

Applies To:

SAP NetWeaver 2004, SAP NetWeaver Portal SPS15 and above, Portal Development Kit 2.0 for Microsoft .NET (PDK 2.0 for .NET), Microsoft Visual Studio 2003

Summary

This article discusses various aspects of debugging .NET portal applications and provides an enhanced overview of the debugging options and tools, as well as detailed instructions on configuring and using them in PDK 2.0 for .NET. It complements the current PDK for .NET Developer's Guide.

By: Anne Tarnoruder

Company: SAP

Date: March 2006

Table of Contents

Introduction.....	2
Prerequisites	2
Set the Option to Include the .pdb files in the PAR	2
Local Debugging	2
Configure the Portal for Development Mode	2
Set Project Debugging Properties	2
Start Debugging.....	3
Remote Debugging	3
Set up Debugger Users	3
Set up Remote Debugging	3
Attach to the Portal Runtime for .NET Process	3
Start Debugging.....	3

Introduction

Debugging of the portal applications is similar to that of other Visual Studio applications – you set breakpoints, examine variables, and use all the tools provided by the [Visual Studio .NET Debugger](#). You can debug portal components as well as portal pages.

PDK for .NET enables two debugging modes: local and remote.

In the [local](#) mode, the portal components are processed by the local instance of *Portal Runtime for .NET* running on the development host. This allows multiple developers to debug at the same time without blocking each other. You would normally debug locally while developing your applications.

In the [remote](#) debugging mode, you attach the Visual Studio debugger a *Portal Runtime for .NET* process that runs on a remote server and processes your portal components. You can use this mode to troubleshoot your applications in a testing or production environment. In this mode, execution of any code, including that of other users, on this remote runtime process is halted while the debugger is on a break point.

Note This remote instance of *Portal Runtime for .NET* also processes previewing in the portal, performed with the **View in Portal** command and the **Preview** tab of Page Designer.

Prerequisites

Set the Option to Include the .pdb files in the PAR

To enable debugging, the *.pdb* files must be included in the project Portal Archive (PAR) before deployment to the portal. You set this option either in the *SAP Options* or *Project PAR Properties* dialog box.

Local Debugging

To debug locally, you perform the following tasks:

Configure the Portal for Development Mode

For local debugging, set the portal-side **Production Mode** property in the current portal to *false*. For more information on configuring portal-side settings, see [Installation, Upgrade and Configuration Guide](#).

Set Project Debugging Properties

When a new project is created, the project debugging properties are set automatically. However, if the project was created in the previous versions of PDK for .NET, you should set these properties manually.

1. Select the project in Solution Explorer, then on the Project menu, click **Property Pages**.
2. In the Property Pages dialog box, click **Start Options**, and set the properties as follows:
 - o **Start external program:** <Portal Add-In Installation Folder>\Debug\ep.exe (by default, C:\Program Files\SAP\Portal Add-in for Visual Studio 2003\Debug\ep.exe)
 - o **Command line arguments:** <port number for runtime> (by default, 8060)

Important! This port number is used by the instance of *Portal Runtime for .NET* that is running on your development host for the needs of debugging. Make sure this port is accessible (not blocked by a firewall) and not used by another application running on the same host. Otherwise, an error message "Mutex could not be created" appears, and debugging is stopped.

Note If you set the port number for runtime in the *SAP Options* dialog box, this setting will be the default for all new SAP Portal Application projects. You can change it in any project, as described above.

Start Debugging

1. In Solution Explorer, right-click the project that you want to debug, and select **Set as StartUp Project**.
2. Right-click the portal component or the page that you want to debug, and select **Set as StartUp Component**.
3. Set the breakpoints in the portal component code.
4. Start debugging by pressing **F5**.

You do not need to deploy your portal application to the portal before debugging, this is done automatically.

Warning The **Start Without Debugging** command is not supported in SAP Portal Application projects. If invoked, the **ep.exe** process starts and not terminates automatically, so you will need to end it manually in Windows Task Manager. Avoid using this command.

Remote Debugging

To debug remotely, you perform the following tasks:

Set up Debugger Users

You must have the necessary permissions to debug. For instruction on setting up debugger users, see [Adding Debugger Users and Configuring DCOM](#).

Set up Remote Debugging

Visual Studio remote debugging components must be installed on the portal server. For more information, see [Setting Up Remote Debugging](#).

Attach to the Portal Runtime for .NET Process

1. On the **Debug** menu, select **Processes...**
2. In the **Processes** dialog box:
 - a. Select **Default** in the **Transport** list box.
 - b. In the **Name** list box, specify the remote machine that you intend to use for debugging.
 - c. Select the **Show system processes** check box.
 - d. Select the **Epsrv.exe** process and click **Attach**.
 - e. In the **Attach to Process** dialog box, select **Common Language Runtime**, and click **OK**.
3. Click **Close**.

Start Debugging

1. Set the breakpoints in the portal component code.
2. In Solution Explorer, right-click the portal component or the page that you want to debug, and select **View in Portal**, or otherwise access it in the portal.

Author Bio



Anne Tarnoruder is an information developer for SAP NetWeaver Product Management in SAP Labs, Israel, where she is responsible for the developer's documentation of PDK for .NET and Visual Composer. Prior to joining SAP in 2004, she held a number of positions in software development, team management, system architecture and technical communications in various high-tech companies. Anne holds an M.S. degree in Applied Mathematics..

Disclaimer & Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.