SAP NetWeaver: Customer Conference Call

SAP NetWeaver® Product Management
Welcome and thanks for joining our call!

Events update

Topic of the Day:

“Web Services within SAP NetWeaver 2004s BI - Create, Discover, and Consume!”

Prakash Darji

SAP NetWeaver RIG, SAP Labs, LLC

Questions

Customer issues/feedback
SAP and ASUG once again team up to enhance the SAP TechEd attendee experience.

SAP TechEd ’06 ▪ September 12 - 15
The Venetian Congress Center ▪ Las Vegas

Register today at www.sapteched.com/usa
SAP TechEd and ASUG: Partners in Education

ASUG Education
- Select from over 100 hours of focused educational content driven by SAP customers.
- Gain practical and technical know-how, real-world tips and techniques from fellow SAP users.

ASUG Influence
- Shape the future of SAP product development and provide input on existing SAP solutions.

ASUG Networking
- Join ASUG in the SDN Clubhouse to exchange ideas and best practices with fellow users and discuss hot topics affecting the industry.

Visit www.sapteched.com/usa to register today!
### Introduction

Overview of Web Services and Standards

SAP BI Web Service for Queries

Creating BI Web Services for BAPIs

Consuming BI Web Services within Applications

Consuming Other Web Services within Applic.

Service Oriented Architecture Scenarios

Summary
Introduction

Overview of Web Services and Standards
SAP BI Web Service for Queries
Creating BI Web Services for BAPIs
Consuming BI Web Services within Applications
Consuming Other Web Services within Applic.
Service Oriented Architecture Scenarios
Summary
Learning Objectives

You will learn about web services within SAP NetWeaver 2004s BI

The following questions will be answered:

- What are web services?
- How do you create web services?
- How do you consume web services?
- Why you should use web services?
SAP Business Explorer Suite

SAP NetWeaver Portal

SAP Business Explorer Suite (BEx)

BEx Broadcaster

BEx Web

BEx Web Analyzer

BEx Web Application Designer

BEx Report Designer

BI Pattern

BEx Analyzer

MS Excel Add-in

BI Consumer Services

3rd Party BI

SAP NetWeaver BI Info Provider

BEx Query Designer

BI Pattern

SAP NetWeaver Portal

SAP Business Explorer Suite (BEx)

BEx Broadcaster

BEx Web

BEx Web Analyzer

BEx Web Application Designer

BEx Report Designer

BI Consumer Services

3rd Party BI

SAP NetWeaver BI Info Provider

BEx Query Designer

BI Pattern
Introduction

Overview of Web Services and Standards

SAP BI Web Service for Queries
Creating BI Web Services for BAPIs
Consuming BI Web Services within Applications
Consuming Other Web Services within Applic.
Service Oriented Architecture Scenarios
Summary
Web Services Definition

What are Web services?

Web services are

*Self-contained, self-describing, modular application functionalities*

*Once deployed, applications and other Web services can be discovered and directly invoked via service requests based on open standards for system interaction, independent of technical architecture*
Web Services and SAP BI – Why Should You Care?

What do Web services do for SAP BI?

Simply put, Web services enable SAP BI to expand its footprint and impact on your organization.

Areas where benefits occur:

1. Extending use of SAP BI’s integrated data beyond traditional users
   - Leveraging existing developments and your data
   - Integrating your data warehouse into your application infrastructure
   - Increasing the value of your data warehouse deployments
   - ...

2. Provision of SAP BI features to other components and applications
   - Facilitating easier access to data
   - Enriching the developer’s toolkit
   - Providing flexibility to developers
   - ...

Web Services and SAP BI – Why Should You Care?

Example: Web services can help address these SAP BI issues:

- I know my SAP BI data has great value to my organization. How can I open the vault and expose the wealth of my SAP BI data to other users and applications?
- I want to be able to supply my SAP BI data on multiple levels:
  - Provide analytical access to applications and users
  - Provide query results to applications and users
  - How can I do it in a way that is open and pervasive (everybody talks XML!)?
- The data is there, I can see it – can I get it into my data warehouse?
  - I need to readily feed my SAP BI with data that is already flowing through my organization
- Finally, my IT budget is finite, how can I do this inexpensively and in a future-looking way?
  - I want to leverage my existing SAP BI investments
  - I cannot introduce multiple tools/packages to deliver this

✓ Leveraging existing developments/capabilities
✓ Lowering TCO
✓ Standards-driven
Service-Oriented Architecture (SOA) Definition

What is Service-Oriented Architecture?

Service-Oriented Architecture is

an application architecture in which all functions (services) are defined using a description language and have invokable interfaces that are called to perform business processes. Each interaction is independent of each and every other interaction.

* Source: webopedia.com
Enterprise Services and Composite Application Definitions

What are enterprise services and composite applications?

**Enterprise services are**

*Web services that provide enterprise-level business functionality providing enterprise quality in scalability, robustness, security, manageability, and supportability are fully described via the Web services model*

**Composite applications are**

*Flexible business scenarios or processes delivered by composing enterprise services. Can then be surfaced as enterprise services in their own right!*
<table>
<thead>
<tr>
<th>Introduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview of Web Services and Standards</td>
</tr>
<tr>
<td><strong>SAP BI Web Service for Queries</strong></td>
</tr>
<tr>
<td>Creating BI Web Services for BAPIs</td>
</tr>
<tr>
<td>Consuming BI Web Services within Applications</td>
</tr>
<tr>
<td>Consuming Other Web Services within Applic.</td>
</tr>
<tr>
<td>Service Oriented Architecture Scenarios</td>
</tr>
<tr>
<td>Summary</td>
</tr>
</tbody>
</table>
Options for Consuming BI Data with Web Services

Within BI, you can use web services to load BI data or consume BI Data. In this discussion, we are going to discuss consuming BI Data.

There are 2 primary web services for consuming BI Data:

1. XML for Analysis (XMLA)
2. XML Query Result Set (query_view_data)
XML for Analysis (XMLA)

- Connection to third-party, front-end tools
- Development of customer-specific front-end tool, which passes XML-packaged Multi-Dimensional eXpressions (MDX) statements on to SAP BI and requests XML-formatted results
- Data interchange with other subsidiaries/companies/dashboards/etc.

Implementation

- An ICF service is delivered pre-configured
- 1:n Web Service to query ratio
  - One service for all queries!
- MDX necessary for query processing for navigation step
- Web service uses existing SAP BI authorizations of specified user
SAP BI Open Analysis Interfaces

**OLE DB for OLAP**
- Based on COM as protocol
- Driver installation on clients
- Windows platform only
- MDX as query language
- Available since BW 1.2B

**OLAP BAPIs**
- Business Application Programming interface
- Based on RFC as protocol
- Access library available on any SAP platform
- MDX as query language
- Available since BW 2.0A

**XML for Analysis**
- Based on XML as exchange format
- Based on HTTP/SOAP as protocol
- Any platform
- MDX as query language
- Available since BW 3.0A
XMLA is a SOAP-based interface for exposing OLAP and data mining data sources as Web services.

It advances some of the successful concepts of OLE DB for OLAP to a cross-platform Web service API.

The XMLA API defines two methods:

- **Discover**: Used to obtain metadata and information about a Web service.
- **Execute**: Used to execute (M)ulti(D)imensional E(X)pressions (MDX) or provider-specific commands.

Benefits:

- Broadly applicable and accepted low-footprint OLAP API.
Beyond E-commerce, there is an even greater need to share information. Add value to your offering by providing self-service access.

**Just-in-Time Inventory**
- Consumer provides to a Supplier just-in-time inventory data reports exposed as Web services.
- Metrics represented in the reports alert about low inventory.
- The reports are interactive, so you drill into historic data, comparing today's fluctuation in stock levels with corresponding days in other weeks or months.

**Market/Environment data**
- Nielsen and others offer their services in the yellow pages.
- Integration and consumption of such data becomes much easier without requiring a very close business relationship with such service providers.

**An electric utility company**
- Exposes its BI platform, so commercial and industrial customers can better understand their power consumption and related costs.

**A financial services company**
- Provides BI Web services, so auditors and regulators have an appropriate level of visibility into the company's dealings.

**A brokerage firm**
- Supports BI Web services so institutional investors can analyze (in a multidimensional way) the performance of financial instruments executed through the brokerage.
XMLA Architecture

Any client (Windows, Linux, Macintosh) at any location

Any server (Solaris, Linux, Windows) at any location
1.1 XMLA Setup and Configuration

Go to transaction SICF and choose execute to display all services
1.2 XMLA Setup and Configuration

Navigate to default_host/sap/bw/xml/soap/xmla and activate and test the service
1.3 XMLA Setup and Configuration

Logon to the WebAS and ensure that the test succeeded (you should see the WSDL file and have the path for this now!)

```xml
<?xml version="1.0" encoding="utf-8" ?>
    xmlns:mime="http://schemas.xmlsoap.org/soap/mime/">
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assignNames>
    <http:assignNames>
        <http:assignName/>
    </http:assign Names>
XML Query Result Set - Motivation

XML query result set

- Simple access to query results as XML list output
- Development of customer-specific, front-end application reading SAP BI data
- Data interchange with other subsidiaries/companies/dashboards/etc.

Implementation

- An ICF service must be created
- 1:n Web Service to query ratio
  - One service for all queries!
- No additional MDX necessary (only XMLA requires this), therefore no slicing-and-dicing navigation directly possible without Web API
  - Simple input parameters: InfoProvider, Query or Query View, filters, setting a navigation status, etc.
  - Support for WEB API commands available. Variables, Variants, or Filter Values would be specified via WEB API commands.
- Web Service uses existing SAP BI authorizations of specified user
Go to SICF and activate this web service: 
“/sap/bc/srt/rfc/sap/QUERY_VIEW_DATA”
Go to transaction WSCONFIG and enter service destination “QUERY_VIEW_DATA” and enter variant “query_view_data” and hit enter. Currently, there are no released services so create a new service.
Choose Save to release this web service.
2.4 XML Query Result Set – Setup and Configuration

Choose ICF Details -> External Aliases

Release Web Services for SOAP Runtime

Web Service Definition
- Name: query_view_data
- SOAP Application: urn:soap-con:soap:runtime:application:rfc
- Security: Authentication Level: Basic
- Security: Transport Guarantee Level: None

Web Service Settings
- Release Text: Web Service QUERY_VIEW_DATA
- Call Details:
  - Virtual Host: default_host
  - URL: /sap/bd/srf/rfc/sap/QUERY_VIEW_DATA

Maintain service
- External Aliases
2.5 XML Query Result Set – Setup and Configuration

Choose “New” to create a new external alias.

Enter external alias “/sap/bw/xml/soap/queryview” and enter description “SAP BI Web Service – XML Result Set”
Specify your security and user ID on the “Logon data” tab.
Specify "/sap/bc/srt/rfc/sap/QUERY_VIEW_DATA" under the "Trg Element" tab.
2.8 XML Query Result Set – Setup and Configuration

Save your external alias.

Maintain external aliases

<table>
<thead>
<tr>
<th>External Alias</th>
<th>/sap/bw/xmlsoap/queryview</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trg Element:</td>
<td></td>
</tr>
</tbody>
</table>
2.9 XML Query Result Set – Setup and Configuration

Go to transaction WSADMIN

Go to -> Administration Settings

Ensure that the path to your J2EE is specified here and save this.
2.10 XML Query Result Set – Setup and Configuration

Find your web service and choose the “Web service homepage”
Login to your ABAP system from the Web Service Navigator

Authorization

The selected endpoint requires basic authentication. Please, enter correct username and password:

- Username: i805866
- Password: ••••••••

Submit
2.12 XML Query Result Set – Setup and Configuration

Choose Test -> GetQueryViewData to test this service.
2.13 XML Query Result Set – Setup and Configuration

Enter and Infoprovieder and Query and send the request to test this web service.

Note: Variables aren’t possible with this web service, so make sure the query you use a query without variables for this test.
## 2.14 XML Query Result Set – Setup and Configuration

Ensure the request and response are successful.

### Request
- `GetQueryViewData`
- Parameters:
  - `Infopreview (String)`: `00_DX_M01`
  - `Parameter (test.types.p1.W3query[])`: `SKIP`
  - `Query (String)`: `00_DX_M01_Q0006`
  - `Viewid (String)`: 

### Response
- `GetQueryViewData`
- Response:
  - `AxisData (test.types.p1.RrwsSxAxisData[])`
  - `Axis (String)`: `000`
  - `Set (test.types.p1.RrwsSxTuple[])`
    - `TupleOrdinal (String)`: `00000`
    - `Chenm (String)`: `93P6J6G03ES2L9ULB`
    - `Caption (String)`: `Net Sales`
    - `Chav (String)`: `5TWOEM1H5G7RGE`
    - `ChavExt (String)`: 
    - `Nobjnm (String)`: 
    - `Tlevel (String)`: `00`
    - `Drillstate (String)`: 
If you have issues, make sure you activate your Web Service and Interface in SE80
Here is an example of how to pass parameters for filter values using WEB API commands. We are passing the FILTER_IOBJNM=0D_CHANNEL and FILTER_VALUE=3.
Here is an example of how to pass parameters for variable values using WEB API commands. In this case, we are using a selection option variable. We are passing the following parameters:

VAR_NAME_1=DISTGOP
VAR_OPERATOR_1=EQ
VAR_VALUE_LOW_EXT_1=3
VAR_VALUE_HIGH_EXT_1=5
## Comparing BI Web Services

### BI Web Service Comparison

<table>
<thead>
<tr>
<th></th>
<th>XMLA</th>
<th>QUERY_VIEW_DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Configuration</strong></td>
<td>An ICF service is delivered pre-configured</td>
<td>An ICF Service must be created</td>
</tr>
<tr>
<td><strong>Maintenance</strong></td>
<td>1:n Ratio – 1 web service for all queries</td>
<td>1:n Ratio – 1 web service for all queries</td>
</tr>
<tr>
<td><strong>Coding</strong></td>
<td>MDX Necessary for query processing</td>
<td>No additional MDX necessary</td>
</tr>
<tr>
<td><strong>Navigation</strong></td>
<td>Query Navigation Steps Possible via MDX</td>
<td>Query Navigation Steps Possible via Web API</td>
</tr>
<tr>
<td><strong>Authorizations</strong></td>
<td>Uses existing SAP BI Authorizations</td>
<td>Uses existing SAP BI Authorizations</td>
</tr>
<tr>
<td><strong>Standards</strong></td>
<td>Microsoft’s XMLA standard</td>
<td>SAP Proprietary Format</td>
</tr>
<tr>
<td>Introduction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overview of Web Services and Standards</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAP BI Web Service for Queries</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Creating BI Web Services for BAPIs</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Consuming BI Web Services within Applications</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Consuming Other Web Services within Applic.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service Oriented Architecture Scenarios</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
You can create BAPI’s to interact with your BI System.

Common BAPI’s you may want to use:

1. BAPI to write to transactional cube
2. BAPI to read and write BI documents on master data and transactional data
3. BAPI to pull a report list
Creating Web Services for BAPI’s

Wrapping BAPI’s in web services allow greater extensibility and flexibility

Procedure:

1. Create or identify Function Group
2. Create RFC Enabled Function Module in ABAP System in your function group
3. Create Web Service for the Function Group
4. Test the Web Service
1. Create or Identify Function Groups

Go to transaction SE80 and choose or create a function group.
2. Create RFC enabled Function Modules in your function group

Go to transaction SE80 and choose or create a function group.
3.1 Create Web Service for your function group

Right click on the Function Group and choose “Create -> Web Service”
3.2 Create Web Service for your function group

Choose Continue to start creating the web service with the wizard

With this wizard a Service Definition can be quickly created and tested for an existing functionality.

A Web Service will be created for the RFC-Enabled Function Group

Z_VC_BAPIS
3.3 Create Web Service for your function group

Give your web service a technical name and description

Enter a name and a short description for the Web Service and choose an endpoint type.

To change the Web Service, use the ABAP Repository Browser (transaction SE80).

Service Definition: Z_WEB_SERVICE01
Short Text: Web Service for VC BAPIs
Endpoint Type: Function Group
3.4 Create Web Service for your function group

Choose continue to use the function group you’ve defined

Enter the name of the function group that is to be offered as Web Service.

If you select the checkbox ‘Mapping of Names’, the existing descriptions of the end point are applied. Initial letters are capitalized and underscores removed.

Function Group: Z_YC_BAPIS

[Options: Back, Continue, Cancel]
3.5 Create Web Service for your function group

Choose all the function modules you want available in this WSDL definition and choose Continue.
3.6 Create Web Service for your function group

Choose your authentication and release the service for runtime

Choose a Profile for Security Settings.

If you select the checkbox 'Release Service for Runtime', the Web Service is immediately released when completed. If the field is not selected, the Service can be released later in transaction WSCONFIG (Release Web Services for SOAF Runtime).

Profile: Basic Authorization SOAP Profile

- Profil für SOAP 1.1 mit zustandsloser HTTP Kommunikation und einfacher Authentisierung

- Release Service for Runtime
3.7 Create Web Service for your function group

Choose complete to create the web service!

The following object is created:

- The service Z_WEB_SERVICE01
4.1 Testing your web service

Go to transaction WSADMIN

Go to -> Administration Settings

Ensure that the path to your J2EE is specified here and save this.
4.2 Testing your web service

Find your web service and choose the “Web service homepage”
4.3 Testing your web service

Login to your ABAP system from the Web Service Navigator

Authorization

The selected endpoint requires basic authentication. Please, enter correct username and password:

Username: 805866
Password: ********

Submit
4.4 Testing your web service

Choose your BAPI to test from this web service definition

Operations
- Z_BAPI_ADD_DATA [test.types.Z_BAPI_ADD_DATA parameters]
- Z_BAPI_DELETE_MASTER_DATA [test.types.Z_BAPI_DELETE_MASTER_DATA parameters]
- Z_BAPI_READ_MASTER_DATA [test.types.Z_BAPI_READ_MASTER_DATA parameters]
- Z_BAPI_READ_MASTER_DATA_2 [test.types.Z_BAPI_READ_MASTER_DATA_2 parameters]
- Z_BAPI_RECORD_COPY [test.types.Z_BAPI_RECORD_COPY parameters]
- Z_BAPI_RECORD_COPY2 [test.types.Z_BAPI_RECORD_COPY2 parameters]
- Z_BAPI_RECORD_COPY3 [test.types.Z_BAPI_RECORD_COPY3 parameters]
- Z_BAPI_RECORD_COPY4 [test.types.Z_BAPI_RECORD_COPY4 parameters]
- Z_BAPI_RECORD_COPY5 [test.types.Z_BAPI_RECORD_COPY5 parameters]
- Z_BAPI_RECORD_COPY6 [test.types.Z_BAPI_RECORD_COPY6 parameters]
- Z_BAPI_WRITE_MASTER_DATA [test.types.Z_BAPI_WRITE_MASTER_DATA parameters]
- Z_BAPI_WRITE_MASTER_DATA_TXTS [test.types.Z_BAPI_WRITE_MASTER_DATA_TXTS parameters]
- Z_BAPI_WRITE_TO_CUBE [test.types.Z_BAPI_WRITE_TO_CUBE parameters]
- Z_BAPI_WRITE_TO_CUBE2 [test.types.Z_BAPI_WRITE_TO_CUBE2 parameters]
- Z_BAPI_WRITE_TO_CUBE4 [test.types.Z_BAPI_WRITE_TO_CUBE4 parameters]
4.5 Enter parameters and ensure successful response

Enter parameters and submit the request and ensure you have a successful HTTP response

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetQueryViewData</td>
<td>GetQueryViewData</td>
</tr>
<tr>
<td>Infopolicy (String) OD_DX_M01</td>
<td>AxisData (test.types.p1.Rrws$xAxisData)</td>
</tr>
<tr>
<td>Parameter (test.types.p1.W3query[]) SKIP</td>
<td>(test.types.p1.Rrws$xAxisData)</td>
</tr>
<tr>
<td>Query (String) OD_DX_M01_Q0006</td>
<td>(test.types.p1.Rrws$xAxisTuple[])</td>
</tr>
<tr>
<td>ViewId (String)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Consuming BI Data in Applications

There are many applications that can consume web services.

In this section, we’ll consume the BI Web Services with these tools:

1. Visual Composer – SAP Modeling Tool to build xApps
   - Consume BI Data via XMLA web service
   - Consume BI Data via query_view_data web service
   - Consume BI Data via direct Web API Connection

2. XML for Analysis Tool – Standalone application to consume xmla web service
   - Consumer BI Data via XMLA web service
Visual Composer can connect to multiple applications and web services.

For Visual Composer to connect to an XMLA connection, this XMLA web service needs to be mapped as a system on the SAP NetWeaver Portal.
Transaction MDXTEST allows you to test your MDX Statements!
1.1 Setup - Consuming BI Data via XMLA with Visual Composer

Login to the NetWeaver Portal and create a new system from the PAR

Choose com.sap.portal.systems.bi

---

Step 1: Portal Archive Selection

This wizard guides you through the steps required to create either a new system or a system from a portal archive (PAR) file.

Choose Portal Archive: *
- com.sap.km.cm.repository.manager
- com.sap.km.common.domino
- com.sap.netweaver.coll.appl.gw
- com.sap.netweaver.coll.appl.sync
- com.sap.pct.mdm.tech.systems
- com.sap.portal.httpconnector.urlsystem
- com.sap.portal.ivs.sl.connector.helper
- com.sap.portal.runtime.application.soap
- com.sap.portal.systems.bi
- com.sap.portal.systems.datasource
- com.sap.portal.systems.EBC
1.2 Setup - Consuming BI Data via XMLA with Visual Composer

Choose the SAP_BI_XMLA system

Specify the system name

<table>
<thead>
<tr>
<th>Step 2: Portal Component Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>The items below are all components contained in the portal archive. A new system or template derives its properties from the component.</td>
</tr>
<tr>
<td>Choose Portal Component: *</td>
</tr>
<tr>
<td>○ SAF_BI_JDBC</td>
</tr>
<tr>
<td>○ SAF_BI_ODBC</td>
</tr>
<tr>
<td>○ SAF_BI_SAPQuery</td>
</tr>
<tr>
<td>○ SAF_BI_XMLA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3: General Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Name: *</td>
</tr>
<tr>
<td>BIR_XMLA</td>
</tr>
<tr>
<td>System ID: *</td>
</tr>
<tr>
<td>BIR_XMLA</td>
</tr>
<tr>
<td>System ID Prefix (Example: com.companyname):</td>
</tr>
<tr>
<td>com.sap.portals</td>
</tr>
<tr>
<td>Master Language: *</td>
</tr>
<tr>
<td>English</td>
</tr>
</tbody>
</table>

THE BEST-RUN BUSINESSES RUN SAP
Create the BI XMLA System within the Portal

Specify the connection properties, user management, and System Alias.
Test the Connection

1. Navigate back to the System Config area.
2. Choose the BI XMLA System created.
3. Navigate to the Connection Tests area.
4. Choose the Test Button.
5. You are done!

Test Connection with Connector

Test Details:
The test consists of the following steps:
1. Retrieve the default alias of the system.
2. Check the connection to the backend application using the connector defined in this system object.

Results:
Retrieval of default alias successful
Connection successful
Verify Data is being retrieved

1. Login to the Visual Composer Storyboard
2. Execute the BI Content Wizard and select the XMLA system created.
3. Choose a Template
4. Configure the template with your desired settings
5. Choose Finish

Select a Query Template
Select a template that addresses your business question.

Configure Top N Parameters
Configure the parameters of the Top N query.

Define Query Ranks and Values
- Top rank type: Count, Value = 10
- Bottom rank type: Percent, Value = 5

Define Members
- Dimension: Material, Hierarchy: Material, Level: Material Level 01
- According to measure: Billed Quantity
Verify Data is being retrieved

6. Execute the Generate Flash Button
7. Deploy the iView
8. Verify Data was retrieved.

### RKT/DKT Training: XMLA

<table>
<thead>
<tr>
<th>Materials</th>
<th>Billed Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Picture Tube Monitor flat 17 CN</td>
<td>228,214.000 ***</td>
</tr>
<tr>
<td>Casing Monitor flat 17 CN</td>
<td>227,014.000 ***</td>
</tr>
<tr>
<td>Logic Monitor flat 17 CN</td>
<td>226,734.000 ***</td>
</tr>
<tr>
<td>Casing Terminal P600 CN</td>
<td>211,986.000 ***</td>
</tr>
<tr>
<td>Motherboard Terminal P600 CN</td>
<td>211,696.000 ***</td>
</tr>
<tr>
<td>Harddrive Terminal P600 CN</td>
<td>210,256.000 ***</td>
</tr>
<tr>
<td>Harddrive Terminal P400 CN</td>
<td>171,860.000 ***</td>
</tr>
<tr>
<td>Motherboard Terminal P400 CN</td>
<td>170,020.000 ***</td>
</tr>
<tr>
<td>Casing Terminal P400 CN</td>
<td>169,480.000 ***</td>
</tr>
<tr>
<td>Picture Tube Monitor flat 21 CN</td>
<td>160,540.000 ***</td>
</tr>
<tr>
<td>Top Total</td>
<td>1,987,800.000 ***</td>
</tr>
</tbody>
</table>
2.1 Connection within 3rd Party Tool

In this example, we will use the XML for Analysis Tool that is free on SDN to demonstrate consuming BI Data via a web service with a 3rd party tool.

**SOAP messages**

**SOAP Request**

```xml
  <SOAP-ENV:Body>
      <Command>
        <Statement> SELECT [[Measures], [01FKSQMARTYCDNBHGDOL/New]], ON COLUMNS, NON EMPTY EXCEPT(TOPCOUNT([0SALESORG].MEMBERS, 6, [Measures],[01FKSQMARTYCDNBHGDOL/New]), [0SALESORG].[All]), ON ROWS FROM [0SD_C03/0SD_C03_C0100] </Statement>
      </Command>
      <PropertyList>
        <BodyInfo>
          <Catalog>Froodmart 2000</Catalog>
          <Format>Multi-dimensional</Format>
          <AxisFormat>TableFormat</AxisFormat>
          <Content>Data</Content>
        </BodyInfo>
        <PropertyList>
        </PropertyList>
      </PropertyList>
    </Execute>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

**SOAP Response**

```xml
  <SOAP-ENV:Body>
    <ExecuteResponse xmlns="urn:schemas-microsoft-com:xml-analysis">
      <return xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xsd:string">
        <root xmlns="urn:schemas-microsoft-com:xml-analysis" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <CubeInfo>
            <Cube>
              <CubeName>[0SD_C03/0SD_C03_C0100]</CubeName>
            </Cube>
          </CubeInfo>
          <AxesInfo>
            <AxisInfo name="Axis1">
              <HierarchyInfo name="Measures">
                <MeasureInfo name="[Measures].MEMBER_1NAME" />
              </HierarchyInfo>
            </AxisInfo>
            <AxisInfo name="Axis2">
              <HierarchyInfo name="[Measures].MEMBER_2NAME" />
            </AxisInfo>
          </AxesInfo>
        </root>
      </return>
    </ExecuteResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
2.2 Connection within 3rd Party Tool

Here is an example of a 3rd Party Javascript/HTML application that consumes BI data through the xmla web service.
2.3 Connection within 3rd Party Tool

Sample tool is available on SDN!


**SOAP messages**

**SOAP Request**

```xml
  <SOAP-ENV:Body>
    <Discover>
      <DiscoverType>MDSCHEMA_CUBES</DiscoverType>
      <Restrictions/>
      <Properties/>
      <PropertyList>
        <DataSourceInfo>default</DataSourceInfo>
        <Catalog>SD_EX_M01</Catalog>
        <Format>Tabular</Format>
      </PropertyList>
    </Discover>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

**SOAP Response**

```xml
  <SOAP-ENV:Body>
    <DiscoverResponse xmlns="urn:schemas-microsoft-com:xml-analysis">
      <!-- Content of DiscoverResponse goes here -->
    </DiscoverResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
Introduction
Overview of Web Services and Standards
SAP BI Web Service for Queries
Creating BI Web Services for BAPIs
Consuming BI Web Services within Applications
Consuming Other Web Services within Applic.
Service Oriented Architecture Scenarios
Summary
Consuming BI BAPI based Web Service

Within Visual Composer, you can consume your custom web services as well (see section on building web services off BAPI’s)!!!

Steps:

1. Create Web Service in source system (BI or ERP)
   - See section on creating these web services (earlier in this presentation)
2. Map Web Service Destination on the SAP NetWeaver Portal
3. Use Web Service in Visual Composer Model!
## 2.1 Create Web Service System on SAP NetWeaver Portal

### Create New System (from Template)

Choose Web Service System using WSDL URL

<table>
<thead>
<tr>
<th>Step 1:</th>
<th>Template Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choose Template: *</td>
<td>- BI JDBC System</td>
</tr>
<tr>
<td></td>
<td>- BI ODBC-Compliant OLAP System</td>
</tr>
<tr>
<td></td>
<td>- BI SAP Query System</td>
</tr>
<tr>
<td></td>
<td>- BI XMLA-Compliant OLAP System</td>
</tr>
<tr>
<td></td>
<td>- CEP 5.0 System</td>
</tr>
<tr>
<td></td>
<td>- HTTP System</td>
</tr>
<tr>
<td></td>
<td>- JDBC System</td>
</tr>
<tr>
<td></td>
<td>- KM Lotus System</td>
</tr>
<tr>
<td></td>
<td>- KM WebDAV System</td>
</tr>
<tr>
<td></td>
<td>- KM Windows System</td>
</tr>
<tr>
<td></td>
<td>- MDM System</td>
</tr>
<tr>
<td></td>
<td>- Portal Tenant System</td>
</tr>
<tr>
<td></td>
<td>- SAP system using connection string</td>
</tr>
<tr>
<td></td>
<td>- SAP system using dedicated application server</td>
</tr>
<tr>
<td></td>
<td>- SAP system with load balancing</td>
</tr>
<tr>
<td></td>
<td>- Web Service System using WSDL URL</td>
</tr>
</tbody>
</table>
2.2 Create Web Service System on SAP NetWeaver Portal

Name your system

Set your WSDL Destination for your custom web service

Property Editor - DOC_WEBSERVICE

<table>
<thead>
<tr>
<th>Property Category:</th>
<th>Web Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Force SAP UI Parameter Management</td>
<td>Yes</td>
</tr>
<tr>
<td>WSDL Caching (Days)</td>
<td>1</td>
</tr>
</tbody>
</table>
2.3 Create Web Service System on SAP NetWeaver Portal

Set your user Management

Create and save a system alias for your system

Property Editor - DOC_WEBSERVICE

Property Category: User Management

- Logon Method: SAPLOGONTICKET
- User Mapping Type: admin,user

System Alias Editor

Create new aliases for this system or delete existing ones. A system alias can be any name, other interfaces, such as User Mapping.

Alias Name: [Input Field]

Defined Aliases

<table>
<thead>
<tr>
<th>Default</th>
<th>Alias Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BI_DOC_VWS</td>
</tr>
</tbody>
</table>

THE BEST-RUN BUSINESSES RUN SAP
2.4 Create Web Service System on SAP NetWeaver Portal

Test and ensure successful connection!!!

System Connection Tests

You can test the connectivity to the backend application represented by the current system object. Choose the relevant tests. Note: The test is based on the properties currently defined in the system object. It does not check if the system properties are correct.

<table>
<thead>
<tr>
<th>Test Name</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Test for Connectors</td>
<td>Tests the connection to a backend system using an associated connector</td>
<td>✔</td>
</tr>
</tbody>
</table>

Results

✔ Test Connection with Connector

Test Details:
The test consists of the following steps:
1. Retrieve the default alias of the system
2. Check the connection to the backend application using the connector defined in this system object

Results
Retrieval of default alias successful
Connection successful
2.5 Create Web Service System on SAP NetWeaver Portal

Use your web service system within VC!!!
Introduction
Overview of Web Services and Standards
SAP BI Web Service for Queries
Creating BI Web Services for BAPIs
Consuming BI Web Services within Applications
Consuming Other Web Services within Applic.
Service Oriented Architecture Scenarios
Summary
BPX Modeling within E-SOA

See this BLOG on SDN:

https://weblogs.sdn.sap.com/pub/wlg/3933

This BLOG discusses looking at consuming multiple services!

1. Service from freight vendors that are available
2. Service for Mileage
   • Google Maps
   • Mappoint
3. Service from SAP NetWeaver Master Data Management
   • Master data for your customers
   • Shipfrom and Shipto Cities
4. Services for BI/ERP Data
   • Transactions from SAP R/3
   • Transactions from SAP NetWeaver BI
Introduction
Overview of Web Services and Standards
SAP BI Web Service for Queries
Creating BI Web Services for BAPIs
Consuming BI Web Services within Applications
Consuming Other Web Services within Applic.
Service Oriented Architecture Scenarios
Summary
Summary

You should know…

- What are web services?
  - Self-contained, self-describing, modular application functionalities, once deployed, applications and other Web services can be discovered and directly invoked via service requests based on open standards for system interaction, independent of technical architecture.

- How do you create web services within BI?
  - Build Web Services from RFC Enabled Function Modules

- How do you consume SAP Delivered BI web services?
  - XMLA
  - QUERY_VIEW_DATA

- Why you should use web services?
  - E-SOA
Summary – Additional Information

Common Links

- Online Help:
  - [http://help.sap.com/saphelp_nw04/helpdata/en/d8/3bfc3f8fc2c542e10000000a1550b0/frameset.htm](http://help.sap.com/saphelp_nw04/helpdata/en/d8/3bfc3f8fc2c542e10000000a1550b0/frameset.htm)

- Example of JAVA Based Web Service:

- Example of ABAP Based Web Service:
  - [http://help.sap.com/saphelp_nw04/helpdata/en/bf/d005244e9d1d4d92b2fe7935556b4c/frameset.htm](http://help.sap.com/saphelp_nw04/helpdata/en/bf/d005244e9d1d4d92b2fe7935556b4c/frameset.htm)

- Consuming a Web Service via a BSP

- BLOG Series on Web Services by Scott Cairncross
Appendix
Here’s a simple MDX tool I put together based off the XML Analysis example

Simple MDX Tool
Do You Have Other...?

- Questions
- Issues
- Feedback
Next Call

Please join us again on August 31, 2006 at 11 AM EST.

We will continue our Web Services series of webinars with: “Consuming a BI Web Service via the Portal”

As always, a replay of this call and the presentation material will be available at http://service.sap.com/nw-cc in a few days.

Also, we will request a few moments of your time to fill out a survey. Your input is invaluable!