

# How to Access Images of SAP Netweaver Demo Model JAVA



## Applies to:

SAP Netweaver Composition Environment 7.2

For more information, visit the [User Interface Technology homepage](#).

## Summary

Up to Netweaver 7.2 the demo model data only contained urls to images for products and business partners. But since SAP Netweaver 7.2 the SAP NW Demo Model is shipped together with these images. This article addresses how these images can be accessed from WebDynpro JAVA. Therefore a demo application based on the demo model and images is presented here.

**Author:** Carl Heckmann

**Company:** SAP AG

**Created on:** 8 June 2009

## Author Bio



Carl Heckmann joined SAP in July 2008 as an intern during his Bachelor studies of Wirtschaftsinformatik at the University of Mannheim. He worked within the SAP Netweaver Product Management Early Demo Team where he worked with Netweaver CE 7.1 and Web Dynpro JAVA and got an insight into upcoming technologies such as Flash Islands or the NetWeaver Widget Foundation. Since October 2008 he is a working student and has been involved in the development of the SAP Netweaver Demo Model JAVA.

## Table of Contents

How to Access the Images .....	3
Scenario Description .....	3
Technical Description (optional) .....	3
Prerequisites .....	3
Building a Simple Web Dynpro .....	3
Importing the Web Service .....	3
Create the Component and Context .....	5
Create the Layout .....	7
Create, Deploy and Launch the Application .....	9
Related Content .....	10
Copyright .....	11

## How to Access the Images

### Scenario Description

This documentation describes how to build a simple Web Dynpro Application based on the **SAP Netweaver Demo Model JAVA** and how to access the included images for products and business partner. The images are provided together with the SAP Netweaver Demo Model.

### Technical Description (optional)

The images are stored within the WebContent folder of a specialized DC of type J2EE/WebModule and only referenced by an url. If the data of the demo model is generated and inserted into the database, the WebResource attribute of the entity product is only stored as an URL within the database. An example would be:

*sap.com~bc~nwdemo~sm~util~img/EarlyDemo/NWDemoDataModel/Data/Product/Pictures/HT-1000.jpg*

This is the path relative to the application server on which the demo model is running. If the picture should be used within a Web Dynpro application the absolute path is needed. This would look like the following:

<http://<hostname>:<port>/sap.com~bc~nwdemo~sm~util~img/EarlyDemo/NWDemoDataModel/Data/Product/Pictures/HT-1000.jpg>

This means, that it is necessary to determine the hostname and port of the system on which the demo model is deployed.

**Note:** But this also means, that developers, who want to enable users to change images, have to implement this carefully.

### Prerequisites

It is assumed, that you have

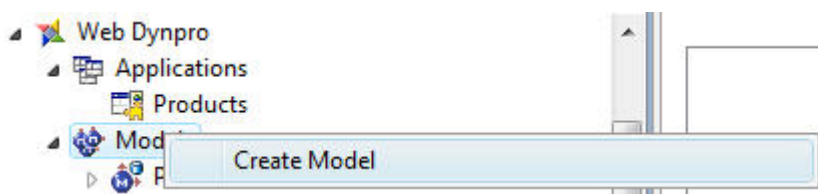
- a **SAP Netweaver 7.2 CE** System up and running with deploy rights
- the latest version of the **SAP Netweaver Demo Model JAVA**. (included with NW 7.2)
- deployed The **SAP Netweaver Demo Model JAVA**. (included with NW 7.2)
- a version of the **SAP Netweaver Development Studio (NWDS)**

### Building a Simple Web Dynpro

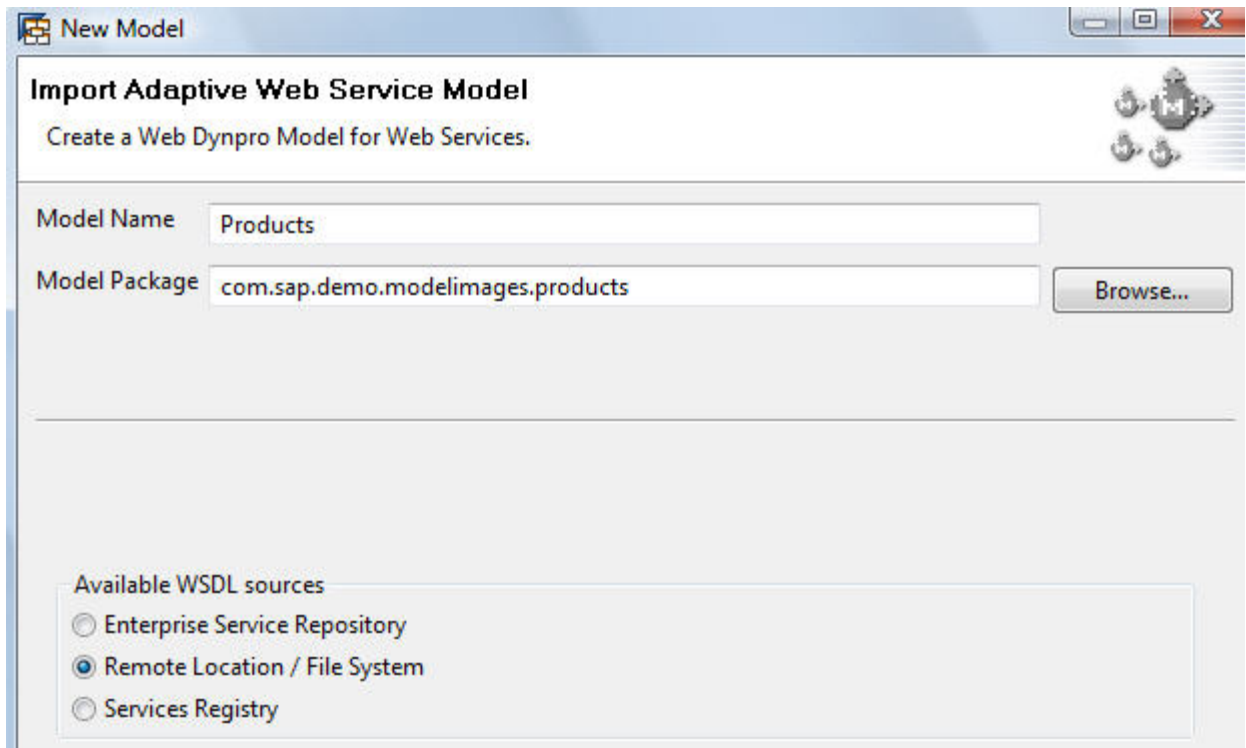
Open your Netweaver Developer Studio and switch to the **WebDynpro View**. Then select **File → New → Create Web Dynpro Component**. In the following dialog choose **LocalDevelopment → MyComponents**. Enter “**modelimages**” as name and finish the creation with a click on the finish button.

### Importing the Web Service

1. Right click on Models and select “Create Model”.



- Then choose “**Adaptive Web Service Model**”. In the next step you have to provide the web service description. Enter “**Product**” as name and select “**Remote Location / File System**”, as shown on the image below.



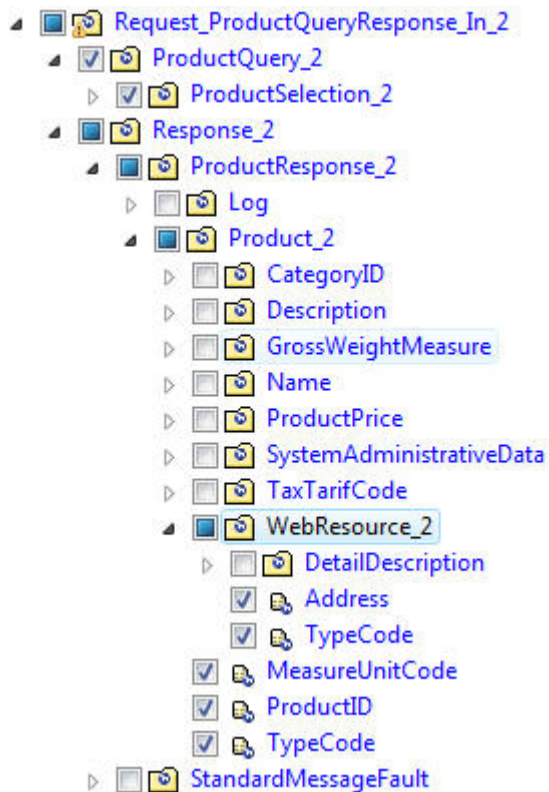
The screenshot shows a window titled "New Model" with the following content:

- Import Adaptive Web Service Model**
- Create a Web Dynpro Model for Web Services.
- Model Name:
- Model Package:
- Available WSDL sources:
  - Enterprise Service Repository
  - Remote Location / File System
  - Services Registry

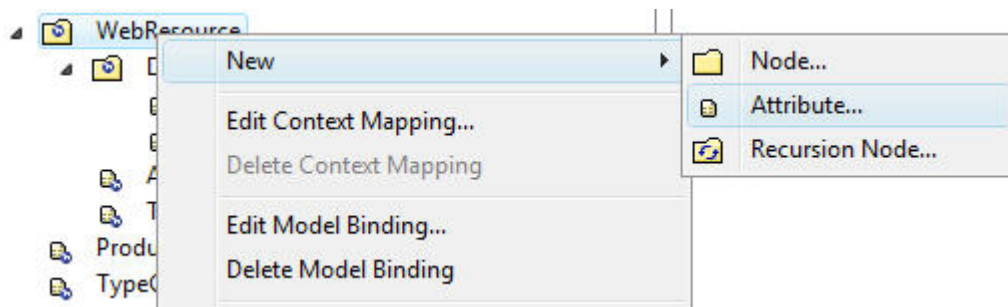
- The next step requires you to enter the url, where the service description is located. Enter [http://<hostname>:<port>/QueryProduct\\_In\\_Service/QueryProductInImplBean?wsdl&mode=ws\\_policy](http://<hostname>:<port>/QueryProduct_In_Service/QueryProductInImplBean?wsdl&mode=ws_policy) as url. Replace <hostname> and <port> with the corresponding values of your system. Then click on the finish button.

## Create the Component and Context

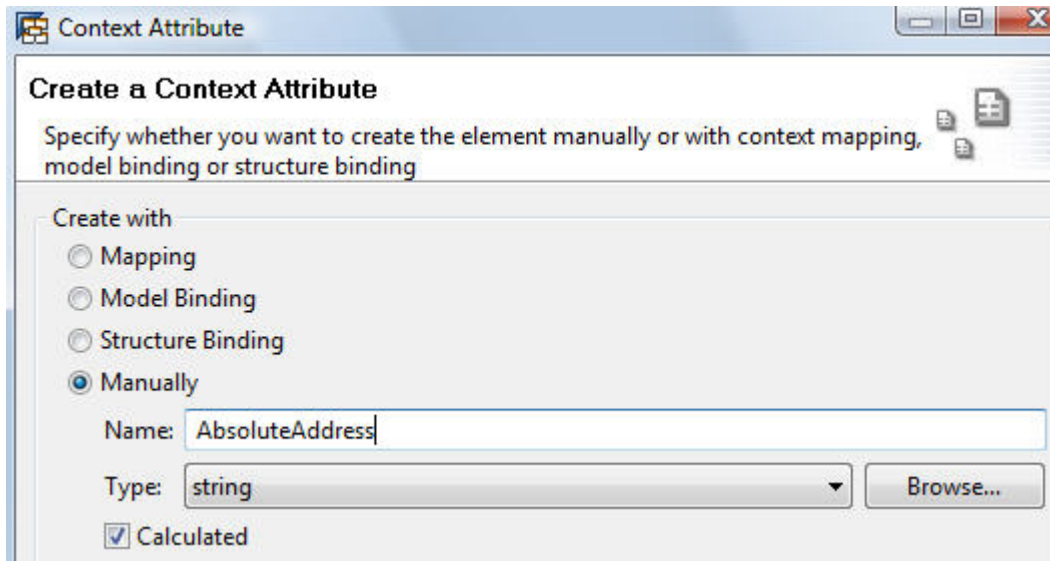
1. Right click on **ProductListView** → **Template** → **Apply** and select **ServiceController**
2. Select the “**Request\_ProductQueryResponse\_In**” service. In the next step select exactly the same nodes and attributes as shown in the screenshot below.



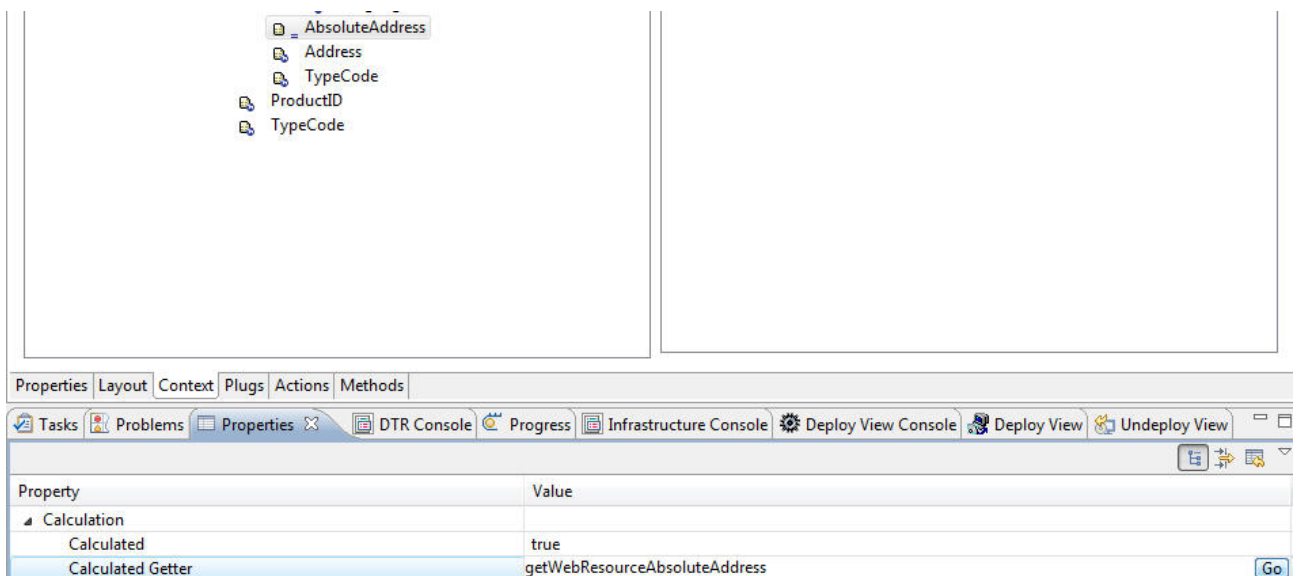
3. Right click on **WebResource** and change the property **Singleton** to **false**
4. The attribute address of node **WebResource** contains the path , where the images are located relatively to the application at the Netweaver application server. To enable the images to be accessed from anywhere this url has to be made absolute. Therefore we need the path to the application. To achieve this task we add a calculated attribute to the **WebResource** node. Right click on the **WebResource** node and select **New**→**Attribute**.



5. The added attribute has to be created manually. Enter **AbsoluteAddress** as name of the attribute and check the **Calculated** checkbox. Click on finish to proceed.



6. Save the file (**CTRL + S**) and select the newly added attribute. Then click on “GO” on the right side of property “**Calculated Getter**”.



- Now replace the body of the method **getWebresourceAbsoluteAddress** with the following code. This code is responsible for adding the system specific parts to the relative URLs, thus building an absolute URL.

```

WDDeployableObjectPart d = wdComponentAPI.getDeployableObjectPart();
String url = "";
try {
    url = wdComponentAPI.getURLService().getGeneralService()
        .getAbsoluteWebResourceURL(d);
} catch (WDURLException e) {
}

int posWD = url.indexOf("webdynpro");
if(posWD < 0)
    posWD = 0;

String cuttedUrl = url.substring(0, posWD);

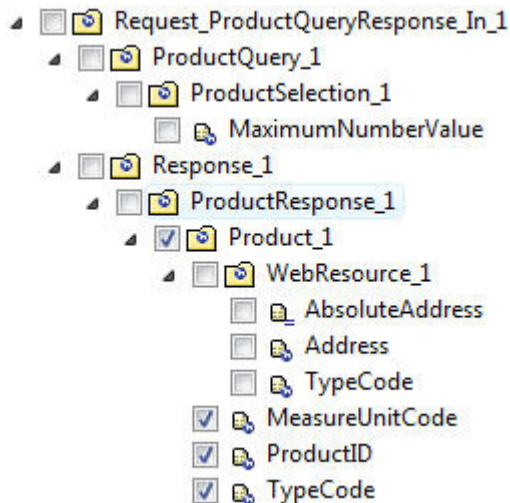
String fqUrl = cuttedUrl + element.getAddress();
return fqUrl;

```

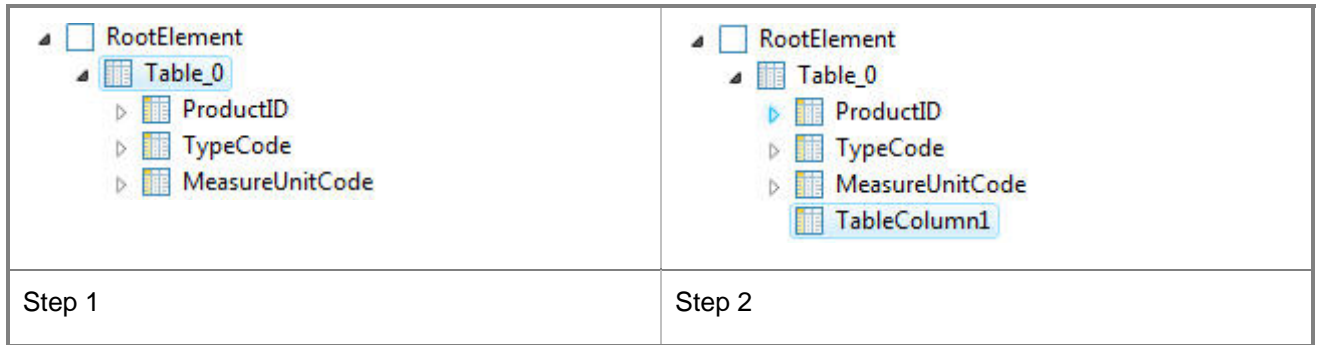
- Go to the wdDoInIt() method and add the following code to the end of this methods body:  
executeProductQueryResponse\_In();
- To import the missing classes just perform keystroke **CTRL+SHIFT+O**

#### Create the Layout

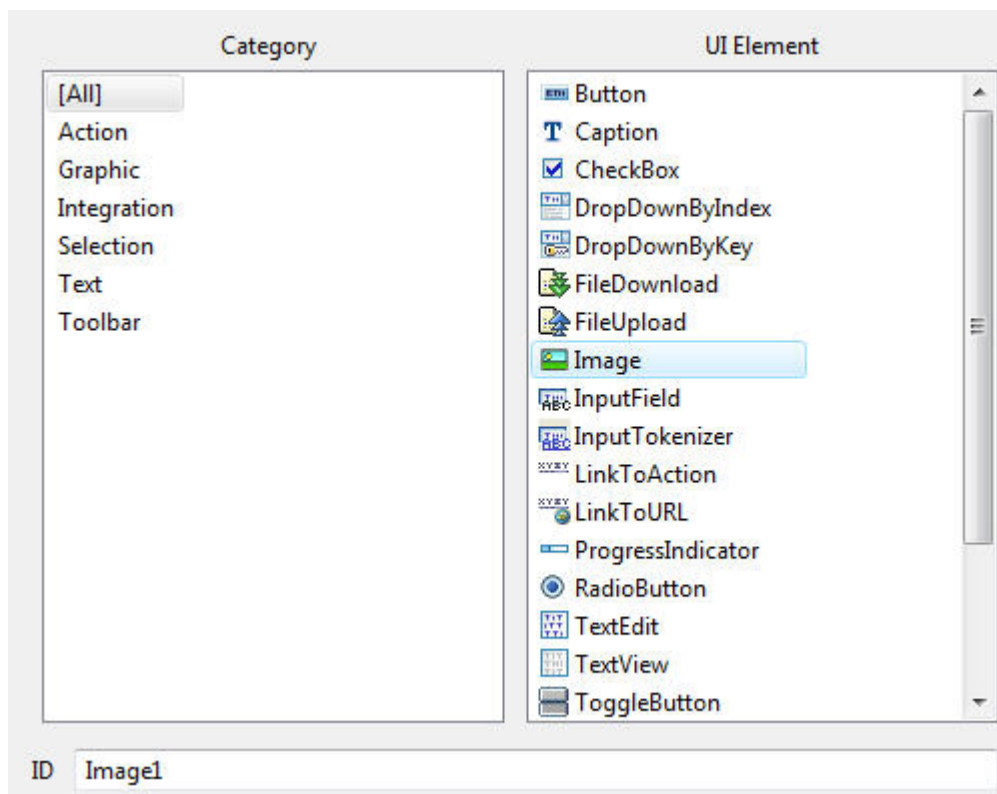
- Now we can switch back to the **"Layout"**-Tab. Then right click on the **"RootElement"** within the **"Outline"** view, select **"Apply Template"** and choose **Table**.
- Select the attributes and nodes from the context, which are shown on the next image.



3. Now right click on the Table entry within the **Outline** view as shown in Step 1 of the image below and select **AddGroupedColumn**.



4. Then right click on the newly created **TableColumn** entry and select **InsertTableCellEditor**. Select Image in the following menu as shown in the next screenshot.

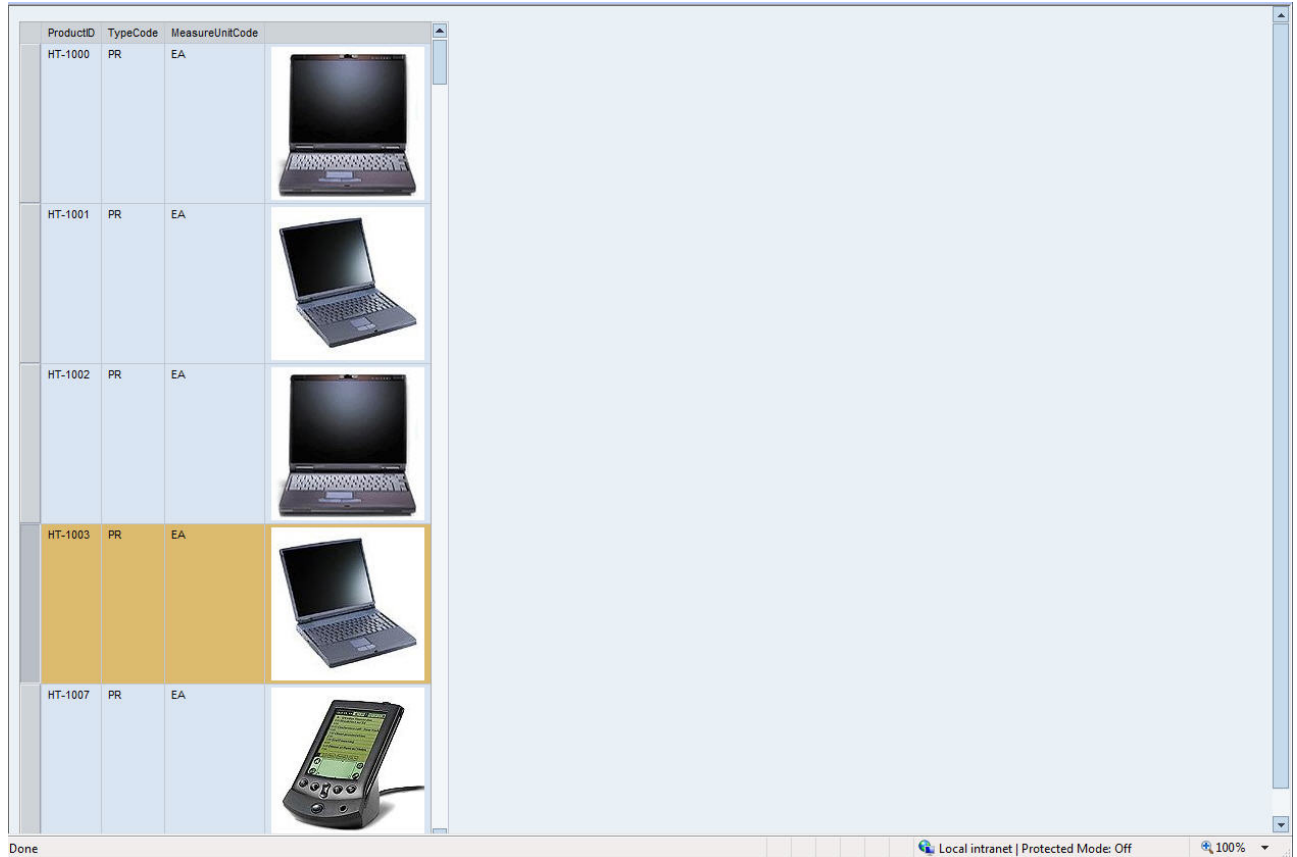


5. Now you only have to bind the **source** property of the newly created image object to the **AbsoluteAddress** attribute of the context.



## Create, Deploy and Launch the Application

1. Finally perform a right click on **“Applications”** within the WebDynpro navigation and select **“Create Application”**.
2. Right click on the created application and select **“Deploy new Archive and Run”**. After a few seconds your internet browser should show up showing the application. The result should look similar to the screenshot provided below.



3. If the browser doesn't start, just go to the following url:  
<http://<hostname>:<port>/webdynpro/dispatcher/demo.sap.com/modelimages/Products>

**Note:** This demonstration application assumes, that the used web service does not require authentication. If your application server is configured in a way, that authentication is required the table will stay empty. Then go to the NWA (<http://<host>:<port>/nwa> --> SOA --> Single Service Administration and remove the authentication.

## Related Content

[SAP Netweaver Demo Model](#)

[Description of the Demo Model Web Services JAVA](#)

For more information, visit the [Composition homepage](#).

For more information, visit the [User Interface Technology homepage](#).

## Copyright

© Copyright 2009 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, 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 S.A. in the United States and in other countries. Business Objects 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.