

## **Tutorial: Building an Android application using SMP and eBay OData**

Includes code snippets for onboarding, CAPTCHA handling

**Author:**  
Suhas Gatt Sridhar

## TABLE OF CONTENTS

<b>INTRODUCTION</b> .....	<b>3</b>
<b>PREREQUISITES</b> .....	<b>3</b>
<b>STEPS</b> .....	<b>3</b>
<b>1. SMP: CONFIGURING THE APPLICATION IN SMP</b> .....	<b>3</b>
1.1 CREATE A NEW APPLICATION IN SMP .....	3
1.2 CONFIGURE APPLICATION ENDPOINT .....	4
1.3 CONFIGURE AUTHENTICATION SETTINGS .....	5
<b>2 CLIENT: BUILD ANDROID APPLICATION</b> .....	<b>6</b>
2.1 CREATE ANDROID APPLICATION PROJECT .....	6
2.2 ONBOARDING TO SMP .....	7
2.3 FETCHING LIST OF ITEMS .....	10
<b>FURTHER INFORMATION</b> .....	<b>11</b>
<b>REFERENCES</b> .....	<b>11</b>

## INTRODUCTION

**SAP Mobile Platform (SMP), enterprise edition, cloud version** can be used to build lightweight, on-demand applications. The platform offers authentication, secure on-boarding, native push notifications, and reporting capabilities for enterprise mobile applications.

The following tutorial elucidates using SMP to build a simple B2C application. It describes the ease of configuring an application on SMP and a quick glance on the REST based APIs exposed by the platform. This tutorial will guide you in developing a simple Android application to display a list of items on eBay using SMP. In this scenario, we will use the eBay OData service as the Backend. This application showcases the special features of SMP namely CAPTCHA validation and Anonymous access.

## PREREQUISITES

Before kick-starting the application configuration, make sure of the following:

- You have a SAP HANA Cloud account subscribed with SAP Mobile Platform application. For information on how to obtain a HANA account with SMP, refer the following link. (<https://help.hana.ondemand.com/mobile/frameset.htm?doc/html/mdw1361529553461.html>)
- You have installed Eclipse with Android SDK and ADT for developing the Android application. (For more information on installing Eclipse and Android ADT, refer <http://developer.android.com/sdk/installing/installing-adt.html>)

## STEPS

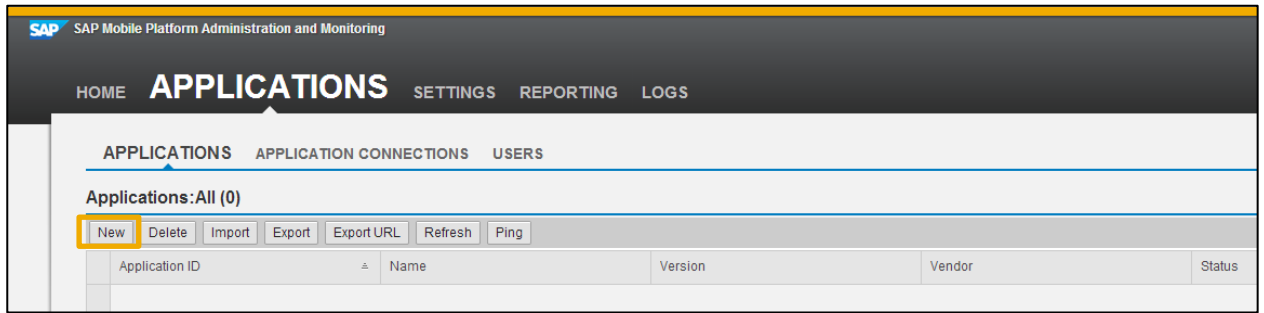
### 1. SMP: CONFIGURING THE APPLICATION IN SMP

Before users can connect to eBay system through SMP, you need to configure the application in the SMP Admin portal.

During the configuration, you need to provide basic details about your application like Application Endpoint and Authentication type. The steps to configure an application on SMP are:

#### 1.1 CREATE A NEW APPLICATION IN SMP

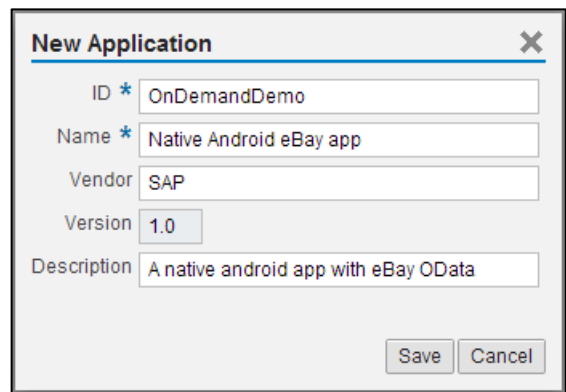
1. Open your SAP HANA Account page, ([https://smp-<account\\_name>.hanatrial.ondemand.com](https://smp-<account_name>.hanatrial.ondemand.com)) The SAP HANA account Welcome page is displayed with a link to **SAP Mobile Platform, enterprise edition, cloud version** -> Click on **SAP Mobile Platform, enterprise edition, cloud version** to open the SAP Mobile Platform Administration and Monitoring portal. (For more information on accessing the SMP Admin portal, refer <https://help.hana.ondemand.com/mobile/frameset.htm?doc/html/soo1349339151497.html>)
2. In the SMP Admin portal, click on the Applications tab. Click “New” to configure an application.



Configuring an application

3. In the New Application popup, provide the following details and click Save. The Application Overview window is displayed.

- a. **ID:** <Application ID>
- b. **Name:** <Name of the Application>
- c. **Vendor:** <Name of Vendor>
- d. **Version:** <Application Version>1.0 (by Default)
- e. **Description:** <Description of the Application>

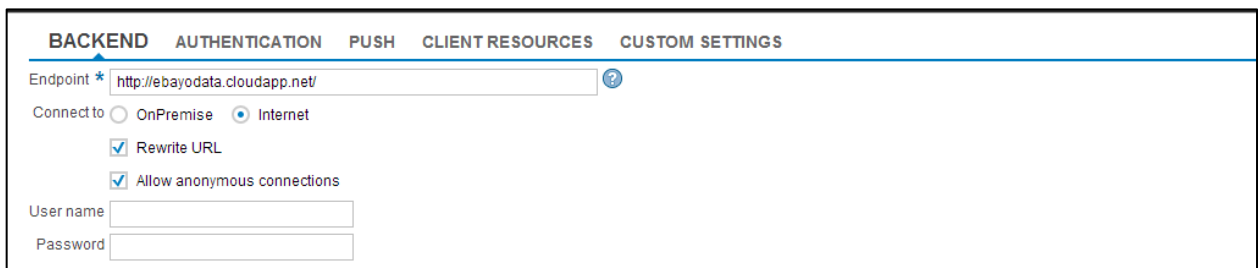


New Application Window

**NOTE:** If you are using SMP on a landscape other than hanatrial.ondemand.com, you will need an additional privilege "SMP Administrator" to configure applications. Please [refer here](#) for different Platform roles in SMP.

### 1.2 CONFIGURE APPLICATION ENDPOINT

Under the Backend tab in Application Details window, you need to specify the Backend which acts as the primary data provider (Backend Enterprise Information Server) for your application. In this case, it is the eBay system, so provide the eBay OData URL as the Application End Point (<http://ebayodata.cloudapp.net/>).



Backend Details

Enter:

- a. **End Point:** <http://ebayodata.cloudapp.net/>. This is the base URL to which all the requests will be made
- b. **Connect To:** Internet. Since it is an Ondemand Backend, we choose Internet.

- c. **Rewrite URL:** This property denotes that the backend URL will be rewritten with the cloud URL in the response. It is enabled by default.
- d. **Allow Anonymous Connections:** Select this configuration if you want to allow anonymous access from the application. eBay OData Service does not expect any authorization. Hence we configure our application with anonymous access so that mobile users need not pass an authorization header in their HTTP requests.
- e. **Username and Password:** You can leave these fields blank as eBay system does not require any credentials.

**NOTE:** If the Backend is within a firewall, then we need to choose OnPremise and whitelist the URL in SAP Cloud Connector. For more details on configuring application endpoint please [refer here](#)

### 1.3 CONFIGURE AUTHENTICATION SETTINGS

Under Authentication tab in Application Details window, you need to specify the authentication type which your application will use. Choose the “New” option to create a new Security Profile.

The screenshot shows the 'AUTHENTICATION' tab in the 'Application Details' window. At the top, there are tabs for 'BACKEND', 'AUTHENTICATION', 'PUSH', 'CLIENT RESOURCES', and 'CUSTOM SETTINGS'. Under 'AUTHENTICATION', there are radio buttons for 'Security Profile' with 'New' selected and 'Existing' unselected. Below this is a text field for 'Security Profile Name' containing 'OnDemandSC'. The page is divided into two main sections: 'General Settings' and 'Client password policy'. 'General Settings' includes three input fields: 'Authentication cache timeout (seconds)' with '3600', 'Maximum number of failed authentications' with '5', and 'Authentication lock duration (seconds)' with '60'. There is also a checked checkbox for 'Enable CAPTCHA'. 'Client password policy' has an unchecked checkbox for 'Enable password policy'. The 'Authentication Type' section has a dropdown menu set to 'Basic Authentication' and an 'Authentication URL' field containing 'http://ebayodata.cloudapp.net/'. At the bottom, there are radio buttons for 'Connect to' with 'OnPremise' unselected and 'Internet' selected.

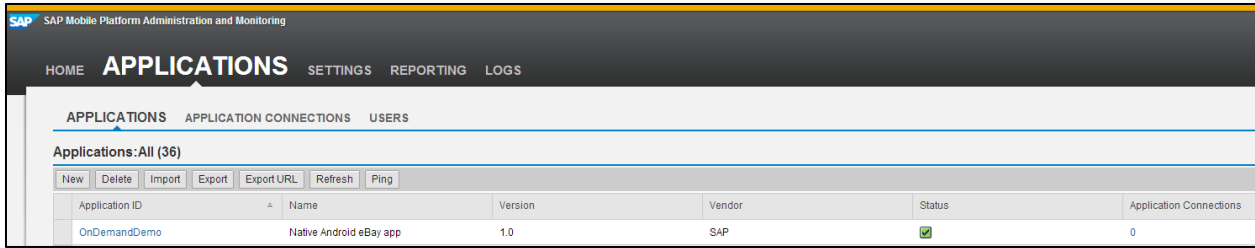
#### Authentication details

Enter the following details:

- a. **Security Profile Name:** <Name of the Security Profile>
- b. **Enable CAPTCHA:** Select this checkbox. We enable CAPTCHA to safeguard the application from Denial of Service (DOS) attacks.
- c. **Authentication Type:** Choose Basic Authentication
- d. **Authentication URL:** <http://ebayodata.cloudapp.net/>. This is the HTTP URL which authenticates the user. We will use the same eBay OData URL.
- e. **Connect To:** Internet. We choose Internet since the Authentication URL is a public URL

**NOTE:** We have used Basic Authentication for our application. However SMP also supports other authentication types. More information on Authentication Settings can be [found here](#)

Click Save to create the application-> Click Yes when you are prompted to confirm whether you want to update the application. The Application will be successfully created. Check if the Status is  , confirming that the application is correctly configured.



Configured Application

You have now successfully configured the application on SMP in just a few easy steps and can start building the Android application.

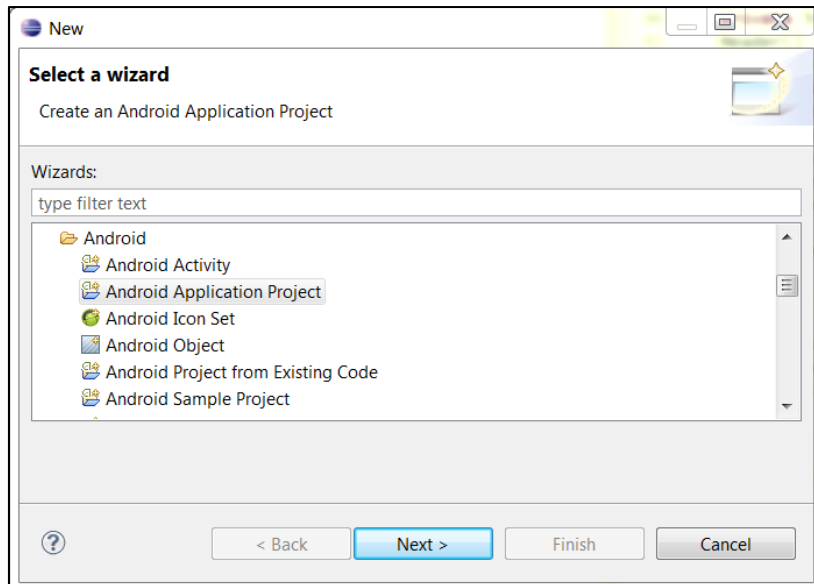
## 2 CLIENT: BUILD ANDROID APPLICATION

For this tutorial we use Eclipse with ADT plugin as the development environment. This tutorial uses Eclipse Indigo and Android API level 8.

### 2.1 CREATE ANDROID APPLICATION PROJECT

First you need to create an Android Application Project.

1. In Eclipse, go to File -> New -> Other. Under Android, choose Android Application project and click Next.



New Android Project

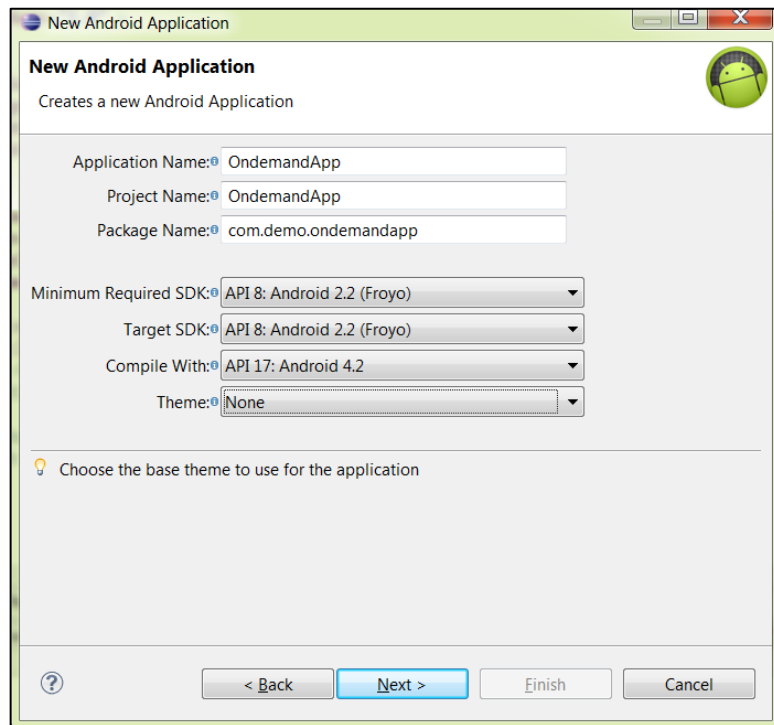
2. In the New Android Application window, provide the following details:

**Application Name:** OndemandApp

**Project Name:** OndemandApp

**Package Name:** com.demo.exampleApp

Choose API 8: Android 2.2 for both Minimum Required SDK and Target SDK. Click Next.



*Android Application details*

3. In the Configure Project window, choose Create Activity and Create Project in Workspace. Enter a location where the application will be saved. Click Next.
4. In the Create Activity window, choose BlankActivity. Click Next.
5. In the New Blank Activity window, provide the following details:
  - a. Activity Name: LoginActivity
  - b. Layout Name: activity\_login
  - c. Choose None as Navigation Type

Click Save. An Android Application project will be created.

Under the newly created project, open AndroidManifest.xml and add the following permissions.

```
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
```

You have now created an Android application. You can continue with writing the LoginActivity for onboarding a new user to SMP.

**NOTE:** For this tutorial, we have chosen API Level 8; you could choose any other API level as per your choice.

## 2.2 ONBOARDING TO SMP

The LoginActivity handles onboarding a user to the configured OndemandDemo application on SMP. For onboarding, the application must send a HTTP POST request to the URL:

**https://{smp base URL}/public/odata/applications/{latest|v1}/{appid}/Connections**

The URL contains these components:

- smp base URL – SMP Application URL as available in accounts page. For example, mobile<account\_name>-<tenantname>.hana.ondemand.com
- public – Used to create an anonymous connection
- odata/applications – Refers to the OData services associated with the application resources.
- {latest|v1} – The version of the service document
- appid – The Application ID as configured in SMP Admin Portal
- Connections – Name of the OData collection.

The Request body being:

```
<?xml version='1.0' encoding='utf-8'?>
<entry xmlns="http://www.w3.org/2005/Atom"
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
<title type="text"/>
<updated>2012-06-15T02:23:29Z</updated>
<author>
  <name/>
</author>
<category term="applications.Connection"
scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
<content type="application/xml">
<m:properties>
  <d:DeviceType>Android</d:DeviceType>
</m:properties>
</content>
</entry>
```

We have specified Android as the Device Type.

The following HTTP Headers need to be sent with the request:

HEADER	DESCRIPTION	VALUE
X-SUP-APPCID	Used to uniquely identify an Application connection. The same value must be passed for all future requests made to SMP	We use a random UUID generator to generate an ID
X-SUP-CAPTCHA-TEXT	Contains the CAPTCHA text as entered by user	Text corresponding to CAPTCHA challenge sent by SMP
Content-Type	Standard HTTP Header denoting the MIME type of the body of the request	application/atom+xml

**Request headers for Onboarding**

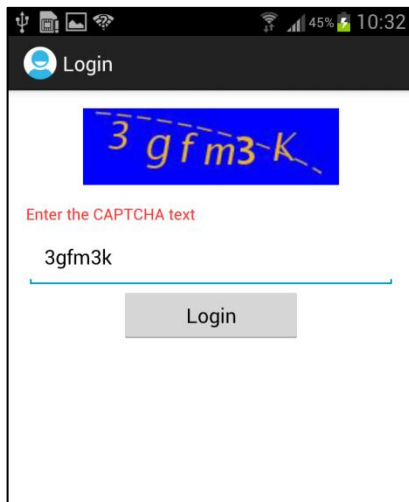
**NOTE:** If you do not pass X-SUP-APPCID header with the first request, SMP will generate an APPCID and send it as a response cookie. The same has to be used for all further requests.



Following is the code snippet for the first onboarding request:

```
//Create HTTP Post object with Onboarding URL
HttpPost httpPost = new HttpPost(URL);
httpPost.setHeader("Content-Type", "application/atom+xml");

//Provide a random or unique Application Connection ID
httpPost.setHeader("X-SUP-APPCID", APPCID);
//Set the Request Body
StringEntity entity = new StringEntity(body);
httpPost.setEntity(entity);
//Create HTTP Client and make HTTP Request
HttpClient httpRequest = new DefaultHttpClient();
HttpResponse httpResponse = httpRequest.execute(httpPost);
int status = httpResponse.getStatusLine().getStatusCode();
```



Application Login Screen

SMP Server responds with status code 401 and a CAPTCHA image in Base64 encoded format. You need to decode the image and display it to the user using an ImageView

```
if(status == 401) {
    //In case of 401 and WWW-Authenticate: CAPTCHA header, server responded with
    CAPTCHA challenge
    responseBody = EntityUtils.toString(httpResponse.getEntity(), "utf-8");

    try {
        byte[] bytearray = Base64.decode(responseBody, 0);
        Bitmap image = BitmapFactory.decodeByteArray(bytearray, 0, bytearray.length);
        ImageView captchaImageView = (ImageView) findViewById(R.id.captcha_image);
        captchaImageView.setImageBitmap(image);
    } catch(Exception ex) {
        Log.d("401 Parse Error", "Response body is not Base64 String");
        Log.d("Response Body", responseBody);
    }
}
```

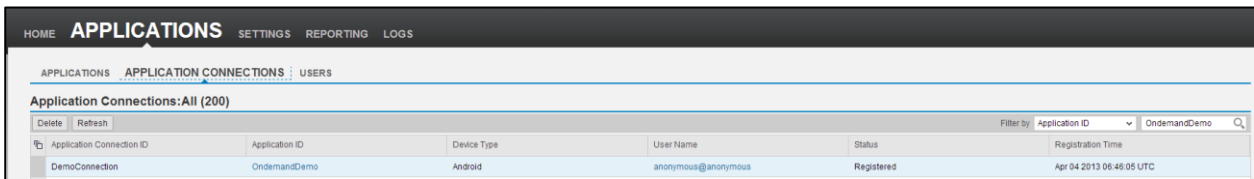
Once the user enters the CAPTCHA text, we need to again make a POST request with the same APPCID and CAPTCHA text.

```
EditText captchaEditText = (EditText) findViewById(R.id.captcha_text);  
String captcha_Text = captchaEditText.getText().toString();  
if(captcha_Text != null && !captcha_Text.equals("")) {  
    httpPost.setHeader("X-SUP-CAPTCHA-TEXT", captcha_Text);  
}
```

If the CAPTCHA text is correct, then onboarding will be successful and SMP returns a 201.  
If CAPTCHA text was incorrect, SMP returns a new CAPTCHA image with status 401.

```
if(status == 201) {  
    //In case of 201 Created, Onboarding successful and application connection is  
    created  
    responseBody = EntityUtils.toString(httpResponse.getEntity(), "utf-8");  
    Log.d("Onboard", "Onboarding Completed");  
    //Start new Activity to fetch movies and display them  
    Intent goToNextActivity = new Intent(LoginActivity.this, DisplayMovies.class);  
    //Pass the APPCID to the next activity  
    goToNextActivity.putExtra("APPCID",APPCID);  
    startActivity(goToNextActivity);  
}
```

Once the onboarding is complete, you can view the newly created connection the SMP Admin Portal under Applications tab -> Application Connections.



Application Connection

### 2.3 FETCHING LIST OF ITEMS

We will use the eBay OData catalog to display the list of OData Programming books available. To fetch the list of books, we make a GET to the URL:

```
https://{smp_base  
URL}/public/<<application_ID>>/Items?search='odata%20programming'
```

This corresponds to the eBay URL <http://ebayodata.cloudapp.net/Items?search='odata%20programming'>. Here 'odata%20programming' is a search criterion used to search relevant titles. You can find more information on filters at <http://ebayodata.cloudapp.net/docs>

We need to pass X-SUP-APPCID Header with the APPCID value which was user for onboarding.



List of Items

```
//Obtain the APPCID and SMP URL from previous activity
final Intent i = getIntent();
String APPCID = i.getStringExtra("APPCID");
//Construct the URL to fetch titles: Same as
http://ebayodata.cloudapp.net/Items?search='odata%20programming'
String itemsURL = SMPURL + "/public/" + appID +
"/Items?search='odata%20programming'";
//Create a HTTP GET object
HttpGet httpGet = new HttpGet(itemsURL);
httpGet.setHeader("X-SUP-APPCID", APPCID);

try {
    HttpClient httpClient = new DefaultHttpClient();
    HttpResponse httpResponse = httpClient.execute(httpGet);
    int status = httpResponse.getStatusLine().getStatusCode();

    //Parse the oData response from eBay
    EBayOdataXMLParser parser = new EBayOdataXMLParser();
    List<String> resultList = parser.parse(httpResponse.getEntity().getContent());

    ListView listView = getListView();
    ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1, resultList);
    //Display the items as a simple List
    listView.setAdapter(adapter);
} catch (Exception e) {
    e.printStackTrace();
}
```

### FURTHER INFORMATION

You can [download the complete source code](#) for this project from the accompanying link on SDN. It also includes a custom parser which extracts list of Items from the response. To parse the OData response from eBay, you can build a custom parser or use existing ones like <http://code.google.com/p/odata4j/>

### REFERENCES

1. SMP Documentation: [https://help.hana.ondemand.com/mobile/frameset.htm?SMP\\_welcome.html](https://help.hana.ondemand.com/mobile/frameset.htm?SMP_welcome.html)
2. eBay OData Documentation: <http://ebayodata.cloudapp.net/docs>
3. Eclipse: <http://www.eclipse.org/downloads/>
4. Installing ADT plugin for Eclipse: <http://developer.android.com/sdk/installing/installing-adt.html>
5. Android SDK: <http://developer.android.com/sdk/index.html>
6. [About OData: http://www.odata.org/](http://www.odata.org/)

© 2013 SAP AG. All rights reserved.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, SAP HANA, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company.

Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase Inc. Sybase is an SAP company.

Crossgate, m@gic EDDY, B2B 360°, and B2B 360° Services are registered trademarks of Crossgate AG in Germany and other countries. Crossgate is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

