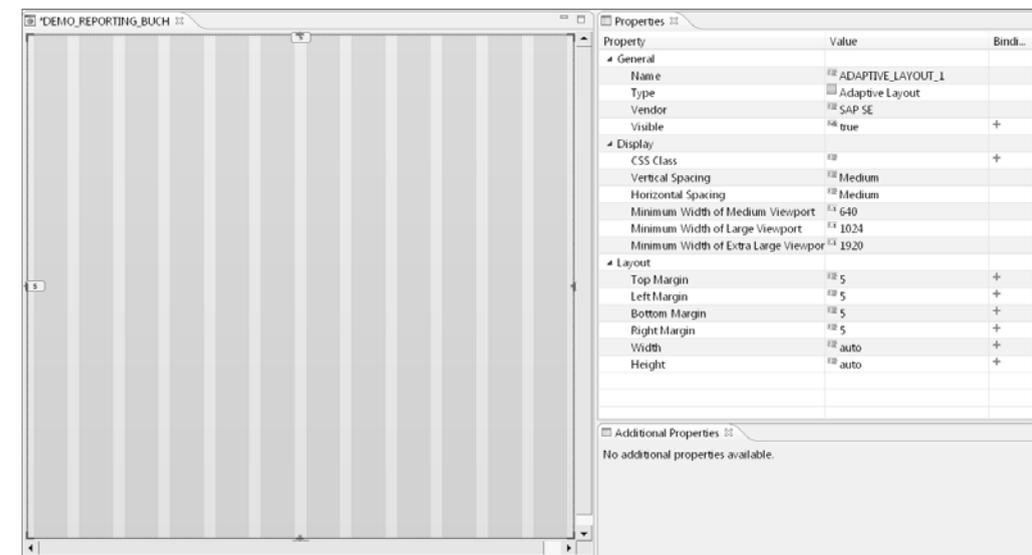


Abbildung 7.14 Eigenschaften des Adaptive Layouts

Eine neue Funktionalität im SAP Lumira Designer ist das anpassungsfähige Layout (**Adaptive Layout**). Diese Funktionalität ermöglicht es allen Komponenten, die Teil

dieser Container Komponente sind, sich automatisch anzupassen, wenn sich die Bildschirmgröße ändert. Somit passt sich eine Applikation automatisch jedem Endgerät (Desktop, Tablet, Smartphone) an und muss nicht mehr separat für jedes Endgerät optimiert werden. Abbildung 7.14 zeigt die Eigenschaften des Adaptive Layouts.

Die Positionierung der Komponenten im Designbereich erfolgt per Drag & Drop. Neue Komponenten fügen Sie ebenfalls per Drag & Drop aus dem Komponenten-View hinzu. Der Quelltext einer Applikation wird als XML-Dokument gespeichert und kann prinzipiell auch manuell angepasst werden. Dies wird aber von SAP nicht unterstützt und auch nicht empfohlen. Um Ergänzungen an den Applikationen vorzunehmen, kann das Software Development Kit des SAP Lumira Designers verwendet werden.

Eigenschaften

Über den Eigenschaften-View (**Properties**) können die Parameter der Komponenten einer Applikation, die Applikation selbst und die zur Applikation definierten Data-Sources verändert werden. Abbildung 7.15 zeigt links die Eigenschaften einer Basis-Komponente vom Typ **Button** und rechts die Eigenschaften einer Analysekomponente vom Typ **Chart**.

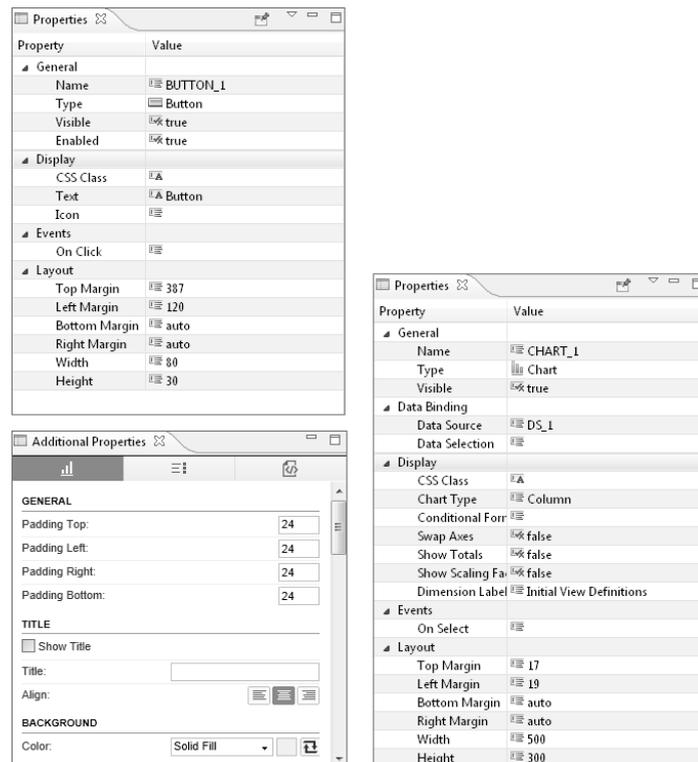


Abbildung 7.15 Eigenschaften zweier Komponenten vom Typ »Button« und »Chart«

Die Eigenschaften sind in Gruppen angeordnet, z. B. **General** und **Display**. Abhängig von der gewählten Komponente, sind mehr oder weniger Eigenschaften verfügbar. Zusätzlich stehen für einige Komponenten auch erweiterte Eigenschaften (**Additional Properties**) zur Verfügung. Die Gruppen Allgemein (**General**) und **Layout** sind für alle aktuell verfügbaren Komponenten vorhanden. Über die Gruppe Ereignis (**Events**) können Sie den Komponenten Ereignisse zuweisen, die durch den Anwender ausgelöst werden können. Hierzu werden über den Skripteditor sogenannte *Skripte* hinterlegt (siehe den folgenden Abschnitt 7.2.3, »Skripteditor«). Die Skripte werden ausgeführt, wenn das entsprechende Ereignis eintritt oder durch den Benutzer ausgelöst wird.

7.2.3 Skripteditor

Der Skripteditor ist das zentrale Werkzeug innerhalb des SAP Lumira Designers, um Applikationen und Komponenten interaktiv zu gestalten. Die meisten Komponenten und auch die Applikation selbst bieten die Möglichkeit, in Skripten die Eigenschaften unter **Events** zu hinterlegen. Im Skripteditor werden Ihnen die Befehle der *BI Action Language* (siehe Abschnitt 7.3) angeboten, die für die jeweilige Komponente oder die Applikation zur Verfügung stehen. Diese können je nach gewählter Komponente variieren.

Mithilfe des Skripteditors erstellen Sie die Skripte, die aufgrund von ausgelösten oder eingetroffenen Ereignissen auszuführen sind. Jede Komponente hat hierbei einen Namen, auf den Sie in den Skripten referenzieren können. Abbildung 7.15 zeigt die Bearbeitung einer Skriptanweisung zur Komponente **Filterpanel** und dem Ereignis **On Apply**.

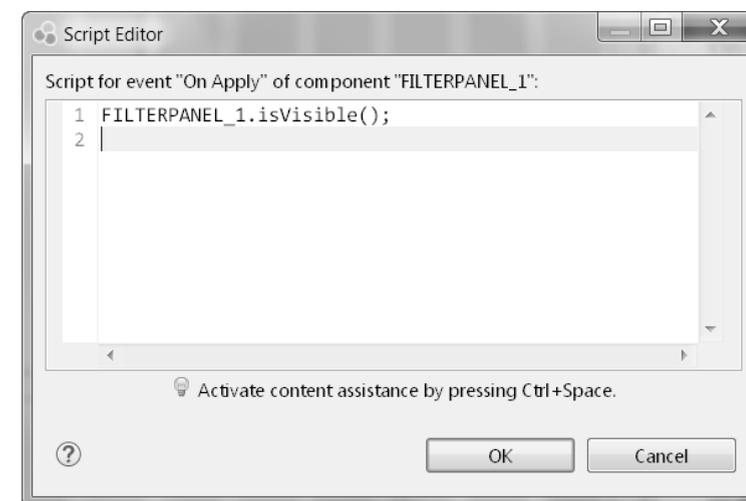


Abbildung 7.16 Skripteditor im SAP Lumira Designer

Neben den Komponenten stellen auch die DataSources einer Applikation und die Applikation selbst eine Instanz dar. In Abbildung 7.14 wird mithilfe der Methode `isVisible()` die Sichtbarkeit der Komponente verändert. Die verfügbaren Methoden sind innerhalb des Skripteditors dokumentiert. Die zu einer Komponente zur Verfügung stehenden Methoden können Sie über die Tastenkombination `[Strg] + Leertaste` ermitteln. Im Kontextmenü werden Ihnen alle bereitstehenden Methoden angezeigt. Auch können Sie Befehle zur Steuerung der Ablauflogik verwenden, derzeit zumindest IF-THEN-ELSE-Schleifen. Aktuell haben Sie nicht die Möglichkeit, wiederverwendbare Skriptanweisungen zu erstellen.

7.2.4 Applikationseigenschaften

Sie können die Applikationseigenschaften auf einem der folgenden Wege aufrufen:

- Klicken Sie auf eine leere Stelle des Arbeitsbereichs Ihrer Applikation.
- Klicken Sie im Bereich **Outline** auf den Namen Ihrer Applikation.

Anschließend werden die Applikationseigenschaften wie in Abbildung 7.17 dargestellt.

In der Kategorie **General** finden Sie die folgenden Eigenschaften:

- **Name**
Hier wird der Name Ihrer Applikation angezeigt.
- **Type**
Hier steht der Typ der Applikation; dies ist konstant **Analysis Application**.
- **Description**
Hier finden Sie eine Beschreibung zu Ihrer Applikation. Diese erscheint, wenn Sie im BI Launchpad mit dem Mauszeiger über den Namen der Applikation fahren, oder im Eigenschaftendialog.
- **Folder**
Dieser zeigt den Pfad zu Ihrer Applikation an.
- **Created by**
Hier sehen Sie, wer die Applikation erstellt hat.
- **Creation Time** und **Last Modification Time**
Datum und Uhrzeit der Erstellung bzw. letzten Änderung der Applikation erfahren Sie hier.
- **CUID**
Dies ist der eindeutige Schlüssel der Applikation.
- **Content Version**
Hier sehen Sie die Lumira-Designer-Version, mit der die Applikation zuletzt gespeichert wurde.

Property	Value	Bindi...
General		
Name	DEMO_REPORTING_BUCH	
Type	Application	
Description		
Folder	Favorites	
Created By	mste	
Creation Time	7/5/17 5:05 PM	
Last Modification Time	7/8/17 12:03 AM	
CUID	Ac1qwdKlIFZPgZH5kCeD8j0	
Content Version	2001	
Behavior		
Maximum Number of Steps	0	
Drag and Drop Between Components	false	
Disable Browser Context Menu	false	
Display		
SAPUI5 m Mode	true	
Browser Title		
Theme	Blue Crystal	
Compact Form Factor	true	
Custom CSS		
Position of Message Button	Bottom Right	
Loading Indicator Delay	1000	
Displayed Message Types	Warnings and Errors	
Prompts		
Merge Prompts	false	
Force Prompts on Startup	false	
Prompt Settings	<no prompts available>	
Maximum Number of Members	100	
Design Studio 1.x Bookmark Loading	Hide and Keep Prompts	
Planning		
Events		

Property	Value	Bindi...
Planning		
Planning Connection	<none>	
Planning Model		
Events		
On Variable Initialization		
On Startup		
On Background Processing		
On Before Prompts Submit		

Abbildung 7.17 Applikationseigenschaften

In der Kategorie **Behavior** finden Sie die folgenden Eigenschaften:

- **Maximum Number of Steps**
Diese Eigenschaft ist im Standard auf Null gesetzt bzw. deaktiviert. Wenn Sie den Wert z. B. auf 5 setzen, hat der Benutzer der Applikation die Möglichkeit, fünf Schritte innerhalb der Applikation zurückzugehen. Der empfohlene Maximalwert ist 20. Sobald die Eigenschaft auf einen Wert größer als Null gesetzt ist, wird auch die Reset-Funktion aktiviert.
- **Drag and Drop between Components**
Hier können Sie festlegen, ob Drag-&Drop-Operationen zwischen verschiedenen Komponenten erlaubt sind. Diese Eigenschaft ist standardmäßig auf `false` gesetzt, was bedeutet, dass Drag-&Drop-Operationen nicht zwischen Komponenten ausgeführt werden können. Drag-&Drop-Operationen innerhalb einer einzigen Komponente sind immer noch möglich, wenn z. B. die Crosstab-(Component-) Level-Eigenschaft auf `true` gesetzt ist. Setzen Sie diese Eigenschaft auf `true`, wenn Sie Drag-&Drop-Operationen zwischen den Komponenten aktivieren möchten (z. B. zwischen dem Navigationsfeld und dem Crosstab).
- **Disable Browser Context Menu**
Hier sehen Sie, ob das Browser-Kontextmenü für die Applikation global deaktiviert ist. Diese Eigenschaft ist standardmäßig auf `false` gesetzt, was bedeutet, dass es nicht deaktiviert ist. Setzen Sie diese Eigenschaft auf `true`, wenn Sie das Browser-Kontextmenü für die Applikation deaktivieren möchten.

In der Kategorie **Display** finden Sie die folgenden Eigenschaften:

- **SAPUI5 m Mode**
Applikationen im 2.x-Format (`.lumx`) basieren immer auf der `SAPUI5.m`-Bibliothek. Diese Eigenschaft ist standardmäßig auf `true` gesetzt, was bedeutet, dass diese Applikation im `SAPUI5.m`-Modus läuft. Diese Eigenschaft würde nur auf `false` stehen, wenn Sie eine ältere Applikation mit dem 1.6-Format (`.biapp`) im Kompatibilitätsmodus geöffnet haben.
- **Theme**
Ein Theme bestimmt das Aussehen aller Komponenten. Auch wenn es spezielle Desktop- und Mobile-Themes gibt, können Sie Ihre Applikationen, unabhängig vom Theme, auf mobilen Geräten anzeigen. Die Auswahl des Themes bestimmt nur das Aussehen der Komponenten.
- **Compact Form Factor**
Der Compact Form Factor ist der Rendering-Modus für Desktop-Style-Anwendungen im `SAPUI5.m`-Modus. Diese Eigenschaft ist standardmäßig auf `true` gesetzt, was bedeutet, dass Komponenten dieser Applikation im Desktop-Stil des `SAPUI5.m`-Modus ausgeführt werden. Setzen Sie diese Eigenschaft auf `false`, wenn Sie eine Applikation nur für mobile Endgeräte erstellen möchten.

- **Custom CSS**
Hier setzen Sie den Pfad zu der zentralen CSS-Datei (siehe Abschnitt 7.5, »Cascading Style Sheets (CSS)«).
- **Position of Message Button**
Falls Fehler oder Warnungen zur Laufzeit auftreten, erscheint ein Button, über den Sie sich weitere Details anzeigen lassen können. Diese Eigenschaft bestimmt, wo der Button erscheint.
- **Loading Indicator delay**
Hier wird der Zeitverzug des Ladezeigers in Millisekunden beim Starten der Applikation angezeigt.
- **Position of Message Window**
Falls ein Benutzer weitere Details zu einer Warnung oder einem Fehler durch einen Klick auf den Message Button anfordert, erscheint das Message-Fenster. Diese Eigenschaft steuert die Position des Message-Fensters.
- **Displayed Message Types**
Hier steuern Sie, welche Art von Meldungen angezeigt werden. Sie können zwischen nur Fehlermeldungen, Fehlermeldungen und Warnungen oder keine Meldungen auswählen.

In der Kategorie **Prompts**, den Eingabeaufforderungen aus der Datenquelle, finden Sie die folgenden Eigenschaften:

- **Merge Prompts**
Diese Eigenschaft gibt an, ob dieselben Eingabeaufforderungen (Variablen), die in verschiedenen Datenquellen in einer Applikation verwendet werden, zusammengeführt werden. Wenn diese Eigenschaft auf `true` gesetzt ist, kann die gleiche Variable, die in mehreren Abfragen verwendet wird, nur einmal eingegeben werden. Wenn diese Eigenschaft dagegen auf `false` gesetzt ist, muss die gleiche Variable, die in mehreren Abfragen verwendet wird, für jede Abfrage separat eingegeben werden.
- **Force Prompts on Startup**
Diese Einstellung steuert, ob Prompts zwingend angezeigt werden, bevor die Applikation startet.
- **Prompt Settings**
Mit dieser Einstellung können Sie steuern, welche Prompts angezeigt werden und in welcher Reihenfolge.
- **Maximum Number of Members**
Hier können Sie die Anzahl angezeigter Werte hinter einem Prompt einschränken.
- **Design Studio 1.x Bookmark Loading**
Diese Eigenschaft gibt das Verhalten von Status und Aufforderungen beim Laden

eines Design Studio 1.x Bookmarks an. Auf diese Weise können Sie, das Verhalten beim Laden eines Design Studio 1.x Bookmarks bestimmen.

In der Kategorie **Planning** finden Sie die folgenden Eigenschaften:

- **Planning Connection**
In dieser Eigenschaft sehen Sie die Verbindung für Planungsszenarien.
- **Planning Model**
Diese Eigenschaft zeigt Ihnen den Namen des Planungsmodells.

In der Kategorie **Events** finden Sie die folgenden Eigenschaften:

- **On Variable Initialization**
Hier können Sie Standardwerte für Prompts hinterlegen.
- **On Startup**
Hier können Sie BIAL-Code hinterlegen, der vor dem eigentlichen Starten der Applikation ausgeführt wird.
- **On Background Processing**
Hier können Sie Code hinterlegen, der als Hintergrundprozess ausgeführt wird (siehe Abschnitt 7.9.1, »Background Processing«).
- **On Before Prompts Submit**
Dieses Ereignis wird aufgerufen, wenn der Button **OK** im Dialogfeld **Eingabeaufforderungen** angeklickt wird und bevor Variablen übergeben werden. Dies ermöglicht es Ihnen, Variablenwerte zwischen Datenquellen zu kopieren und bedeutet, dass Sie nicht unbedingt die gleiche Variable für jede Abfrage manuell eingeben müssen.

7.3 BI Action Language (BIAL)

Der SAP Lumira Designer bietet eine eigene Programmiersprache, um Flexibilität zur Laufzeit durch Erweiterungen zu erreichen. Diese Skriptsprache wird BI Action Language (BIAL) genannt und orientiert sich in ihrer Syntax an JavaScript. Beachten Sie allerdings, dass BIAL nicht mit JavaScript gleichgesetzt werden kann, da BIAL eine Untermenge von JavaScript darstellt. Eine typische Anweisung in BIAL sieht folgendermaßen aus:

```
<Komponente>.<Methode>( <Argumente> )
```

So können Sie beispielsweise folgendes Skript ausführen, wenn auf einen Button geklickt wird, um die Überschrift einer Textkomponente zu setzen:

```
TEXT_1.setText("Hallo Welt");
```



Groß- und Kleinschreibung in BIAL

Beachten Sie, dass für BIAL die Groß- und Kleinschreibung wichtig ist. So gilt schon dann ein Befehl als fehlerhaft, wenn auch nur ein einziger Buchstabe entsprechend falsch geschrieben wurde. Wird eine Zeile als fehlerhaft ausgewiesen, wird wiederum der ganze Skriptblock als fehlerhaft angezeigt. Damit werden alle Zeilen vor und nach dem Fehler nicht ausgeführt.

In BIAL gelten die folgenden allgemeinen Regeln:

- Alle Befehle enden mit einem Semikolon (;).
- Überflüssige Leerzeichen werden ignoriert, außer wenn diese innerhalb von Strings vorkommen.
- Groß- und Kleinschreibung ist wichtig.
- Einzelne Codezeilen können mit einem doppelten Schrägstrich (//) am Anfang der Zeile auskommentiert werden.
- Sie können mehrere Zeilen gleichzeitig auskommentieren, wenn Sie am Anfang einen Schrägstrich und einen Stern (/*) und am Ende einen Stern und einen Schrägstrich platzieren (*//).

Wie in jeder anderen Programmiersprache wird empfohlen, dass Sie den Code immer mit Kommentaren versehen, um die Wartung zu vereinfachen. In BIAL existieren verschiedene Datentypen, die in Tabelle 7.2 mit je einem Beispiel aufgelistet sind.

Datentyp	Beispiel
String	"Hallo Welt"
Integer	123
Float	123.45
Boolean	True
String Array	["A", "B"]
Integer Array	[1, 2]
JSON	{"key": "value"}

Tabelle 7.2 Datentypen in BIAL

7.3.1 Wizard

Um die Skriptsprache BIAL im SAP Lumira Designer zu nutzen, müssen Sie kein Programmierer sein; eigentlich müssen Sie nicht einmal den Code selbst in die Applika-

tion eintippen. Der SAP Lumira Designer bietet einen Wizard an (siehe Abbildung 7.18), der Ihnen immer geeignete Elemente für das Coding zur Verfügung stellt. Um den Wizard aufzurufen, verwenden Sie einfach das Tastenkürzel **[Strg] + Leertaste**, während Sie den Code im Skripteditor implementieren.

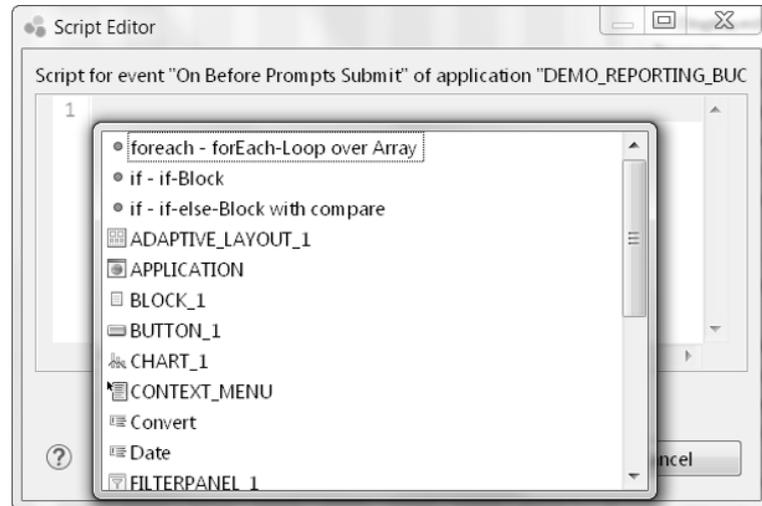


Abbildung 7.18 Code-Wizard im Skripteditor

7.3.2 Variablen

Variablen werden genutzt, um (Zwischen-)Ergebnisse, z. B. von Berechnungen, zu speichern. Es gibt zwei Arten von Variablen: globale und lokale. Globale Variablen werden verwendet, um Werte zu speichern, auf die verschiedene Komponenten zugreifen sollen. Sie können globale Variablen auch verwenden, um mithilfe einer URL Parameter an den SAP Lumira Designer zu übergeben. Solche URL-Parameter sind Parameter, die der URL der Applikation angehängt werden. Damit können Sie Werte von einer Lumira-Designer-Applikation in die nächste übertragen. Um globale Variablen als URL-Parameter nutzen zu können, muss der Name der Variablen mit einem großen X beginnen. Um eine globale Variable anzulegen, gehen Sie wie folgt vor:

1. Gehen Sie in der Übersicht (**Outline**) zu den globalen Variablen (**Global Variables**), und erstellen Sie im Kontextmenü eine neue globale Variable.
2. Wählen Sie die neue Variable aus, und gehen Sie auf die Eigenschaften.
3. Geben Sie einen Namen, einen Typ und einen Standardwert für die Variable ein. In der Spalte **Expose as URL Parameter** geben Sie mit `true` oder `false` an, ob die Variable als URL-Parameter dienen soll (siehe Abbildung 7.19).

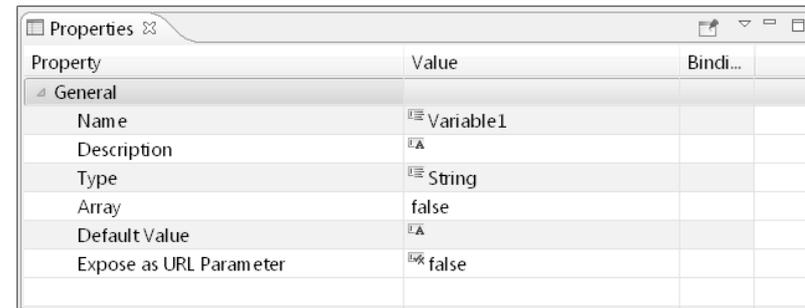


Abbildung 7.19 Globale Variablen anlegen

Nachdem Sie eine globale Variable angelegt haben, können Sie von jeder Komponente aus auf diese zugreifen, indem Sie anstelle eines konkreten Wertes den Namen der globalen Variablen im Skripteditor angeben. Mit der folgenden Referenz könnten Sie beispielsweise die in Abbildung 7.19 angelegte Variable in einer Komponente verwenden:

```
TEXT_1.setText(Variable1);
```

Sie können auch den Wert einer globalen Variablen verändern, indem Sie von einer beliebigen Komponente aus z. B. den folgenden Code anwenden:

```
Variable1 = "Hallo Welt";
```

Die Lebensdauer einer globalen Variablen ist unbegrenzt, solange die Applikation geöffnet ist. Im Gegensatz dazu beschränkt sich die Lebensdauer einer lokalen Variablen auf eine einzige Prozedur einer Komponente, z. B. das Skript, das beim Klick auf einen Button ausgeführt wird. Sie können eine lokale Variable definieren, indem Sie das Schlüsselwort `var` am Anfang einer Zeile verwenden, gefolgt von dem Namen der zu erstellenden Variablen und dem initialen Wert:

```
var Variable2 = "Hallo Welt";
```

7.3.3 Operatoren

In Tabelle 7.3 finden Sie einen Überblick über die Operatoren, die Ihnen im SAP Lumira Designer zur Verfügung stehen.

Operator	Bedeutung	Beispiel	Ergebnis
+	Konkatenation	AB + CD	ABCD
+	Summe	1 + 2	3

Tabelle 7.3 Operatoren in BIAL

Operator	Bedeutung	Beispiel	Ergebnis
+	Konkatenation	1 + A	1A
-	Subtraktion	3-1	2
*	Multiplikation	2*2	4
/	Division	4/2	2
< (=)	kleiner als	1<2	true
> (=)	größer als	1>2	false
==	überprüft, ob die Operanden gleich sind.	1 == 1	true
!=	überprüft, ob die Operanden ungleich sind.	1 != 1	false
&&	logisches Und	true && true	true
	logisches Oder	true false	true
!	logisches Nicht (Umkehrung)	!true	false

Tabelle 7.3 Operatoren in BIAL (Forts.)

7.3.4 Arrays und Schleifen

Ein Array ist eine Ansammlung von Objekten eines gemeinsamen Datentyps. Sie können beispielsweise die Zahlen 1, 2 und 3 gemeinsam in einer einzigen Variablen speichern, nämlich in einem Array. Im SAP Lumira Designer deklarieren Sie ein solches Array beispielsweise wie folgt:

```
var array = [1, 2, 3];
```

Um auf die Elemente eines Arrays zuzugreifen, müssen Sie durch die Elemente des Arrays iterieren; dies wird *Schleife* genannt. Eine Schleife wird im SAP Lumira Designer mit dem Schlüsselwort `forEach` erstellt. Im folgenden Beispiel wird ein Array aus Zahlen deklariert, um anschließend mit einer Schleife auf jedes einzelne Element zuzugreifen und diese Elemente aufzuaddieren:

```
var sum = 0;
var array = [1, 2, 3];
array.forEach(function(element, index) {
  sum = sum + element;
});
```

Mit dieser Technik können Sie z. B. durch Wertelisten iterieren, indem Sie die Werteliste als Array deklarieren.

7.4 Applikation erstellen

In diesem Abschnitt zeigen wir Ihnen, wie Sie eine einfache Applikation im SAP Lumira Designer erstellen. Als Basis hierfür nehmen Sie eine BW Query. Um diese BW Query verwenden zu können, müssen Sie zunächst eine neue DataSource anlegen. Hierzu klicken Sie entweder im Menü des SAP Lumira Designers mit der linken Maustaste auf den Button **Add DataSource** oder mit der rechten Maustaste auf den Knoten **Data Source** im Übersichts-View und wählen anschließend im Kontextmenü den Eintrag **Add DataSource**. Es öffnet sich der Dialog zum Einfügen einer DataSource (siehe Abbildung 7.20).

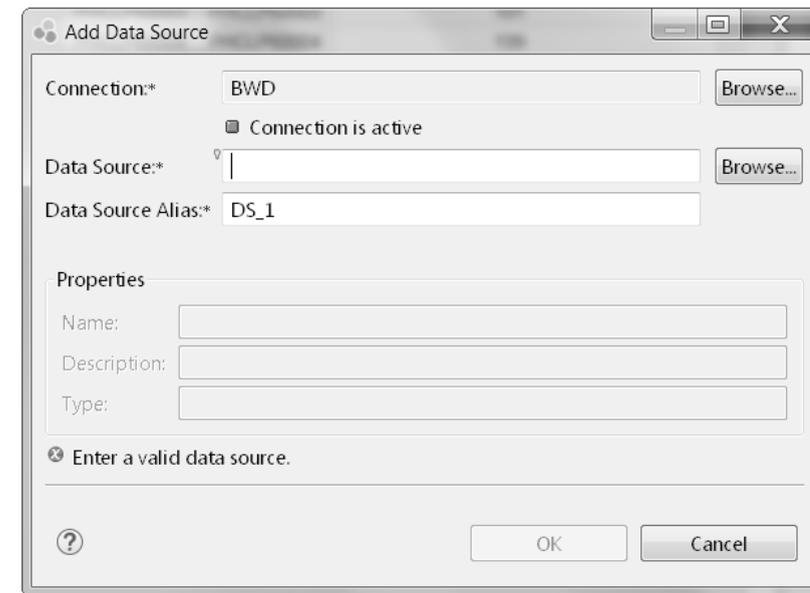


Abbildung 7.20 DataSource anlegen

Zunächst wählen Sie über den Button **Browse** hinter dem Feld **Connection** eine Datenverbindung aus. Diese Datenverbindung müssen Sie zuvor auf dem BI Server in der Central Management Console angelegt haben. Nachdem Sie die Datenverbindung bestimmt haben, können Sie die BW Query für Ihre Applikation auswählen. Hierzu klicken Sie auf den Button **Browse** hinter dem Feld **Data Source**. Im letzten Schritt können Sie den Namen der DataSource verändern. Standardmäßig wird als Name immer `DS_<Zahl>` verwendet. Zu einer Applikation können Sie eine beliebige Anzahl an DataSources anlegen.

Die angelegte DataSource steht Ihnen anschließend im Übersichts-View zur Verfügung. Klicken Sie mit der rechten Maustaste auf die DataSource, können Sie im Kontextmenü **Edit Initial View** auswählen, um den Erstauffrisch der BW Query zu verän-

dern. Sie müssen hierzu nicht in den BEx Query Designer wechseln, sondern können einfache Änderungen direkt im Initial View vornehmen (siehe Abbildung 7.21).

Durch einen Klick auf **OK** und **Create Crosstab** können Sie direkt aus dem Initial View heraus eine Tabelle auf der Basis der BW Query erstellen lassen. Möchten Sie die Datenquelle für eine andere Komponente verwenden, verlassen Sie den Initial View durch einen Klick auf **OK**.

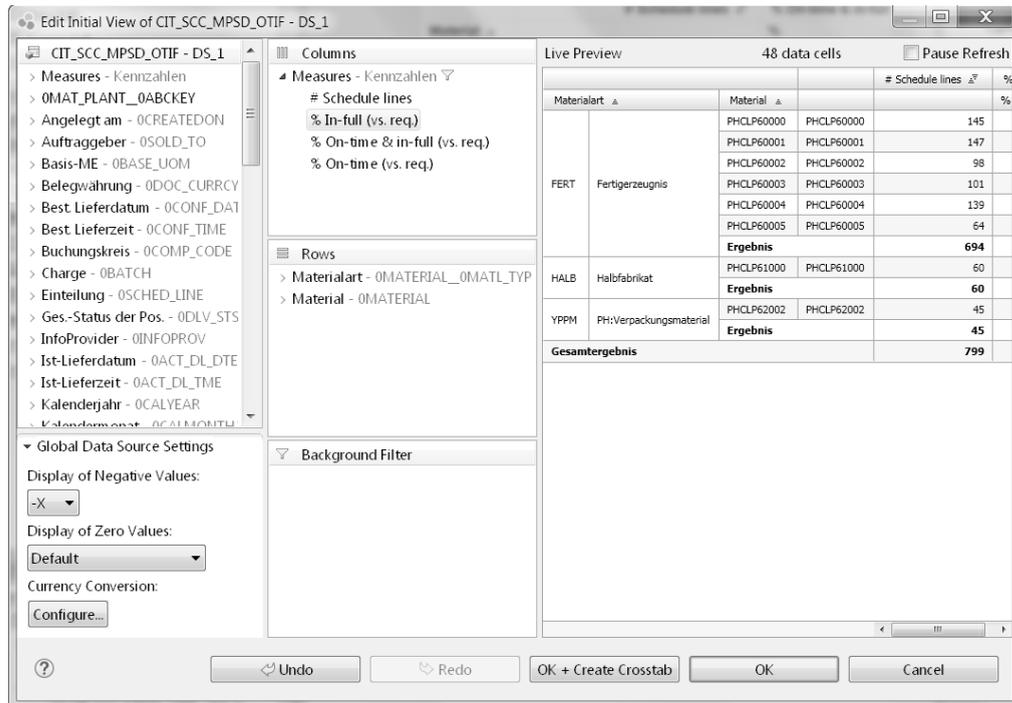


Abbildung 7.21 Erstaufriß einer BW Query ändern

Als Beispiel möchten wir jetzt dem Designbereich eine Chart-Komponente hinzufügen. Hierzu ziehen Sie die Komponente **Chart** aus dem View **Analytic Components** in den Designbereich. Anschließend können Sie die zuvor erstellte DataSource per Drag & Drop mit dem Chart verbinden (siehe Abbildung 7.22). Alternativ können Sie im Eigenschaften-View den Parameter **Data Binding** auf den Wert **Data Selection** umstellen.

Neben dem Chart soll noch eine Komponente Filter Panel eingefügt werden, damit der Benutzer die verwendeten Daten filtern kann. Auch für das Filter Panel können Sie die DataSource entweder per Drag & Drop oder im Eigenschaften-View festlegen.

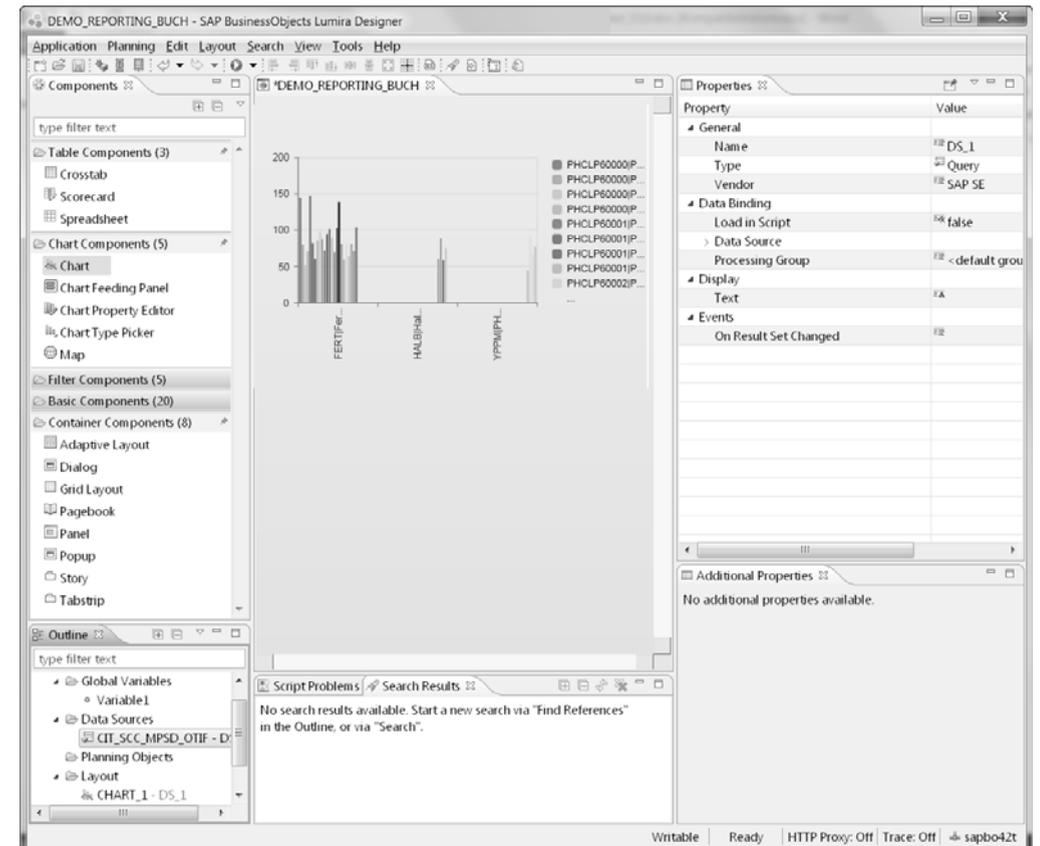


Abbildung 7.22 DataSource per Drag & Drop mit einer Komponente verbinden

Um die erstellte Applikation zu testen, können Sie die Applikation entweder direkt lokal auf dem Client oder über die BI Plattform ausführen (siehe Abbildung 7.23). Um die Applikation über die BI Plattform ausführen zu können, muss zuvor das Add-on für den SAP Lumira Designer in Ihrem System installiert worden sein.



Abbildung 7.23 Wahl der Ausführungsoption

Die erstellte Applikation können Sie bei deren Ausführung im Browser sofort testen und sehen, ob z. B. die von Ihnen vorgesehene Filterung auf bestimmte Werte eines Merkmals hin möglich ist (siehe Abbildung 7.24).

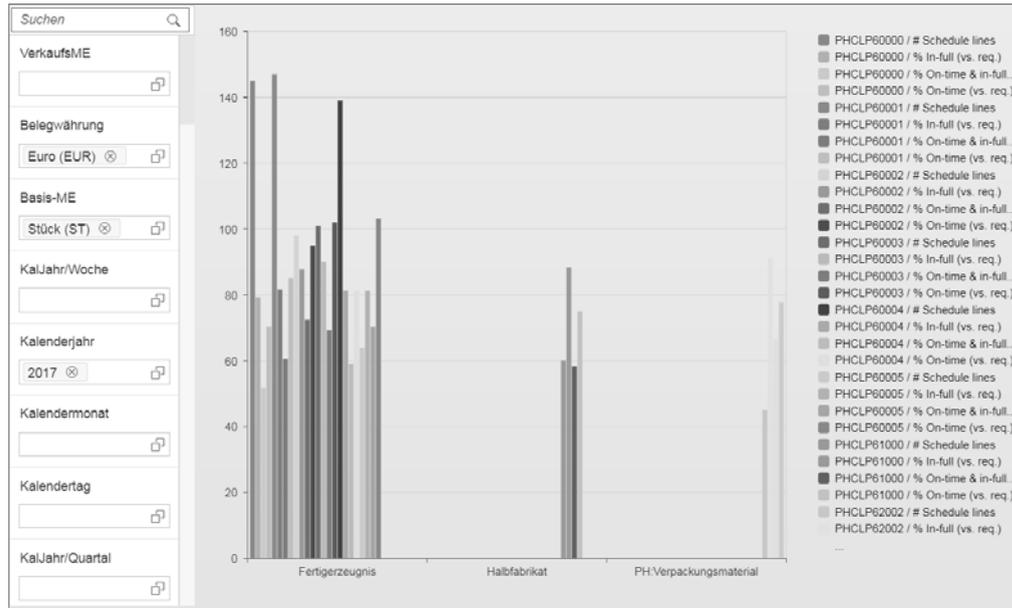


Abbildung 7.24 Erstellte Applikation ausführen

Für dieses einfache Beispiel haben Sie bisher keine Skripte benötigt, da das Filter Panel automatisch auf die Datenquelle wirkt. Da beide Komponenten, Chart und Filter Panel, auf derselben Datenquelle basieren, werden ausgewählte Filterwerte auch in das Chart übernommen. Sie könnten aber eine Tabelle zu unserer Beispielapplikation hinzufügen. Diese Tabelle basiert auf derselben Datenquelle wie die anderen Komponenten. Über ein Skript könnten Sie dann steuern, dass beim Klick auf eine Zeile der Tabelle automatisch auch das Chart und das Filter Panel beeinflusst werden. Es lassen sich somit relativ einfach komplexe Interaktionen des Benutzers in einer Applikation umsetzen.

Positiv fällt auf, ist, dass alle Komponenten, sobald Sie sie mit DataSources verbunden haben, direkt die Daten anzeigen, sodass Sie direkt im SAP Lumira Designer einen Eindruck von der fertigen Applikation gewinnen (WYSIWYG-Funktion). Diese Funktion ist innerhalb des Clients möglich, da in der Entwicklungsumgebung eine eigene Instanz der Engine zur Generierung der Applikation integriert ist.

Im Beispiel haben wir die Komponenten bisher wahllos im Designbereich positioniert. Zur genaueren Positionierung und Formatierung der Komponenten bietet es sich an, die Layout-Komponenten zu verwenden, z. B. das Grid Layout.

7.5 Cascading Style Sheets (CSS)

Die Formatierung wird im SAP Lumira Designer mit Cascading Style Sheets (CSS) vorgenommen. CSS ist eine Technik, die im Zusammenhang mit HTML-Seiten im Web bekannt wurde. Mit CSS können Sie das Aussehen der Komponenten im SAP Lumira Designer anpassen. In der Regel gibt es zwei Eigenschaften einer Komponente, die es Ihnen erlauben, CSS zu verwenden:

- CSS Style
- CSS Class

Über die Eigenschaft **CSS Style** können Sie Ihren CSS-Code direkt in der Komponente eingeben. Dies ist nützlich, um während der Dashboard-Entwicklung die Auswirkungen Ihres CSS-Codes direkt nachzuvollziehen, ganz ohne Verzögerungen. Um beispielsweise die Schriftgröße des Textes in einer Textkomponente zu verändern, geben Sie für die Eigenschaft **CSS Style** einer Textkomponente den folgenden Code ein:

```
font-size: 15px;
```

Sie können diesen Code auch zentral in eine separate Datei auslagern und auf einzelne Elemente aus der zentralen CSS-Datei zurückgreifen. Mit dieser Vorgehensweise können Sie Ihren CSS-Code in vielen Applikationen verwenden. Eine Änderung der zentralen Datei wirkt sich dann auf alle Applikationen aus, die auf diese Datei verweisen. Um diese Dateien nutzen zu können, müssen Sie die Eigenschaft **CSS Class** verwenden. Hier können Sie keinen Code eingeben, sondern stattdessen geben Sie an dieser Stelle den Namen einer CSS-Klasse an, die Sie zuvor in einer zentralen CSS-Datei definiert haben. Den Namen der zentralen CSS-Datei geben Sie in den Eigenschaften der Applikation im Feld **Custom CSS** ein.

Um mithilfe einer zentralen CSS-Datei beispielsweise die Schriftgröße einer Textkomponente zu verändern, gehen Sie wie folgt vor:

1. Erstellen Sie eine Datei mit der Erweiterung ».css« (z. B. mit dem Notepad++), beispielsweise **custom.css**.
2. Geben Sie als Inhalt der CSS-Datei den folgenden Code ein:

```
.klasse_1 {
font-size: 15px;
}
```

Das Wort nach dem Punkt bezeichnet den Namen Ihrer selbst definierten CSS-Klasse. Innerhalb der geschweiften Klammern befindet sich die Eigenschaft, die Sie mit dieser Klasse anwenden möchten.

- Laden Sie die Datei auf Ihre Plattform hoch. Falls Sie lokal arbeiten, reicht es, wenn Sie die Datei in demselben Ordner speichern wie Ihre Applikation.
- Verweisen Sie auf die zentrale CSS-Datei, indem Sie in den Applikationseigenschaften unter **Custom CSS** auf Ihre CSS-Datei verweisen. Wenn Sie lokal arbeiten, geben Sie hier einfach den Namen der CSS-Datei an, z. B. **custom.css**.
- Schließlich müssen Sie der Textkomponente den Namen der CSS-Klasse bekannt machen, indem Sie in den Eigenschaften der Komponente die Eigenschaft **CSS Class** mit dem Namen der Klasse ergänzen, in unserem Beispiel ist das `klasse_1`.

Beachten Sie, dass das CSS zwischen Groß- und Kleinschreibung unterscheidet.



Nutzung der CSS Styles vermeiden

Wir empfehlen Ihnen, immer auf die Eigenschaft **CSS Styles** zu verzichten und Ihren CSS-Code stattdessen in einer zentralen CSS-Datei zu verwalten, um den Wartungsaufwand zu minimieren.

7.5.1 Layout von Standardkomponenten anpassen

Um das Aussehen existierender Komponenten zu verändern, müssen Sie die existierenden Klassen der jeweiligen Komponenten überschreiben. Wie die existierenden Klassen zu den Standardkomponenten aussehen, erfahren Sie, indem Sie den Standard-CSS-Code mit einem CSS Inspector inspizieren. Ein CSS Inspector hilft Ihnen dabei, das CSS einer Internetseite zu durchsuchen. Im Internet Explorer ist z. B. ein CSS Inspector in den sogenannten *Developer Tools* integriert. Sie können die Developer Tools des Internet Explorers aufrufen, indem Sie die Taste `F12` betätigen, während Sie eine bestimmte Webseite besuchen.

7.5.2 Button anpassen

Im Folgenden zeigen wir Ihnen, wie Sie das Aussehen eines Buttons ändern können. Abbildung 7.25 zeigt einen Button im Standard-Layout.

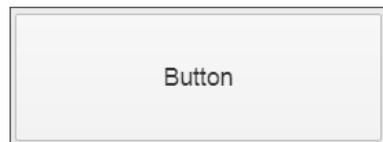


Abbildung 7.25 Standard-Button

Um die Standard-CSS-Styles eines Buttons herauszufinden, öffnen Sie eine Lumira-Designer-Applikation, die einen Button enthält, öffnen z. B. die Developer Tools des Internet Explorers und klicken auf die Registerkarte **CSS**. Klicken Sie dann in der

Symbolleiste auf den Mauszeiger (☞) und anschließend auf einen Button in der Lumira-Designer-Applikation. Sie sehen eine CSS-Analyse, ähnlich der Analyse in Abbildung 7.26.

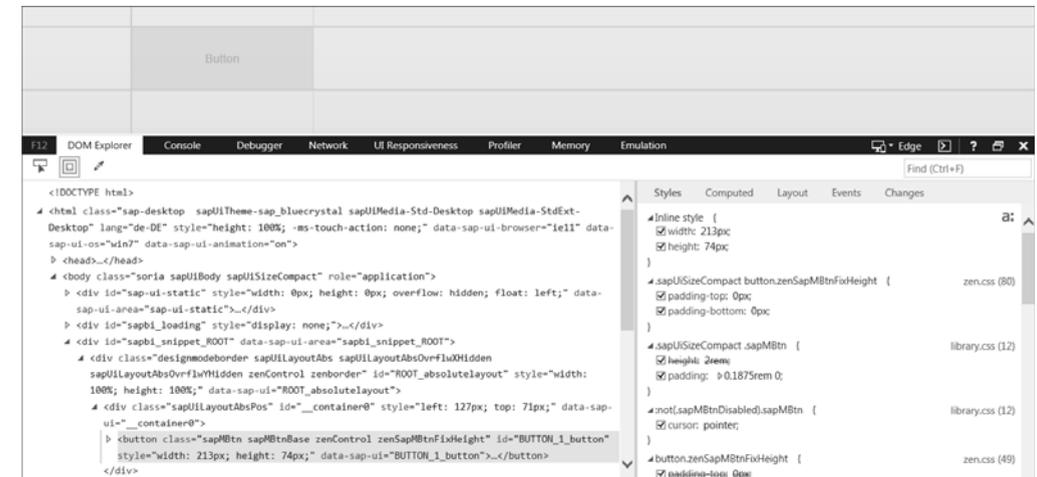


Abbildung 7.26 CSS-Inspektion mit den Developer Tools des Internet Explorers

Auf der linken Seite sehen Sie das Document Object Model (DOM) der HTML-Seite Ihrer Lumira-Designer-Applikation. Auf der rechten Seite sehen Sie alle CSS-Eigenschaften aus allen CSS-Klassen, die das Aussehen des Buttons verändern. In der DOM-Darstellung sehen Sie blau markiert die Zeile im HTML-Code, die für die Button-Komponente verantwortlich ist. Sie können diese Zeile (siehe Listing 7.1) über das Kontextmenü in die Zwischenablage kopieren:

```
<button class="sapMBtn sapMBtnBase zenControl zenSapMBtnFixHeight"
id="BUTTON_1_button"
style="width: 213px; height: 74px;" data-sap-ui="BUTTON_1_button">
<div class="sapMBtnDefault sapMBtnHoverable sapMBtnInner sapMBtnText
sapMFocusable sapMIE"
id="BUTTON_1_button-inner" style="height: 74px;">
<span class="sapMBtnContent" id="BUTTON_1_button-content"
style="line-height: 74px;">Button</span>
<div class="sapMBtnFocusDiv"></div></div></button>
```

Listing 7.1 HTML-Code zur Definition eines Buttons

Der HTML-Code beschreibt vor dem Gleichheitszeichen die Attribute des Elements Button. Wichtig für den SAP Lumira Designer sind die Codeausschnitte für die Attribute `class` und `style`. Hinter dem Attribut `class` werden alle CSS-Klassen aufgelistet, die den Button beeinflussen. Diese Klassen sind hierarchisch aufgebaut, und ihre

Werte werden entsprechend vererbt. Hinter dem Attribut `style` werden die CSS-Codezeilen aufgelistet, die nur für das aktuelle Element (hier den Button) gelten, und nicht z. B. aus einer Klasse gelesen werden.



!important-Schlüsselwort

Die Attribute, die hinter dem Attribut `style` stehen, können Sie in Ihrem Code nur überschreiben, indem Sie vor dem Ende der Zeile `!important` hinzufügen, z. B.:

```
height: 40px !important;
```

Den Developer Tools können Sie entnehmen, dass die Eigenschaft zur Definition des Button-Hintergrunds von der Klasse `sapMBtn` bestimmt wird (siehe Abbildung 7.27). Wenn Sie die Checkbox zu der Eigenschaft `background-color` deaktivieren, sehen Sie sofort eine Veränderung der Button-Komponente: Der Button-Hintergrund wird von **transparent** auf **RGB (240, 240, 240)** gesetzt.

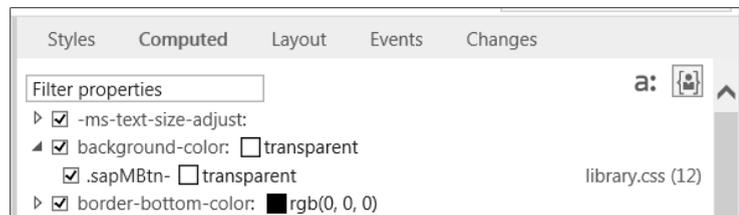


Abbildung 7.27 Klasse für den Hintergrund eines Buttons

Indem Sie die Klasse aus Abbildung 7.27 in Ihre CSS-Datei übernehmen, können Sie also den Hintergrund verändern. Beispielsweise können Sie die Farbe des Hintergrunds mit dem folgenden Code ändern:

```
.sapMBtn {
background-image: #009900;
```

Haben Sie diesen CSS-Code in einer ausgelagerten Datei eingegeben und in Ihrer Applikation auf diese CSS-Datei verwiesen, wird Ihr Button wie in Abbildung 7.28 dargestellt.



Browser-Cache löschen

Wenn Sie mit dem CSS arbeiten, wird das CSS Ihrer Applikation von Ihrem Internetbrowser im Speicher (Cache) gehalten. Dies führt dazu, dass Sie Änderungen Ihres CSS-Codes nicht sofort in der Anwendung sehen können. Um die neuen Änderungen zu sehen, müssen Sie den Browser-Cache löschen. Im Internet Explorer können Sie den Browser-Cache löschen, indem Sie bei gehaltener `[Strg]`-Taste mit der rechten

Maustaste auf Ihre Applikation klicken und anschließend im Kontextmenü **Refresh** auswählen. Dies funktioniert sowohl bei der Ausführung Ihrer Lumira-Designer-Applikation als auch während der Entwicklung innerhalb des SAP Lumira Designers.



Abbildung 7.28 Veränderter Button

Durch das Überschreiben der Eigenschaft `background-image` werden alle Buttons in Ihrer Lumira-Designer-Applikation an den neuen CSS-Code angepasst. Um eine bessere Kontrolle darüber zu haben, welche Buttons von dem CSS-Code betroffen sind, können Sie an dieser Stelle auch Ihre eigenen Klassen verwenden. Durch den folgenden Code können Sie eine Klasse namens `ButtonKlasse_1` definieren:

```
.ButtonKlasse_1 .sapMBtn {
background-image: #009900;
```

Anschließend können Sie durch Zuweisung der Klasse `ButtonKlasse_1` bestimmen, welche Buttons in Ihrer Applikation keinen Hintergrund aufweisen sollen.

Automatische Codegenerierung

Es gibt im Internet viele Webseiten, über die Sie im WYSIWYG-Modus CSS-Code generieren können, z. B. <http://css3.me> oder <http://selfcss.org>. Sie können dort beispielsweise eine Farbe aus einer Farbpalette auswählen, woraufhin Ihnen die Seite anzeigt, wie der CSS-Code zu dieser Farbe aussieht. Diesen CSS-Code können Sie dann direkt in Ihre CSS-Dateien übernehmen.

CSS-Editoren nutzen

Schreiben Sie Ihren CSS-Code mit einem Editor, wie z. B. Notepad++. Dieser unterstützt Sie u. a. durch ein farbiges Syntax-Highlighting und hilft Ihnen so, keinen syntaktischen Fehler zu begehen.

7.6 Komponenten

Um Analyseapplikationen zu gestalten, stellt der SAP Lumira Designer analog zum BEx Web Application Designer verschiedene Komponententypen zur Verfügung, die wir im Folgenden beschreiben.

7.6.1 Analytische Komponenten

Abbildung 7.29 zeigt die verschiedenen analytischen Komponenten, die Ihnen im SAP Lumira Designer zur Verfügung stehen und die wir im Folgenden beschreiben.

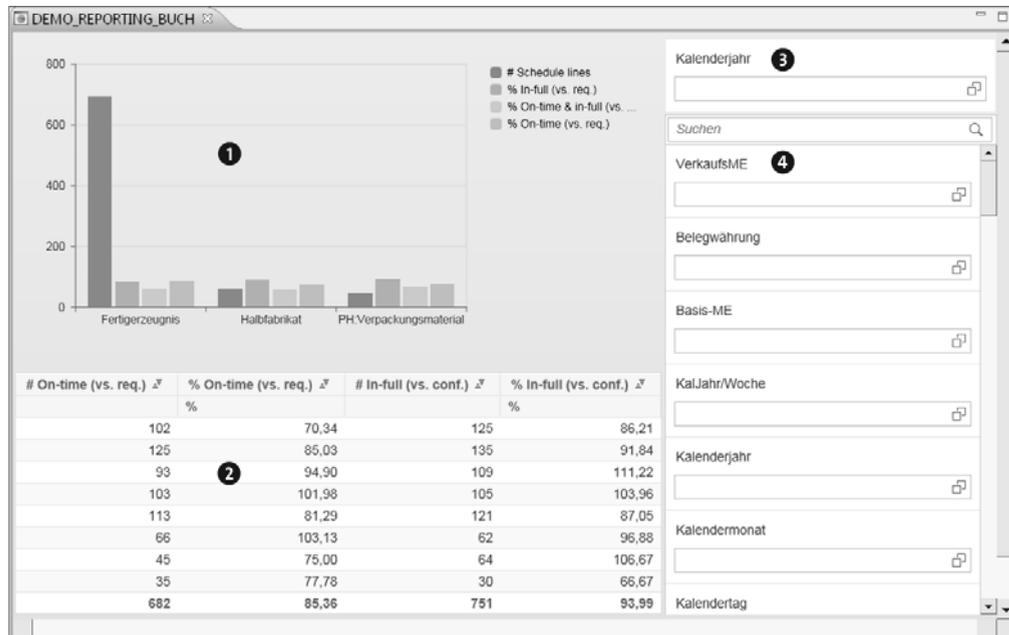


Abbildung 7.29 Analytische Komponenten im SAP Lumira Designer

Über die Komponente **Chart 1** stehen Ihnen die folgenden Chart-Typen zur Auswahl:

- **Column**
Zur Verfügung stehen ein einfaches Säulendiagramm, ein Kombinations-Chart, auch mit zwei x-Achsen, ein gestapeltes Säulendiagramm, auch auf 100 % normiert, sowie ein Säulendiagramm mit zwei x-Achsen.
- **Bar**
Dieser Chart-Typ bietet ein einfaches Balkendiagramm, ein Kombinations-Chart, ein gestapeltes Balkendiagramm, auch auf 100 % normiert, sowie ein Balkendiagramm mit zwei x-Achsen.
- **Stacked Column**
Die gestapelten Säulendiagramme zeigen die Beziehung jedes Elements zum Ganzen an und vergleichen den Betrag jeder Komponente mit einer Gesamtmenge über Kategorien.
- **Stacked Bar**
Gestapelte Balkendiagramme werden verwendet, um die Beziehung jedes Elements zum Ganzen anzuzeigen, wobei der Betrag jeder Komponente mit einer Gesamtmenge über Kategorien verglichen wird.

- **Marimekko**
Das Marimekko ist ein Balkendiagramm, das sowohl durch Breite als auch durch Höhe kommuniziert. Damit kann jeder Balken zwei Maße mit der Größe des Balkens sowie zwei Dimensionen durch Stapeln von Werten von Abmessungen in einem einzigen Balken und den Achsenbeschriftungen vermitteln.
- **Waterfall**
Zur Verfügung steht ein Wasserfalldiagramm, das auch horizontal sowie gestapelt verwendet werden kann.
- **Bullet**
Ein Bullet-Chart-Typ kann verwendet werden, um den Fortschritt auf ein Ziel anzuzeigen. Sie müssen zuerst Grenzwerte für die Maßnahme festlegen, mit der Sie arbeiten.
- **Line**
Zur Verfügung stehen ein einfaches Liniendiagramm, ein horizontales Liniendiagramm sowie ein Liniendiagramm mit zwei x-Achsen.
- **Stacked Area**
Gestapelte Flächendiagramme können für mehrere Datenreihen mit Teil-zu-Ganz-Beziehungen verwendet werden oder um eine kumulative Wertereihe zu zeigen.
- **Pie**
Zur Verfügung stehen ein Tortendiagramm sowie ein Diagramm mit multiplen »Torten«.
- **Donut**
Der Donut-Chart-Typ wird verwendet, um den Prozentsatz einer Art von Dimension innerhalb eines Maßes zu zeigen.
- **Scatter Plot**
Verwenden Sie einen Scatterplot, um herauszufinden, ob es eine Korrelation zwischen zwei spezifischen Maßnahmen gibt.
- **Bubble**
Um die Korrelation zwischen drei Maßnahmen zu sehen, verwenden Sie ein Blasendiagramm. Sie zeigen ein Maß auf jeder Achse, und ein drittes Maß spiegelt sich in der Größe der Blase.
- **Heat Map**
Die Teile auf einer Heat Map ändern die Farbe auf der Grundlage des Wertes der Maßnahme. Heat Maps sind eine bequeme Weise, um ein großes Volumen der Datenpunkte in einem Diagramm darzustellen, das einfach zu lesen ist.
- **Tree Map**
Das Tree Map Chart funktioniert ähnlich wie die Heat Map. Der Unterschied besteht darin, dass sich die Dimensionskacheln, je nach Wert der Maßnahme,

ändern. Tree Maps sind auf eine einzige Maßnahme und Dimension pro Diagramm beschränkt.

■ Numeric Point

Verwenden Sie ein Numeric Point Chart, um Summen auf einen Blick zu zeigen. Numerische Punkte können auch mit Schwellenwerten erhöht werden. Wenn die Schwellen aktiviert sind, ändert sich die Farbe der Zahl, um zu zeigen, wo sie in den Schwellenwert fällt.

■ Line Chart for Time Series

Eine weitere Möglichkeit, um Trends im Laufe der Zeit zu zeigen, ist ein Zeitreihenplan. Durch die Wahl eines Zeitreihendiagramms können Sie leicht nach verschiedenen Zeiträumen filtern. Diese Grafik würde z. B. schnell zeigen, ob ein Projekt vor oder hinter dem Zeitplan liegt und wann die größten Abweichungen vom Zeitplan auftraten.

■ Scatter Plot for Time Series

Verwenden Sie dieses Chart, um herauszufinden, ob es eine Korrelation zwischen zwei spezifischen Maßnahmen über einen bestimmten Zeitraum gibt.

■ Bubble Chart for Time Series

Um die Korrelation zwischen drei Maßnahmen über einen bestimmten Zeitraum zu sehen, verwenden Sie dieses Blasendiagramm. Es zeigt ein Maß auf jeder Achse, und ein drittes Maß spiegelt sich in der Größe der Blase.

■ Tag Cloud

Ein Tag Cloud Chart ist eine monodimensionale Visualisierung, die die Daten als Wörter darstellt, wobei die Schriftgröße ihr relatives Gewicht im Datensatz verkörpert.

■ Box Plot

Box-Plots zeigen die Verteilung von Daten auf der Grundlage einer Fünf-Zahlen-Zusammenfassung: Minimum, erstes Quartil, Median, drittes Quartil und Maximum. Wenn Datensätze einzelne Punkte mit extrem hohem Maximum oder überraschend niedrigem Minimum aufweisen, werden sie als Ausreißer angezeigt.

■ Radar

Radardiagramme sind nützlich, um die Verteilung der Werte über die Dimensionen zu sehen und Ausreißer auszumachen. Diese Diagramme können verwendet werden, um zu sehen, welche Werte für eine bestimmte Dimension hoch oder niedrig sind.

■ Funnel

Die Trichtergrafik können Sie nutzen, um sich den prozentualen Anteil jedes Wertes anzeigen zu lassen. Je größer der Wert, desto weiter oben im Trichter wird er angezeigt.

■ Parallel Coordinates

Das ist eine Visualisierung von hochdimensionalen Strukturen und multivariater Daten. Anders als im Streudiagramm, in dem zwei Koordinatenachsen rechtwinklig zueinander angeordnet sind, verlaufen sie hier parallel und in gleichem Abstand. Jede Linie von links nach rechts entspricht dabei einem Datenpunkt und wird durch einen Polygonzug mit Ecken auf den parallelen Achsen dargestellt.

■ Tree

Beim Tree Chart sind Sie auf zwei Dimensionen beschränkt, wobei die erste Dimension immer den Wurzelknoten darstellt.

Die Auswahl des Chart-Typs erfolgt in den Eigenschaften der Chart-Komponente über **Chart Configuration** im Bereich **Display** (siehe Abbildung 7.30).

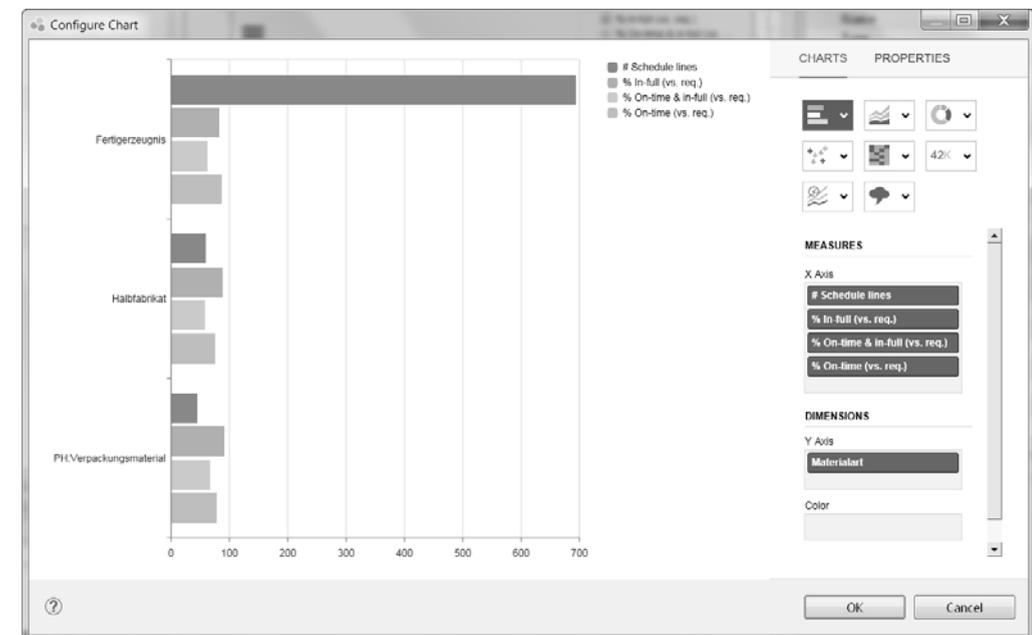


Abbildung 7.30 Auswahl des Chart-Typs

Die in Abbildung 7.29 gezeigte Komponente Tabelle (Crosstab)  ermöglicht die tabellarische Darstellung von Daten. Kreuztabellen zeigen Datenpunkte nur als Werte an, anstatt eine visuelle Darstellung dieser Werte vorzusehen. Infolgedessen sind sie nützlich, wenn Ihre Analyse davon abhängt, genaue Werte zu betrachten oder Daten aus mehreren Maßnahmen mit verschiedenen Skalen oder Maßeinheiten zu untersuchen. Sie können auch bedingte Formatierungen in Kreuztabellen verwenden, um nennenswerte Datenpunkte zu identifizieren und Summen in den Zeilen oder Spalten hinzuzufügen.

Der Dimensionsfilter (**Dimension Filter**) ③ erlaubt die Filterung auf einen Wert einer Dimension. Hierzu wählen Sie in den Eigenschaften des Dimensionsfilters die Datenquelle und das Ziel des Filters aus. Anschließend wählen Sie die Dimension in der Datenquelle aus, auf die der Filter angewendet werden soll. Über den Dimensionsfilter können Sie immer nur auf eine Dimension hin filtern, während Sie beim Filter Panel auch auf mehrere Dimensionen hin filtern können.

Das **Filter Panel** ④ erlaubt demgegenüber die Filterung auf eine oder mehrere Dimensionen einer Datenquelle. Hierzu wählen Sie die Datenquelle in den Eigenschaften des Filter Panels aus, woraufhin automatisch alle Dimensionen der Datenquelle in das Filter Panel eingefügt werden. Sie können nur das Filtern oder beide Möglichkeiten in einem Filter Panel anbieten. Navigieren bedeutet in diesem Zusammenhang, dass Dimensionen dynamisch den Zeilen und/oder Spalten einer Tabelle hinzugefügt werden können. Diese Eigenschaft verändern Sie im Bereich **Properties • Display • Mode** (siehe Abbildung 7.31).

Property	Value	Bindi...
General		
Name	FILTERPANEL_1	
Type	Filter Panel	
Vendor	SAP SE	
Visible	true	+
Data Binding		
Data Source	DS_1	
Target Data Sources		
Display		
CSS Class		+
Mode	Filtering	
Dimensions	Filtering	
Member Display	Filtering and Navigation	

Abbildung 7.31 Eigenschaften des Filter Panels einstellen

7.6.2 Basiskomponenten

Die in Abbildung 7.32 gezeigten Basiskomponenten (Basic Components) umfassen die im Folgenden beschriebenen Designelemente.

Über **Buttons** ① können Sie verschiedene Funktionen mithilfe der integrierten Skriptsprache des SAP Lumira Designers (siehe Abschnitt 7.3, »BI Action Language (BIAL)«) aufrufen. Auch die **Checkbox** ② ermöglicht den Aufruf verschiedener Funktionen, in diesem Fall die Selektion oder Deselektion der Checkbox. Die **Checkbox Group** ③ zeigt mehrere Checkboxes an, jeweils für ein Element. Alle Checkboxes sind vertikal ausgerichtet. Sie ermöglichen es Ihnen, innerhalb der Applikation zu interagieren.



Abbildung 7.32 Basiskomponenten des SAP Lumira Designers

Um z. B. einen Filter einer Datumsdimension zu verändern, können Sie die Komponente *Date Field* verwenden ④. Durch einen Klick auf die Kalenderblätter am rechten Rand der Komponente öffnet sich eine Kalendersicht, in der der Benutzer das gewünschte Datum auswählen kann. Über ein geeignetes Skript können Sie so auf ein bestimmtes Datum hin filtern.

Die *Drop-down-Box* ⑤ können Sie z. B. nutzen, um einen Wert einer Dimension als Filterwert auszuwählen. Hierzu müssen Sie die Drop-down-Box zunächst mithilfe eines Skripts mit den Werten einer Dimension füllen. Zusätzlich benötigen Sie ein Skript, das bei der Selektion eines Wertes ausgewählt wird, um die Datenquelle entsprechend zu filtern.

Um Text darzustellen, gibt es drei Komponenten: *Formatted Text View* ⑥, *Text* ⑦ und *Text Area* ⑧. Bei der Komponente Formatted Text View haben Sie im Unterschied zur Komponente Text die Möglichkeit, in den Additional Properties den anzuzeigenden Text in einem Editor zu formatieren (Abbildung 7.33).

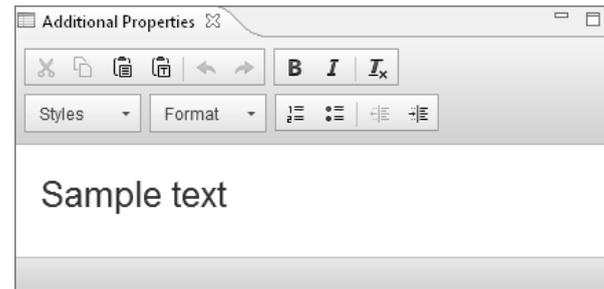


Abbildung 7.33 Formatierter Text im SAP Lumira Designer

In der Komponente Text können Sie die Darstellung des Textes nur über die entsprechende CSS-Klasse oder Standardformate (**Properties • Display • Style**) formatieren. Möchten Sie längere Texte mit Detailformatierungen und Zeilenumbrüchen darstellen, empfiehlt sich die Verwendung des Formatted Text Views. Soll demgegenüber z. B. nur eine Überschrift eingefügt werden, eignet sich hierzu die Komponente Text. Die Komponente *Text Area* ermöglicht es Ihnen, benutzerspezifischen Text während der Laufzeit der Applikation zu schreiben.

Die Komponente *Icon* 7 zeigt Icons an. Die Icons können Sie entweder aus einem Satz von Standard-Icons, die von SAP geliefert werden, oder aus einer benutzerdefinierten Schriftart im TTF- oder OTF-Format auswählen.

Um z. B. ein Unternehmenslogo in Ihre Applikation einzubinden, verwenden Sie die Komponente *Image* 8. Ist die Grafik noch nicht auf Ihrem BI Server abgelegt, können Sie über den Dialog **Open Image** die Grafik auf den Server hochladen und anschließend zur Verwendung auswählen.

Über die Komponente *Input Field* 9 können die Benutzer Ihrer Applikation Text in ein Feld eingeben. Beachten Sie, dass eine Formatierung des eingegebenen Textes nicht möglich ist. Zudem müssen Sie berücksichtigen, dass der eingegebene Text derzeit nicht gespeichert werden kann. Somit ist die Komponente derzeit nur sehr eingeschränkt zu nutzen.

Über die Komponente *Link* 10 können Sie eine URL hinterlegen die Sie aus der Applikation heraus aufrufen möchten.

Listendarstellungen ermöglichen es dem Benutzer, einen Wert, z. B. einer Dimension auszuwählen. Dieser Wert kann anschließend als Filterwert auf eine Datenquelle wirken. Hierzu steht die Komponente *Listbox* 11 zur Verfügung. Ähnlich wie über die Listbox kann auch durch eine *Radiobutton Group* 12 eine Liste von Werten zur Filterung einer Dimension zur Verfügung gestellt werden. Durch die Auswahl des entsprechenden Radiobuttons und die Hinterlegung eines Skripts kann auf den ausgewählten Wert hin gefiltert werden.

Die Komponente *Progress Indicator* 13 zeigt den Fortschritt eines Prozesses grafisch an.

Mit der Komponente *Segmented Button* 14 können Sie mehrere Buttons nebeneinander als Gruppe in Ihre Applikation aufnehmen.

Über die Komponente *Shape* 15 haben Sie die Möglichkeit, aus einer großen Auswahl von Formen und Gebilden zu wählen.

Die Komponente *Switch* 16 ist eine Benutzeroberflächensteuerung, die für den Wechsel zwischen binären Zuständen verwendet wird. Sie können die Taste, die sich auf dem Switch befindet, ziehen oder tippen, um den Zustand zu ändern.

Eine weitere Button-Komponente ist der *Toggle Button* 17, bei dem Sie zwischen gedrücktem und normalem Zustand umschalten können.

Die Komponente *Tree* 18 können Sie an eine Hierarchie einer Dimension anbinden. Wenn die Dimension keine Hierarchie hat, zeigt die Komponente eine einfache Liste an.

7.6.3 Container-Komponenten

Um Formatierungen Ihrer Applikation zu ermöglichen, stehen verschiedene Container-Komponenten zur Verfügung, die wir im Folgenden beschreiben.

Durch die Verwendung des *Grid Layouts* (siehe Abbildung 7.34) können Sie verschiedene Komponenten auf einer Seite darstellen. Die Komponenten werden hierbei jeweils einer Grid-Zelle zugewiesen. Vor allem um bei der Entwicklung einer Applikation die Übersicht zu behalten, ist diese Komponente hilfreich. Zudem können die Komponenten hierdurch exakt angeordnet werden. Das Grid Layout wird häufig zur Anordnung von Buttons oder von mehreren Charts neben- oder übereinander verwendet.

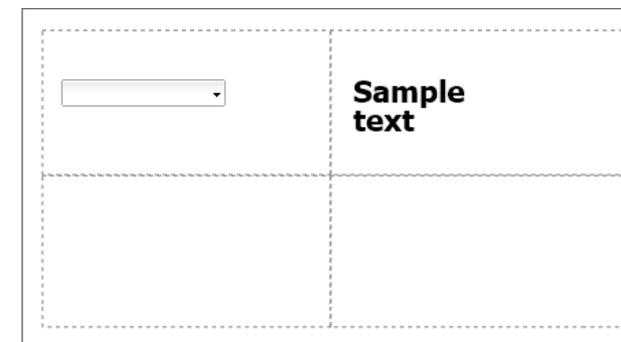


Abbildung 7.34 Grid Layout im SAP Lumira Designer

Das *Pagebook* können Sie verwenden, um mehrere Seiten darzustellen. Diese Darstellungsweise wird häufig auf mobilen Endgeräten verwendet, wie z. B. dem iPad. Hier können Sie mit einem Fingerwischen zwischen den verschiedenen Seiten der Applikation wechseln. Im SAP Lumira Designer erstellen Sie hierzu mehrere Seiten. Diese

werden im Übersichts-View dargestellt. Auf den Seiten können Sie jeweils eine oder mehrere Komponenten ablegen und diese so für den Benutzer zur Verfügung stellen.



Verwendung des Pagebook Layouts

Eine Verwendung des *Pagebook Layouts* in einer Desktop-Applikation empfiehlt sich nicht, da für den Benutzer im Webbrowser nicht zu erkennen ist, dass mehr als eine Seite zur Darstellung zur Verfügung stehen. Für eine Desktop-Applikation sollten Sie eine der anderen Container-Komponenten verwenden.

Ähnlich wie das Pagebook Layout verhält sich auch das *Story Layout*. Nur ist diese Komponente für alle Endgeräte geeignet und kann somit ohne Einschränkungen eingesetzt werden.

Das *Panel Layout* (siehe Abbildung 7.35) können Sie nutzen, um mehrere Komponenten zu gruppieren. Durch die Gruppierung können Sie mehrere Komponenten im Übersichts-View einfacher verschieben. In Abbildung 7.33 werden z. B. ein Dimensionsfilter und eine Chart-Komponente durch das Panel Layout zusammengefasst. Auf diese Weise können Sie die Komponenten innerhalb des Panel Layouts und anschließend das Panel Layout selbst gegenüber anderen Komponenten exakt positionieren.



Abbildung 7.35 Panel Layout im SAP Lumira Designer

Über die *Pop-up-Komponente* können Sie eine zum Pop-up-Fenster hinzugefügte Komponente dynamisch über einen Button ein- bzw. ausblenden. Diese Funktion können Sie für jede Komponente zwar auch erreichen, indem Sie die Eigenschaft

Visible auf true oder false setzen. Um eine Animation beim Ein- bzw. Ausblenden der Komponente zu ermöglichen, müssen Sie jedoch das Pop-up Layout verwenden.

Fügen Sie dazu die Pop-up-Container-Komponente dem Designbereich einer Applikation hinzu, und wählen Sie anschließend die Komponente aus, die in das Pop-up-Fenster eingefügt werden soll (siehe Abbildung 7.36).

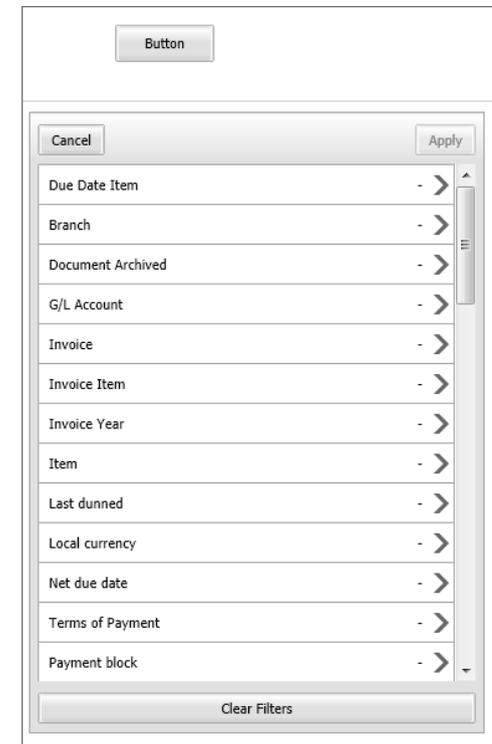


Abbildung 7.36 Pop-up-Fenster im SAP Lumira Designer

Um das Ein- bzw. Ausblenden zu ermöglichen, wird häufig ein Button verwendet, der per Skript die Pop-up-Komponente aktiviert. Die Deaktivierung des Pop-up-Fensters erfolgt zumeist über die Auswahl eines Filters. Die Animation des Pop-ups aktivieren Sie in den Eigenschaften des Pop-ups über **Properties • Display • Animation**. Zur Auswahl stehen die folgenden Animationstypen:

- **No Animation**
Es erfolgt keine Animation.
- **Fade Animation**
Die Komponente wird langsam eingeblendet bzw. abgeblendet.
- **Flip Animation**
Die Komponente wird zunächst gespiegelt eingeblendet und dreht sich dann herum.

■ Pop Animation

Die Komponente erscheint unmittelbar in der Applikation.

■ Horizontal Slide Animation

Die Komponente wird von der linken oder rechten Bildseite aus eingeblendet.

■ Vertical Slide Animation

Die Komponente wird von der unteren oder oberen Bildseite aus eingeblendet.

Mithilfe der Pop-up-Komponente können Sie auch in Desktop-Applikationen Animationen erzeugen, die sonst nur für mobile Endgeräte zur Verfügung stehen. Häufig kommt das Pop-up-Fenster zum Einsatz, um das Filter Panel ein- bzw. auszublenden oder um die Auswahl von Werten über Checkboxes oder Listboxen zu ermöglichen.

Die Container-Komponente, das *Tabstrip Layout* (siehe Abbildung 7.37), können Sie sowohl in Desktop-Applikationen als auch in Applikationen für mobile Endgeräte verwenden, um mehrere Sichten in einer Applikation darzustellen.

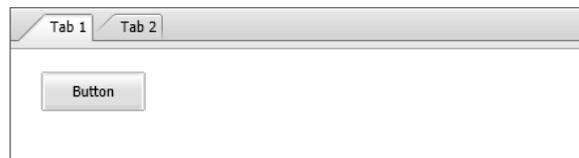


Abbildung 7.37 Tabstrip Layout im SAP Lumira Designer

Das Tabstrip Layout stellt hierzu Registerkarten (sogenannte *Tabs*) zur Verfügung, auf denen die Komponenten einer Sicht abgelegt werden. Die einzelnen Registerkarten können Sie mit einem Titel versehen. Durch einen Klick auf die jeweilige Registerkarte wird die entsprechende Sicht angezeigt. Hierdurch lassen sich auch komplexe Applikationen entwickeln, die mehrere Sichten zur Verfügung stellen. Sie können z. B. ein Übersichts-Chart auf der ersten Registerkarte, mehrere Detail-Charts auf der zweiten Registerkarte und eine Analysetabelle auf der dritten Registerkarte anzeigen.

Die Dialogsteuerung (**Dialog**) können Sie verwenden, um die aktuelle Bearbeitung einer Applikation zu unterbrechen und den Benutzer zur Information oder zur Antwort zu veranlassen.

Das *Pop-up Layout* ermöglicht es Ihnen, schnell Informationen einzugeben, Einstellungen zu konfigurieren oder eine Auswahl zu treffen.

Wenn Sie eine Applikation erstellen wollen, die auf jedem Endgerät und mit jeder Auflösung funktioniert, sollten Sie möglichst die Komponente *Adaptive Layout* nutzen. Das Adaptive Layout arrangiert Blöcke von Komponenten in ein Raster das sich immer den jeweiligen Bildschirmgrößen anpasst. In Abbildung 7.38 sehen Sie ein Beispiel für die Übersicht (**Outline**) eines Adaptive Layouts. Wie Sie es hier sehen, wurde jedem Block eine Grafik zugeordnet.

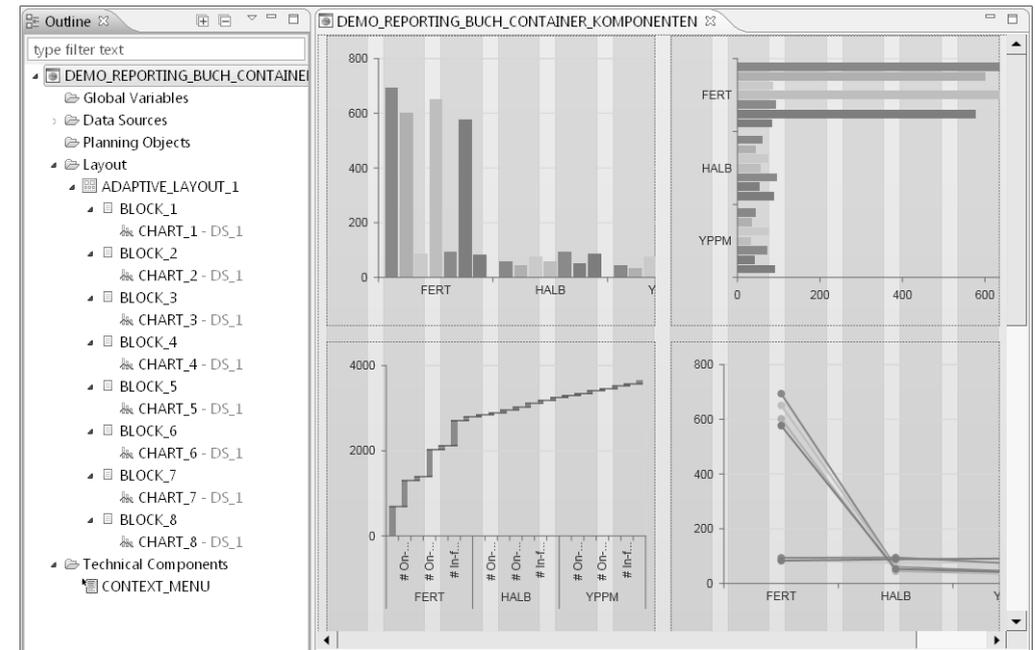


Abbildung 7.38 Adaptive Layout – Outline

Wenn Sie die Applikation auf dem Desktop ausführen, erhalten Sie – wie es in Abbildung 7.39 zu sehen ist – eine Applikation mit zwei Reihen und jeweils vier Blöcken.

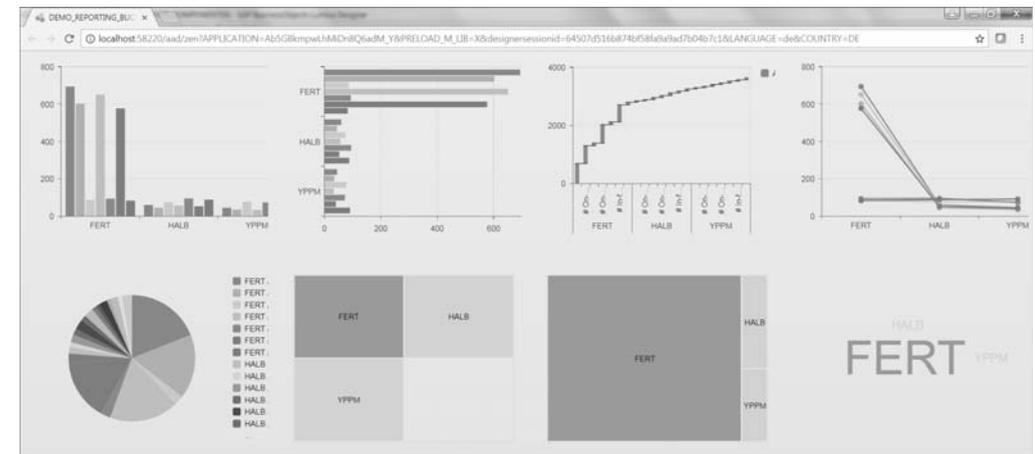


Abbildung 7.39 Adaptive Layout – Desktop

Wenn Sie dieselbe Applikation auf einem Tablet ausführen, erhalten Sie, wie es in Abbildung 7.40 zu sehen ist, vier Reihen mit jeweils zwei Blöcken; der dritte und vierte Block rücken automatisch in die zweite Reihe.

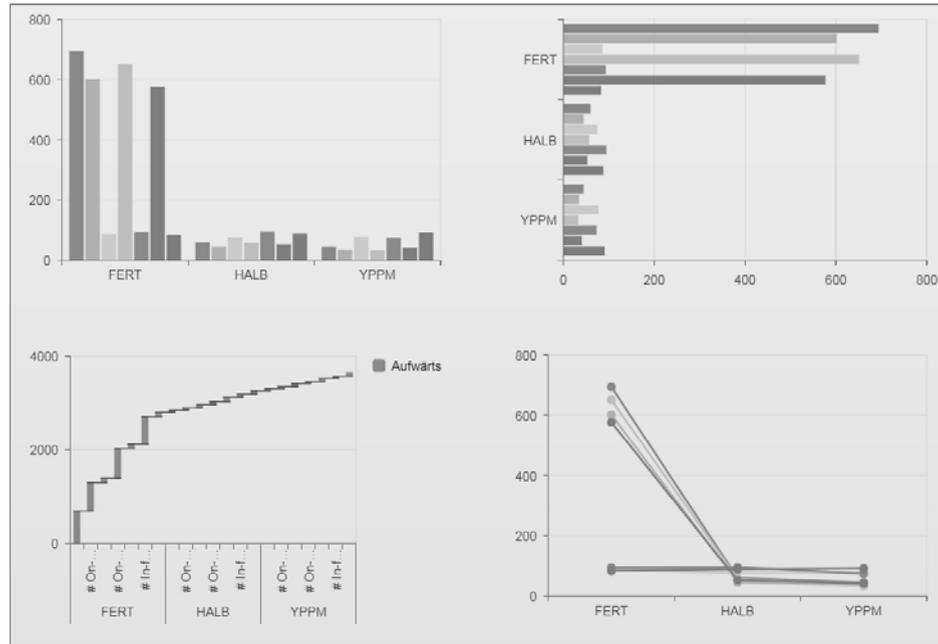


Abbildung 7.40 Adaptive Layout – Tablet

Auch auf dem Smartphone lässt sich diese Applikation ausführen. Dort sehen Sie dann acht Reihen mit jeweils einem Block (siehe Abbildung 7.41).

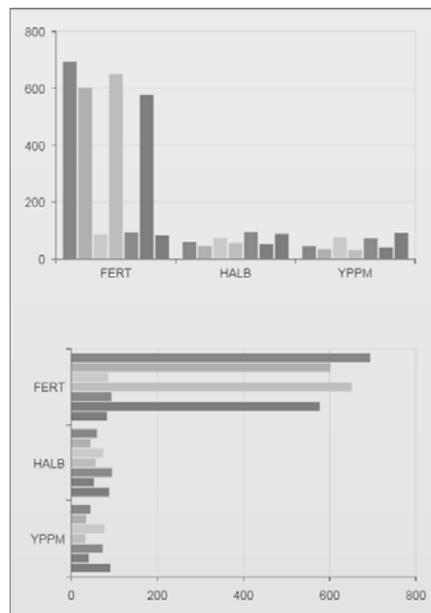


Abbildung 7.41 Adaptive Layout – Smartphone

Die Adaptive-Layout-Komponente unterstützt somit verschiedene Layouts für sämtliche Bildschirmgrößen. Sie können Ihre Applikation auf dem Smartphone genauso nutzen wie auf einem großen Bildschirm.

7.7 Integration mit SAP BusinessObjects Analysis for Microsoft Office

Der SAP Lumira Designer bietet sehr gute Integrationsmöglichkeiten mit SAP BusinessObjects Analysis for Microsoft Office. Sie können Ihre Analysis-Berichte mit nur wenigen Klicks in SAP-Lumira-Designer-Applikationen verwandeln.

7.7.1 Integration mit SAP BusinessObjects Analysis for Microsoft Office

Mit SAP BusinessObjects Analysis for Microsoft Office haben Sie die Möglichkeit, Ihre Analysen in den SAP Lumira Designer zu übertragen. Sie können dazu **Smart Copy** und **Smart Paste** nutzen. Damit können Sie die DataSource aus SAP BusinessObjects Analysis for Microsoft Office kopieren und in den SAP Lumira Designer einfügen. Dabei werden u. a. die folgenden Eigenschaften der DataSource übertragen:

- bedingte Formatierungen, die Sie in SAP BusinessObjects Analysis for Microsoft Office erstellt haben
- Berechnungen, die Sie in SAP BusinessObjects Analysis for Microsoft Office erstellt haben
- Skalierungen und die Anzahl der Dezimalstellen von Kennzahlen

Um eine DataSource zu kopieren, öffnen Sie das Kontextmenü z. B. einer Tabelle in SAP BusinessObjects Analysis for Microsoft Office und wählen den Eintrag **Smart Copy** (View kopieren). Öffnen Sie anschließend im SAP Lumira Designer das Kontextmenü zum Eintrag **Data Sources** im Übersichts-View, und wählen Sie den Eintrag **Smart Paste** aus (siehe Abbildung 7.42).

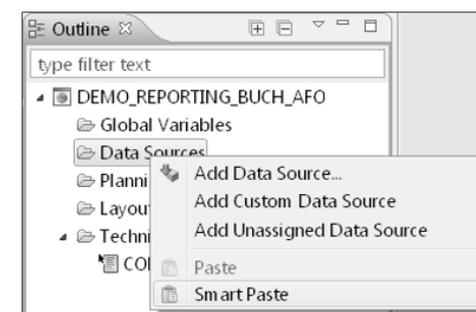


Abbildung 7.42 Smart Paste

7.8 Software Development Kit (SDK)

Mit dem SAP Lumira Designer wird auch ein *Software Development Kit (SDK)* zur Entwicklung eigener Komponenten ausgeliefert, mit dem Sie fehlende Komponenten selbst entwickeln können. Beispiele könnten sein:

- Kartendarstellungen
- Chart-Typen, die derzeit noch nicht standardmäßig ausgeliefert werden
- Kommentarfunktionen, um dem Benutzer die Eingabe bzw. Anzeige von Kommentaren zu ermöglichen

Da der SAP Lumira Designer auf Eclipse basiert, nutzen Sie für die Programmierung XML und JavaScript. Sie können zwar beliebige XML- und JavaScript-Editoren verwenden, da Eclipse aber weit verbreitet ist, wird die Verwendung dieses Tools empfohlen. Wenn Sie Eclipse für die Entwicklung nutzen, können Sie direkt aus Eclipse heraus auf den SAP Lumira Designer zugreifen. Die von Ihnen entwickelte Komponente ist auf diese Weise automatisch im SAP Lumira Designer verfügbar. Die folgenden Voraussetzungen gelten für die Entwicklung neuer Komponenten:

- Sie sollten ein Grundverständnis der SAP-Lumira-Designer-Konzepte haben.
- Sie sollten über gute HTML- und JavaScript-Kenntnisse verfügen.
- Kenntnisse im CSS- und jQuery-JavaScript-Framework sind vorteilhaft, da diese dem SAP Lumira Designer zugrunde liegen.
- Wenn Sie Eclipse als Entwicklungsumgebung verwenden möchten, benötigen Sie natürlich auch eine Eclipse-Installation.



Hilfreiche Links zum Start in die Programmierung neuer Komponenten

Unter den folgenden URLs finden Sie hilfreiche Informationen zur SAP-Lumira-Designer-Erweiterung:

- Eclipse-Umgebung: <http://www.eclipse.org>
- Beispiele zur Entwicklung mit dem SAP Lumira Designer SDK:
 - <http://s-prs.de/v563927>
 - <http://s-prs.de/v563928>
- JavaScript- und HTML-Grundlagen:
 - <http://de.selfhtml.org>
- D3-Tutorials zur Programmierung:
 - <http://d3js.org>
 - <http://s-prs.de/v563929>
- Informationen zur Google Maps API
 - <http://s-prs.de/v563930>

Die Einrichtung der Eclipse-Umgebung wird im SAP BusinessObjects Lumira Designer Developer Guide ausführlich beschrieben; daher verzichten wir hier auf eine Darstellung. Sie finden den Developer Guide unter <http://s-prs.de/v563931>.

Neben der Möglichkeit, für Ihr eigenes Unternehmen selbst Komponenten zu entwickeln, können Sie mit dem SAP Lumira Designer SDK auch als Drittanbieter oder SAP-Partner Komponenten entwickeln und diese im SAP Store zur Verfügung stellen. Einige Komponenten sind bereits über den SAP Store erhältlich und können im SAP Lumira Designer eingebunden werden.

7.9 Performanceoptimierung

Es gibt viele Wege, um die Performance einer SAP-Lumira-Designer-Applikation zu optimieren. Im Folgenden zeigen wir Ihnen, wie Sie die Laufzeit ihrer Applikationen verkürzen können.

7.9.1 Background Processing

Das Background-Processing-Ereignis ermöglicht es Ihnen, DataSources im Hintergrund zu laden. Auf diese Weise können Sie die initiale Ladezeit der Applikation verkürzen. Die Gesamtladezeit der Applikation bleibt dabei konstant. Der entscheidende Vorteil ist, dass das Dashboard schon angezeigt wird, während zusätzliche DataSources im Hintergrund geladen werden. Dadurch entsteht der Eindruck, dass die Applikation schneller lädt. Nehmen wir als Beispiel eine einfache Applikation, die drei DataSources enthält, die in drei verschiedenen Tabellen angezeigt werden, wie es in Abbildung 7.43 dargestellt ist.

Measurestructure	Act cum CY	Year	Act cum CY	Period	Act cum CY
Heads	236,887	2012	27,615,863	1	3,809,085
FTE	223,170	2011	27,298,439	2	2,539,388
Sales	13,336,019	Total	54,914,302	3	3,174,236
Net Sales	11,751,389			4	5,078,780
Personnel Costs	8,369,060			5	5,078,782
				6	5,078,780
				7	3,174,236
				8	2,539,388
				9	3,809,085
				10	6,031,050
				11	7,300,746
				12	7,300,746
				Total	54,914,302

Abbildung 7.43 Applikation ohne Background Processing

Wenn jede der drei DataSources allein fünf Sekunden benötigt, um initial geladen zu werden, müssen Sie beim Starten der Applikation 15 Sekunden warten, bis das Dashboard angezeigt werden kann. Mithilfe des Background Processings können Sie das Dashboard sofort starten lassen, und erst dann werden die DataSources nacheinander geladen. Fünf Sekunden nach dem Start wäre die erste DataSource geladen (siehe Abbildung 7.44), nach weiteren fünf Sekunden die nächste und wiederum fünf Sekunden später die dritte DataSource. Sie sehen also Teile des Dashboards bereits nach fünf Sekunden, anstatt 15 Sekunden auf die gesamte Applikation warten zu müssen.

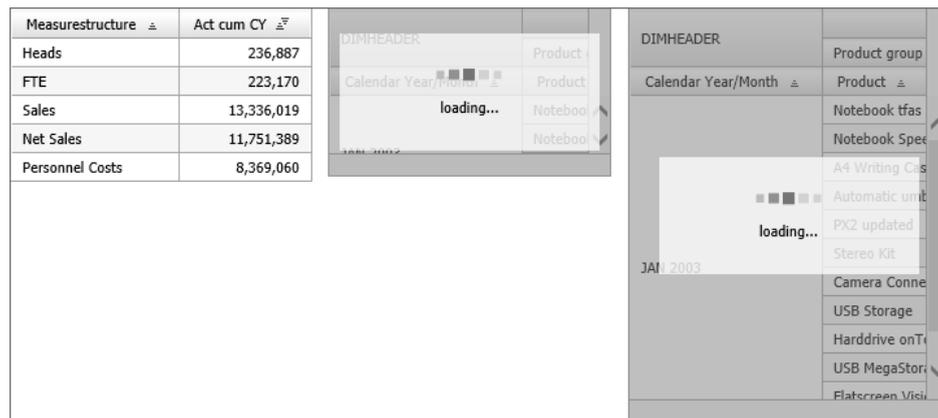


Abbildung 7.44 Applikation mit Background Processing

Um das Background Processing einzustellen, gehen Sie wie folgt vor:

1. Klicken Sie mit der linken Maustaste auf Ihre DataSource, und setzen Sie in den Eigenschaften die Eigenschaft **Load in Script** auf **false**.
2. Erstellen Sie eine globale Variable, die im Skript als Zähler verwendet wird, um durch die DataSources zu iterieren. Wechseln Sie dafür in die Applikationseigenschaften, klicken Sie dann auf **Global Script Variables**, und erstellen Sie eine Variable mit den Eigenschaften, die Sie in Abbildung 7.45 sehen.

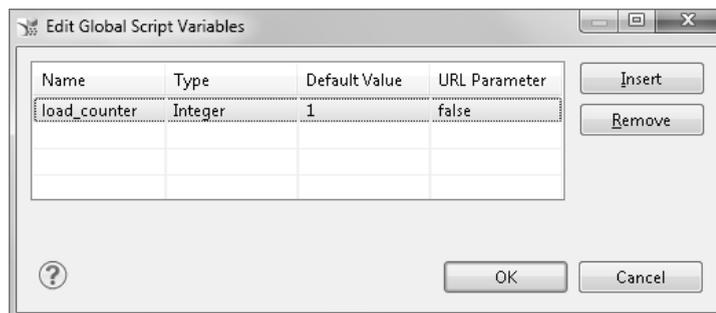


Abbildung 7.45 Globale Variable als Zähler

3. Hinterlegen Sie in den Applikationseigenschaften für das Ereignis **On Startup** den Code aus Listing 7.2, um den Ladestatus der Tabelle anzuzeigen. Mit der letzten Zeile rufen Sie die Prozedur auf, die in das Background-Processing-Ereignis eingestellt wird:

```
CROSSTAB_1.showLoadingState("loading...");
CROSSTAB_2.showLoadingState("loading...");
CROSSTAB_3.showLoadingState("loading...");
APPLICATION.doBackgroundProcessing();
```

Listing 7.2 Skript zur Anzeige des Ladestatus in Tabellen

4. Als Letztes fügen Sie den Code aus Listing 7.3 in die Applikationseigenschaften für das Ereignis **On Background Processing** ein. Dieser Code wird rekursiv ausgeführt und ist dafür verantwortlich, die DataSources nacheinander zu laden. Gleichzeitig wird der Ladestatus der entsprechenden Komponenten ausgeschaltet.

```
if (load_counter == 1) {
  CROSSTAB_1.hideLoadingState();
  DS_1.loadDataSource();
}
else if (load_counter == 2) {
  CROSSTAB_2.hideLoadingState();
  DS_2.loadDataSource();
}
else if (load_counter == 3){
  CROSSTAB_3.hideLoadingState();
  DS_3.loadDataSource();
}
if (load_counter < 3){
  load_counter = load_counter + 1;
  APPLICATION.doBackgroundProcessing();
}
```

Listing 7.3 Skript zur Ausführung des Ladeprozesses einer DataSource

7.9.2 Anzahl der DataSources reduzieren

Vermeiden Sie es, wenn möglich, verschiedene DataSources für dieselbe Datenquelle zu erstellen. Verwenden Sie stattdessen Methoden (z. B. `moveDimensionToRows`), um den Aufriss einer Datenquelle so zu verändern, dass diese für verschiedene Komponenten verwendet werden kann. Dies funktioniert jedoch nur, wenn die Komponenten, die dieselbe DataSource verwenden, nicht gleichzeitig in der SAP-Lumira-Designer-Applikation angezeigt werden.

Wenn Sie verschiedene Diagramme verwenden, die auf der Basis mehrerer DataSources auf dieselbe Datenquelle verweisen, überprüfen Sie, ob es möglich ist, DataSources mithilfe der Eigenschaft **Data Selection** zu eliminieren. Die Eigenschaft **Data Selection** kann nur für Diagramme verwendet werden und erlaubt es Ihnen, nur einen Ausschnitt der DataSource in dem Diagramm anzuzeigen.

7.9.3 Nutzung von BEx-Variablen vermeiden

Wenn Sie BW Queries als Datenquellen in Ihren SAP-Lumira-Designer-Applikationen verwenden, vermeiden Sie es, wenn möglich, BEx-Variablen für das Filtern zu nutzen. Verwenden Sie stattdessen die Methoden `setFilter` und `setFilterExt`. Dies funktioniert für Standardvariablen, die nach einem Muster arbeiten, wie z. B. einzelne Werte, mehrere einzelne Werte oder Bereiche. Bei anderen Variablenarten ist ein Verzicht nicht möglich, wie z. B. bei Datumsvariablen mit einem Offset.

7.9.4 Profiling

Wenn Sie ein komplexes Dashboard erstellen, ist es später schwierig, genau zu erörtern, wo der größte Performanceengpass liegt. Mit dem sogenannten *Profiling* können Sie die Performance Ihrer Applikation genau analysieren. Um das Profiling zu aktivieren, gehen Sie wie folgt vor:

1. Starten Sie Ihre Applikation im Local Mode mit einem Klick auf **Application • Execute Locally**.
2. Kopieren Sie die URL, die in der Adressleiste Ihres Internetbrowsers erscheint.
3. Fügen Sie an das Ende der URL den Parameter `&PROFILING=` ein. Die URL müsste dann wie folgt aussehen: `http://localhost:5555/aad/web.do?APPLICATION=<APPLICATION_ID>&designersessionid=<DESIGNERSESSIONID>d6dd55&PROFILING=X`
4. Es öffnet sich das Fenster **Statistics** (siehe Abbildung 7.46).

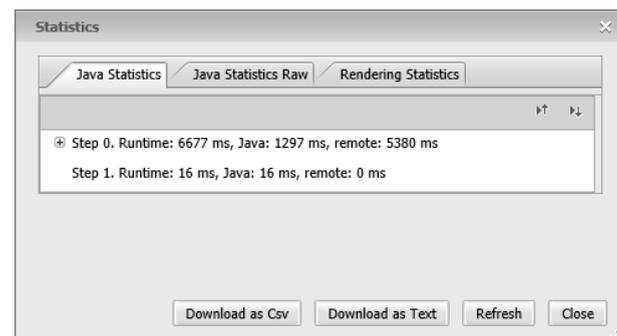


Abbildung 7.46 Profiling-Fenster

Das Profiling-Fenster ist in drei Registerkarten unterteilt:

- **Java Statistics**
Hier wird die Laufzeit der Applikation in Bezug auf die Datenquellen analysiert (Backend). Die Ausgabe erscheint optimiert in einem für Menschen lesbaren Format.
- **Java Statistics Raw**
Hier wird die Laufzeit ebenfalls in Bezug auf die Datenquellen (Backend) analysiert. Die Ausgabe erscheint in einem maschinenlesbaren Format.
- **Rendering Statistics**
Hier wird die Laufzeit in Bezug auf die Frontend-Komponenten analysiert.

Wie in Abbildung 7.47 werden die durch das Profiling gewonnenen statistischen Ergebnisse so angezeigt (beispielsweise auf der Registerkarte **Java Statistics**), dass die Funktion `CREATE_DATA_PROVIDER` fünfmal ausgeführt wird (da die Applikation fünf DataSources enthält). Sie sehen auch, dass eine der fünf DataSources wesentlich mehr Zeit in Anspruch nimmt als die anderen. Aufgrund dieser Erkenntnisse können Sie sich bei der Performanceoptimierung auf bestimmte DataSources konzentrieren.

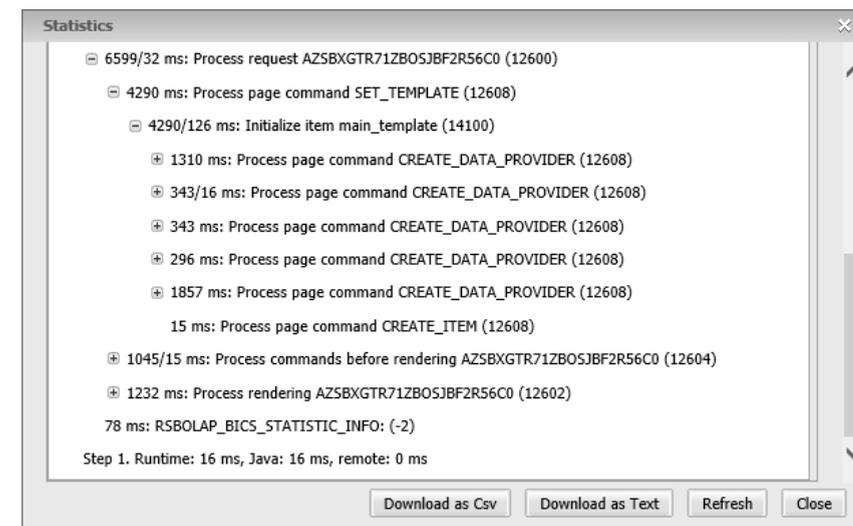


Abbildung 7.47 Statistiken aus dem Profiling

7.9.5 CSS-Klassen nutzen

Wenn Sie dynamisch verschiedene Objekte ein- und ausblenden möchten, deren Unterschied sich auf das Aussehen der Komponente beschränkt, können Sie versuchen, die Anzahl der Komponenten zu verringern, indem Sie dynamisch CSS-Klassen zuweisen. Das Verwenden einer CSS-Klasse ist insgesamt performanter als die Verwendung verschiedener Objekte (siehe Abschnitt 7.5, »Cascading Style Sheets (CSS)«).

7.9.6 Variablen anstelle von Textelementen verwenden

Wenn Sie Werte zwischenspeichern müssen, um mit diesen Werten zu einem späteren Zeitpunkt eine Berechnung durchzuführen, vermeiden Sie es, diese Werte in versteckten Textobjekten zu speichern. Nutzen Sie stattdessen globale Variablen, um Werte zwischenzuspeichern. Das Speichern und Auslesen von Werten in Textelementen ist nicht performant, da dafür zwischen Text und Zahl konvertiert werden muss. Um zwischen Datentypen zu konvertieren, nutzen Sie das Objekt **Convert**, gefolgt von der jeweiligen Methode (z. B. `Convert.stringToInt("1");`).