August 2004

# SAP Enterprise Portal Development Kit for Microsoft .NET 1.0 Beta

# (PDK for .NET)

Ofer Feldman, Software Engineer, SAP Labs Israel Tsufit Naor, Development Team Lead, .NET group, SAP Labs Israel Yossi Pik, Development Manager, .NET group, SAP Labs Israel Yoram Friedman, Product Definition, SAP Labs Israel Thomas Meigen, Project Director CTSC, SAP AG Daz Wilkin, SAP Program Manager, Microsoft Corp.

### **Executive Summary**

The objective of the SAP PDK for .NET is to enable customers to develop components for SAP Enterprise Portal, using the .NET framework and writing in .NET languages (VB.NET, C#). The development environment of the PDK is seamlessly integrated into Microsoft Visual Studio .NET 2003.

Components developed with the PDK for .NET can leverage functionalities of the SAP Enterprise Portal, including services and client-side events. Integration with the SAP .NET Connector ensures that communication with SAP backend systems, including mySAP ERP, is now easier then ever.

Customers can take full advantage of the Microsoft .NET framework (e.g. use Web Services); they can easily achieve integration with Microsoft servers, including Microsoft SQL Server and Microsoft Content Management Server (CMS), and reuse existing .NET/COM code.

# **Applies to**

- SAP Enterprise Portal 6.0 Service Pack 2 Patch 2 or Patch 3 (patch 4 is not supported)
- Microsoft Visual Studio .NET 2003
- Microsoft .NET Framework 1.1

### **Keywords**

PDK for .NET , Visual Basic :NET, Visual C#, SAP Enterprise Portal, iView

### Level of difficulty

Technical consultants, Developers



Copyright 2004 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, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation 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.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, 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 in several other countries all over the world. 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.



# Contents

Executive Summary	1
Applies to	1
Keywords	1
Level of difficulty	1
Contents	3
Business Overview	4
Technical Overview	5
Architecture The SAP .NET Runtime Engine The Interoperability Framework The Runtime Process Flow The Visual Studio .NET Add-in The SAP .NET Connector	6 6 7 8 10
Development Process Portal Component Portal Services SAP NetWeaver Controls Deployment	11 
Step-by-Step Development Example	13
System Requirements SAP .NET Runtime Engine 1.0 SAP Portal Development Kit for .NET 1.0	
Availability	19
Summary	19



### **Business Overview**

The Microsoft .NET platform is one of the two most common platforms for developing distributed applications. Organizations that adopted Microsoft .NET have already invested a large amount of resources in developer training and migration from other platforms.

The Microsoft development environment, Visual Studio .NET, is the most common development environment today, with a committed developer community focused on rapid development using C# and VB.NET. Adoption of the SAP PDK for .NET by .NET developers as the standard way for developing .NET based applications for SAP Enterprise Portal can benefit many SAP-centric organizations.

To increase productivity, the .NET environment offers a unified development environment and a standard set of tools common to all languages and developer tasks. A .NET programmer experienced in user interface development of a corporate Web site is able to use the same set of skills when creating self service applications, Web services, or any other Web based application. Using the SAP PDK for .NET, the development, testing, and deployment of SAP portal components becomes as easy as developing any other .NET Web application.

Preserving current knowledge and skill sets is an important goal for any organization. Succeeding in this lowers TCO significantly and helps an organization become more agile and competitive.

Since developing portal components (the "template" from which iViews are created) is a frequent task for the portal developer, the SAP PDK for .NET enables .NET developers to leverage their existing knowledge and be as productive when creating portal components as they are when creating any other .NET components.



### **Technical Overview**

The PDK for .NET allows Microsoft Visual Studio .NET developers to build portal components for SAP Enterprise Portal. Once these portal components are added to portal pages by a content administrator they are called iViews. Therefore, for the purpose of this document, .NET portal components will be referred to as .NET iViews.

The process of creating iView projects is similar to that for ASP.NET projects. The SAP .NET Connector is leveraged by the PDK for .NET and provides access to the complete library of SAP business objects. SAP provides a library of SAP-specific .NET controls, including data entry fields, buttons, and data-grids, all of which may be used to ensure a consistent look and feel across .NET iViews and all other iViews, as well as fully supporting portal themes. Importantly, .NET iViews are able to draw upon the strengths of the .NET Framework and ASP.NET, and a key subset of SAP Enterprise Portal services. iViews may be deployed directly from Visual Studio to the portal. In summary, SAP customers using SAP Enterprise Portal and the SAP Portal Development Kit for .NET can make full use of SAP business processes and SAP NetWeaver technology while using Microsoft .NET based technologies and solutions.

.NET iViews may be written using any .NET language. SAP provides full language support for C# and VB.NET. The PDK for .NET exposes all the functionality that a .NET developer needs to develop SAP portal components efficiently, including:

- Portal archive project, a new type of project dedicated to creating a group of portal components, is available from the projects list
- Listing of available SAP Enterprise Portal systems in the Visual Studio Server Explorer
- SAP Unified Rendering, ensuring that iViews built with the .NET controls provide a consistent look and feel across the portal
- Access to SAP Enterprise Portal services through .NET
- Deployment of iViews to the portal directly from the Visual Studio SAP Menu or Solution Explorer
- Launching and debugging iViews from within Visual Studio



# Architecture

The SAP PDK for .NET is comprised of these components: the SAP .NET Runtime Engine, an Interoperability Framework, a Visual Studio .NET Add-in, and the SAP .NET Connector 2.0.



Figure 1 PDK Architecture

### The SAP .NET Runtime Engine

To run .NET iViews created with the SAP .NET PDK, SAP Enterprise Portal uses a runtime engine called the SAP .NET Runtime Engine. The SAP Enterprise Portal acts as the only web server, therefore IIS is neither needed nor used. The SAP .NET Runtime Engine utilizes the ASP.NET Engine to process .NET iViews at runtime. All ASP.NET objects (request, response, and session), major portal objects (iView Profile, user context) and major portal services (Systems Landscape Service, User Management Service) are exposed to iView developers.

The SAP .NET Runtime Engine can run either as an NT service or as a console application. SAP Enterprise Portal has a special Java iView and Java portal service responsible for communication with the SAP .NET Runtime Engine.

#### The Interoperability Framework

The Interoperability Framework allows API calls between the Java and .NET stacks. The Interoperability Framework enables .NET developers to call Java-based portal services, and enables SAP Enterprise Portal to pass requests to the SAP .NET Runtime Engine. TCP/IP is the underlying communication protocol between the two components (Enterprise Portal and .NET Runtime Engine). On the .NET side, .NET Remoting is used.



### **The Runtime Process Flow**



Figure 2 The Runtime Process Flow

- 1. SAP Enterprise Portal receives a request for a .NET iView.
- 2. The portal dispatches the request to the Java "Interoperability" iView responsible for communication with the SAP .NET Runtime Engine.
- 3. The interoperability framework passes the request to the SAP .NET Runtime Engine.
- 4. The SAP .NET Runtime Engine executes the appropriate .NET iView.
- 5. The SAP .NET Controls are rendered by the SAP .NET Runtime Engine, using the portal theme and other portal parameters.
- 6. The iView may call one or more portal services. Communication back to SAP Enterprise Portal services is accomplished through the interoperability framework.
- 7. The SAP .NET Connector connects to backend SAP systems and retrieves or updates data. The connection information is provided by the portal.
- 8. The response from the .NET iView is returned to SAP Enterprise Portal through the interoperability framework.
- 9. The result is returned to the user's browser.



#### The Visual Studio .NET Add-in

The PDK for .NET provides, within Visual Studio, everything that a developer needs in order to develop, deploy, and test iViews. Developing iViews allows a developer to use the same set of skills acquired developing other .NET applications.

The Visual Studio Add-in allows .NET developers to build and write iViews visually, without requiring a running portal. A running portal is only needed for previewing and debugging iViews, and this can be done using a local or remote portal.

The SAP .NET Portal Development Kit 1.0 Add-in extends the functionality of Visual Studio 2003, creating a development environment for iView developers.

Among the new functionalities which can be found:

#### 1. A new SAP Portal Archive project

New Project				×
Project Types:	Templates:			:
Visual Basic Projects Visual C# Projects Visual J# Projects Visual J# Projects Visual C++ Projects	SAP Portal Archive	Windows Cl Application	ass Library	
Setup and Deployment Projects     Other Projects     Visual Studio Solutions	<b>1</b>	<b>F</b>		
	Control Library	mart Device AS Application A	P.NET Web opplication	
A project for creating SAP portal archive				
Name: PortalArchive1				
Location: D:\My Documents\	Visual Studio Projects\	•	Browse	
Project will be created at D:\My Documents\	Visual Studio Projects\Portal	Archive1.		
<b>¥</b> Mor <u>e</u>	ОК	Cancel	Help	





2. Enhancement of the Solution Explorer menu



#### 3. An SAP Option dialog

Dptions  Environment  SAP  Portal  Source Control  Text Editor  DevPartner  Database Tools  Debugging Device Tools  HTML Designer  Vindows Forms Designer  XML Designer	General PAR Settings General PAR Settings Include source files Include debug information files PAR Location Current project directory Custom directory C:\SAP Browse Doverwrite existing PAR 'Deploy' Behavior Display deployment progress bar Always deploy before 'View in Browser'
	OK Cancel Help



4. New nodes in the Server Explorer



5. An SAP Menu and Toolbar



#### The SAP .NET Connector

The SAP .NET Connector 2.0 connects .NET applications with business processes running in SAP systems, including ERP (R/3), CRM, and BW. The connector exposes the entire SAP business API through either SAP's binary transport protocol (RFC) or via Web Services. The connection information to the backend system can be retrieved automatically from the portal system landscape (a storage of systems connection information) and from the user management repository, thus applying the benefits of administrative system management, user mapping and single sign-on. In conclusion, developers using the SAP .NET Connector within the PDK for .NET can build .NET iViews that interact seamlessly with back-end SAP systems.



### **Development Process**

iView development begins with the creation of a new SAP Portal Archive project in Visual Studio. Portal services are exposed by the PDK to be used as necessary. The SAP NetWeaver controls are available through the Visual Studio Toolbox, and ensure that the theme of the iView user interface will be consistent with the other iViews running in the portal. Access to SAP systems data is provided by the SAP .NET Connector that receives relevant connection information through the PDK from the portal. Once the development of an iView is finished, the iView can be deployed directly to the portal from within Visual Studio. Developers can choose to view and even debug the iView directly from Visual Studio.

### **Portal Component**

One way to add content to the portal is by developing .NET iViews. The development of .NET iViews is similar to the development of ASP.NET Pages. A new class,

SAP.Portal.Web.UI.PortalComponent, is introduced for this purpose. All developed iViews should inherit from it in order to be displayed in the portal and to obtain the expected functionality of such components.

.NET attributes are used to define iView properties. The deployment process converts these attributes to iView properties.

Context objects are available through the properties of the portal component class, such as:

- The Profile object provides runtime information about the instance of the iView that is currently running
- Logger a logging object providing a mechanism for logging errors and messages in the portal
- Other information from the portal User information, Paths to resources, etc.

#### **Portal Services**

The SAP PDK for .NET gives the developer the ability to use Java portal services from the .NET environment. The exposed services are available through the interoperability framework.

The following portal services are supported in the current version of the PDK:

- System Landscape a portal service for accessing the portal system landscape. It contains classes used to get connection information for all the systems that are accessed from the portal, and to determine the user credentials that are required to access this system
- User management a portal service for accessing the User Management Engine (UME), which provides basic user management functionality

#### SAP NetWeaver Controls

SAP PDK provides a set of .NET controls that may be used as the building blocks of an iView's UI. These SAP NetWeaver controls are available through the Visual Studio Toolbox. They support the same features as standard controls, but there are some differences (e.g. property names) and additions. They use SAP's unified rendering (UR) to render and support all SAP-supported browsers. They may be bound to data sources including tables retrieved from SAP backend systems through the SAP .NET connector.



Toolbox <b>P</b> ×	Start Page	PortalComp	onent1.ascx	
SAP NetWeaver	Label InputF	field	Button	
Revinter				
🐮 Button	checkBox2	Enter text he	re	
🔧 CheckBox		1		
CheckBoxGroup	CHECKDOX4			
TopDownByKey			E	
🎦 InputField			<u>kToAction</u> Link	ToUrl
于 Label	na ListDauthourd			
🔏 LinkToAction	listBoxitem1			
🐱 LinkToURL	listBoxItem3	<b>F</b>		
🗾 ListBox	listBoxItem4	Šìm	ple Text Area	
👫 Table	E			
TextEdit	lable			
TextView	Column0	Column1	Column2	
	abc	abc	abc	
	abc	abc	abc	
	abc	abc	abc	
	abc	abc	abc	
	abc	abc	abc	
SAP Proxy			Page 1 / 1	ſ
My Liser Controls				

This screenshot shows the SAP NetWeaver controls in the toolbox and also when added to the iView

#### Deployment

.NET iViews are packed in Portal Archive (PAR) files. The structure of PAR files is the same for both .NET iViews and Java iViews, so the existing portal deployment mechanism is also used to deploy .NET PAR files. The assemblies and the files are kept in a repository called the Portal Content Directory (PCD), and they are copied locally to each node of the cluster on demand. The same runtime directory structure is used for both Java and .NET PARs. The SAP PDK for .NET enables the developer to deploy to a remote server from within Visual Studio.



## **Step-by-Step Development Example**

This section demonstrates the development process in the PDK for .NET. It shows the various development actions a developer will encounter using the PDK. This example is of an iView that lets the user search for a list of books according to author name. The information will be retrieved from Microsoft SQL Server, from the Pubs database.

The steps the developer performs:

1. Create a new SAP Portal Archive project.

<b>96</b> M	licrosoft Development Enviror	nment [design] - Object Browser	New Project	The second s		×
Eile	Edit <u>View Build T</u> ools <u>1</u> New	Window Help	Project Types:		Templates:	
	Open ► ⊆lose	Elle Ctrl+N	Visual Ba:	sic Projects Projects Projects	SAP Portal Windows	Class Library
~~	Add Project	✓ <u>C</u> ustomize	Visual C+     Setup an	+ Projects d Deployment Projects	Archive Application	
80°	Opgn Solution Close Solution	Empty	Visual Stu	ojects Idio Solutions	Windows Smart Devic Control Library Application	e ASP.NET Web
			A project for crea	ating SAP portal archive		
			Name:	PortalArchive1		
			Location:	C:\Development\test p	rojects	Browse
			Project will be crea	ated at C:\Development\test p	rojects\PortalArchive1.	
					OK Cancel	Help

2. Add a new Portal Component (iView).

Add New Item - PortalArchive1		×		plorer - PortalArchive1	4 ×
Categories:	Templates:	000 5-5- 000 5-5-	Soluti	n 'PortalArchive1' (1 project)	
UI Code	SAP Portal SAP			🕮 Build R <u>e</u> build	
Web	Component Connec			Deploy current project	
SAP SAP			Add New Item	Add	,
			Add Existing Item	Add <u>R</u> eference Add W <u>e</u> b Reference	
			Add Component	Set as StartUp Project Debug	,
A class for creating SAP portal cor	nponents using the visual designer		-	Save PortalArchive1	
Name: PortalComponent2.as	spx				
	Open Cancel	Help			

3. Using two SQL Data Adapters, create two queries to the database. One returns an authors table and the other returns a books table (using the parameter "author name"). This is all done easily with the Data Adapter connection wizard and the query builder wizard.



4. Finally, create a typed Dataset, based on these two SQL Data Adapters, and add an instance of it to the portal component.

PortalComponent2.asp	к			$\triangleleft \triangleright \times$
The pa To use abso	age you are working on is ir top-to-bottom as lute (x and y) positioning, c	i flow layout mode, and o in a word processing doo hange the pageLayout p GridLayout.	objects will be arranged cument. property of the DOCUMENT to	
📽 sqlDataAdapter1	🐮 sqlDataAdapter2	🥰 sqlConnection1	🖗 PubsData5et	
Le Design 🛛 🗠 HTML				

5. Now drag some SAP NetWeaver controls from the toolbox to the portal component to create the iView's GUI.





PortalComp	onent2.aspx			4 Þ ×
) Author Name	System.Data.Da	ataRc 💌 Search		
Titles				
title	type	price		
abc	abc	0		
abc	abc	0.1		
abc	abc	0.2		
abc	abc	0.3		
abc	abc	0.4		
	]	Page 1 / 4		
Nore: <i>TextVi</i> e	em.			
🎇 sqlData	aAdapter1 ataSet	🎇 sqlDataAdapter2	🥞 sqlConnection1	
G Design	🖸 HTML			

6. Set the control names and properties to finish the GUI.

7. Now bind the controls to the Dataset, using the Data properties. This will ensure that the iView will display the data with nearly no code needed to be written. Bind the dropdown control to the authors table in the Dataset, and the Table to the titles table.

Properties	<b>4</b> ×
Table1 SAP.Web.UI	.Controls.Table 🔹
🗄 🛃 🔳 🗲   [	
Behavior	
EnableViewState	True
🖻 Data	
(DataBindings)	
DataMember	
DataSource	
🛛 Design	PubsDataSet
Design	JIANDARD
FillUpEmptyRows	True
FirstVisibleRow	0
FooterVisible	True

8. Now add some code to handle user events, such as Page load, Button click, table row select (called LeadSelect).



PortalComponent2.aspx.cs			4 Þ ×
Contail Archive 1. Portal Component 2	•	📌 sqlDataAdapter1	
<pre>{     {         /// <summary>         /// Summary&gt;         /// Summary description for Por         /// </summary>         [PortalComponentClassProperty("         [PortalComponentAttribute("Port         public class PortalComponent2 :         {</pre>	stalComponent2. "DemoName","Demo salComponent2"," : SAP.Portal.Web d code	Value", UserPersonalization PortalComponent2.aspx")] .UI.PortalComponent	= PersonalizationType.NoD
<pre>private void Page_Load(obje {     if (!Page.IsPostBack)     {         sqlDataAdapter1.Fil         Session["Data"] = d     }     else     {         dsBooks1 = (dsBooks     }         this.DataBind();</pre>	ect sender, Syst ll(dsBooks1.auth dsBooks1; s)Session["Data"	em.EventArgs e) ors); ];	
<pre>-</pre>	(object sender, ; Command.Paramete sBooks1.titles; lect(object send 1.titles[e.Row][	SAP.Web.UI.Controls.Abstrac rs["@au_id"].Value = drpAut er, SAP.Web.UI.Controls.Tab dsBooks1.titles.notesColumn	tButton.ActionEventArgs e) hor.SelectedKey.ToString() le.LeadSelectEventArgs e) .ColumnName].ToString();

9. The iView is ready for testing. Define a portal to which to deploy. For this, use the Server Explorer window, and add a new portal under the "Portals" node (This needs to be done once, and this defined portal can be used in later sessions).

🚰 Data Connections		14	
	Display name	LocalPortal	
Application Servers	Host	localhost	
Refresh	Port	50300	
a Ser X Delete	User	testadmin	
Add Portal	Password	*****	Remember password
	Portal response timeout	15	seconds
	Use SSL protocol		

10. Set his portal as "Current". This indicates that this is the portal the developer means to work with.





11. Deploy the project. (One possible way is to do it from the Solution Explorer window.)



12. Now you can view the iView in the portal. From here, you can see the running iView in the portal.





# **System Requirements**

### SAP .NET Runtime Engine 1.0

The beta version of the PDK for .NET requires one instance of the SAP .NET Runtime Engine for each instance of SAP Enterprise Portal. The SAP .NET Runtime Engine 1.0 is installed both as a console application and as an NT service. The system administrator can choose which mechanism to run in order to instantiate the SAP .NET Runtime Engine.

System Requirements

- Operating system of Windows 2000 (Service Pack 4 or above) or Windows 2003
- SAP Enterprise Portal 6.0 Service Pack 2 Patch 2 or Patch 3 (patch 4 is not supported)
- SAP Enterprise Portal 6.0 does not have to be on same machine as the SAP .NET Runtime Engine 1.0
- Microsoft .NET Framework 1.1

### SAP Portal Development Kit for .NET 1.0

Install the SAP .NET Portal Development Kit 1.0 on a development machine. The installation adds the SAP PDK for .NET Add-in to Visual Studio .NET. After the installation is completed, you can start developing SAP portal components.

System Requirements

- Operating system of Windows 2000 (Service Pack 4 or above), Windows 2003, or Windows XP
- Microsoft Visual Studio .NET 2003
- Microsoft .NET Framework 1.1
- SAP Enterprise Portal is needed for some development activities (see the "Development Process" section above)

**Note**: To take advantage of the SAP .NET Connector during design time, you need to install the SAP .NET Connector 2.0 Beta separately, on the development machine.



# Availability

SAP is running a beta ("early adopter") program for the SAP PDK for .NET for a limited number of customers.

SAP will leverage customer feedback and will use commercially reasonable efforts to make the first version of SAP PDK for .NET generally available for download.

### **Summary**

This paper describes the use of the SAP PDK for .NET. The PDK for .NET enables customers to use .NET languages (VB.NET, C#) to build iViews for SAP Enterprise Portal. The development environment is seamlessly integrated into Microsoft Visual Studio .NET 2003 and enables developers to leverage existing knowledge to reduce development time.

Developers using the PDK for .NET can leverage capabilities of the portal, including services and client side events. It is easy for developer to use the SAP .NET Connector to communicate with SAP backend systems. Because the development is done in .NET, developers can take full advantage of the Microsoft .NET framework, easily achieve integration with Microsoft Servers, and reuse existing .NET/COM code.

