Defining Windows Adapter Design Properties

Pega ACADEMY

Trademarks

For information about the third-party software that is delivered with the product, refer to the third-party license file on your installation media that is specific to your release.

Notices

This publication describes and/or represents products and services of Pegasystems Inc. It may contain trade secrets and proprietary information that are protected by various federal, state, and international laws, and distributed under licenses restricting their use, copying, modification, distribution, or transmittal in any form without prior written authorization of Pegasystems Inc.

This publication is current as of the date of publication only. Changes to the publication may be made from time to time at the discretion of Pegasystems Inc. This publication remains the property of Pegasystems Inc. and must be returned to it upon request. This publication does not imply any commitment to offer or deliver the products or services described herein.

This publication may include references to Pegasystems Inc. product features that have not been licensed by you or your company. If you have questions about whether a particular capability is included in your installation, please consult your Pegasystems Inc. services consultant.

Although Pegasystems Inc. strives for accuracy in its publications, any publication may contain inaccuracies or typographical errors, as well as technical inaccuracies. Pegasystems Inc. may make improvements and/or changes to the publication at any time.

Any references in this publication to non-Pegasystems websites are provided for convenience only and do not serve as an endorsement of these websites. The materials at these websites are not part of the material for Pegasystems products, and use of those websites is at your own risk.

Information concerning non-Pegasystems products was obtained from the suppliers of those products, their publications, or other publicly available sources. Address questions about non-Pegasystems products to the suppliers of those products.

This publication may contain examples used in daily business operations that include the names of people, companies, products, and other third-party publications. Such examples are fictitious and any similarity to the names or other data used by an actual business enterprise or individual is coincidental.

This information is the property of:

Pegasystems Inc.
One Rogers Street
Cambridge, MA 02142-1209
USA
Phone: (617) 374-9600
Fax: (617) 374-9620
www.pega.com

Chapter 4: DEFINING WINDOWS ADAPTER DESIGN PROPERTIES

This chapter provides an in-depth look at how to use the OpenSpan Studio Windows adapter to integrate Windows applications with your solution.

Building Blocks

When you complete this chapter, you should be able to:

- Identify the key Windows adapter properties.

This chapter includes the following topics:

Setting the Windows Adapter Properties

The first step in using a Windows application with your solution is setting the properties required to start the application. Windows adapter properties that apply to starting a Windows application include the following:

- Path
- TargetPath
- HookChildProcesses
- StartMethod
- StartOnProjectStart
- WorkingDirectory
- Arguments
- HideApplicationAtRuntime

Windows Adapter Properties

Path Property

If the application runs from a file in the same folder where all of the specific application's files are installed, enter in the Path property the full path to that folder, including the executable file name. For example:

```
C:\Program Files\OpenSpan\CRM Setup\CRM.exe
```

If the application is in a folder that is already set in the system path, you can enter the application file name without the full path. For example:

```
CRM.exe
```

Shortcuts may differ between the design environment and the Runtime environment. Make sure the entry used is compatible with the runtime systems.

The Path property does not support links (lnk), wildcards, or Regex text entries. If you are deploying solutions to desktops where the target application is running in different folders, you need to deploy different configurations of the solution or use the Folder property. The Folder property lets you select a system folder and file location for the Runtime solution.

To learn more about the Folder property, refer to OpenSpan Help.

TargetPath

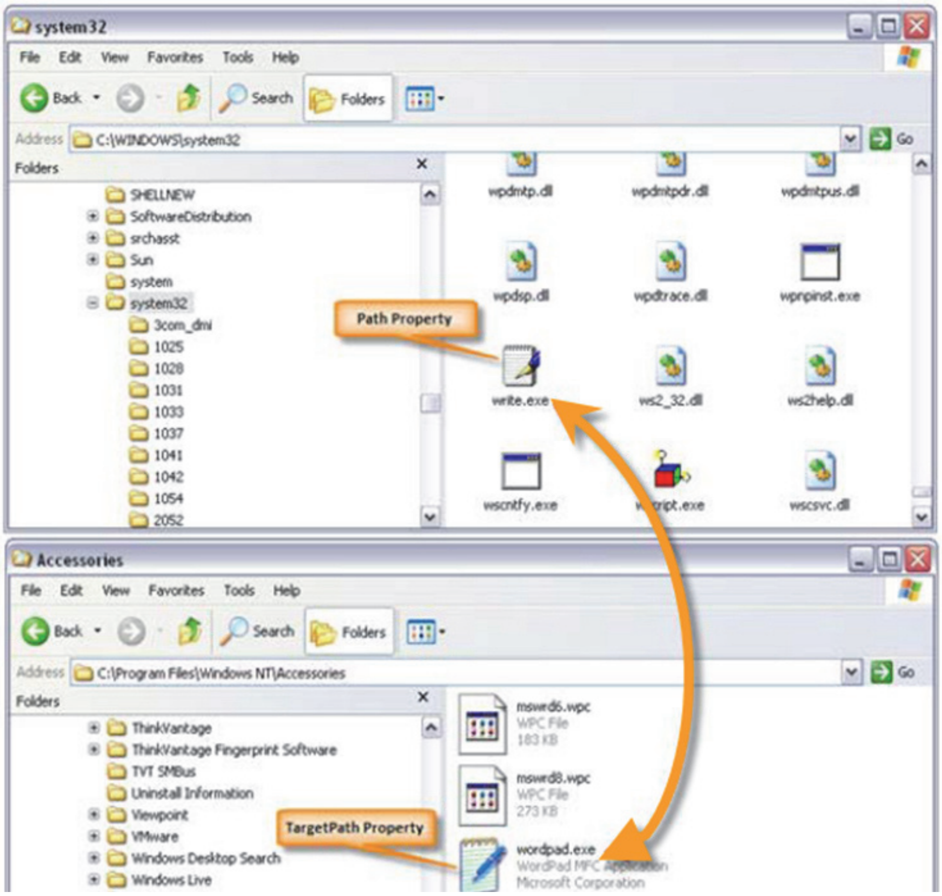
The TargetPath applies when the target application is launched by one or more other applications. The target application is launched as a result of one or more other processes (applications) occurring first. The Path property refers to an executable that starts the target application, but is not the target application itself. The TargetPath property sets the process name for the actual target application.

The TargetPath has a direct relationship with the StartMethod property — the TargetPath is only used when the StartMethod is set to StartandWait.

When you make entries in both the Path and TargetPath fields and set the StartMethod property to StartandWait, the application you enter in the Path property (or any process that it starts) cannot be interrogated. The only application you can interrogate is the application entered in the TargetPath.

In the following example, write.exe launches wordpad.exe, and wordpad.exe is interrogated. Accordingly, the Path property is write.exe (the application folder is in the system path), the TargetPath is wordpad.exe, and the StartMethod is set to StartandWait.

Note An example of when Path and Target Path properties are used is when working with batch files. The Path property defines the path of the executable batch files. The Target Path property is the result plus the interrogated application.



HookChildProcesses

The HookChildProcesses property applies when one application starts other applications and both parent and spawned applications are required for the solution. In this case, you have one application that launches another. That application may in turn launch another. Each of the applications require some interaction via an OpenSpan solution. Therefore, OpenSpan Studio must hook into each application process. Unlike where you are only interested in interacting with the final target process (TargetPath application), here you set the HookChildProcesses property to enable OpenSpan Studio to work with the starting application and each Windows application starting afterward.

An example of this is when a user launches a single sign-on application. Once the user signs on, each business application is signed into and launched. If the solution is interacting with each of the business applications, OpenSpan Studio must integrate with all of the application processes. To support this scenario, set the Path property to the path and file name of the application that initiates the launch process, and set the HookChildProcesses property to True.

StartMethod

The StartMethod property applies to the Windows adapter. This property determines how the Path application is started. The options are listed in the following table:

Option	Description
Start	Causes OpenSpan to launch and hook the application defined by the Path property when the adapter is started. This is the default.
StartAndWait	Causes OpenSpan to launch the application identified in the Path property when the adapter is started, wait until the application in the TargetPath is launched, and then hook the TargetPath application. Note: OpenSpan does not hook the Path application, so you cannot interrogate or automate the Path application.
MonitorAll	Causes OpenSpan to wait until the Path application is run and then hooks the application. In this case, starting the adapter does not cause the Path application to launch. The Path application must be launched independently after the adapter starts (for example, from an external application, or manually). Once the application is launched, OpenSpan hooks the application and the interrogated controls can be used in the project automations. When using the MonitorAll StartMethod, you can omit the full path when specifying the application executable in the Path property and enter the executable file name. Note: Using the StartMethod lets the adapter remain running if the Path application shuts down (due to an application error or the end-user closing the application). When the application is restarted, OpenSpan again hooks the application.

The adapter must be started before the Path application is launched or OpenSpan does not hook the application. The driver service must be used with the StartAndWait and the MonitorAll start methods when the StartMethod is not set to Start.

In cases where multiple instances of the Path application can be run, set the UseKeys property to True on the top level form in Object Explorer for the application. OpenSpan then correctly identifies and interacts with the various instances of the application.

Stopping the project stops all application instances with which OpenSpan is interacting.

StartOnProjectStart

The StartOnProjectStart property applies to all OpenSpan adapters. Use this property to indicate whether the adapter is started by OpenSpan when the project is started. The default is True, resulting in the adapter starting when the project is started. The adapter may or may not launch the associated Path or TargetPath applications. Starting the applications associated with the adapter depends on the StartMethod property.

Setting the adapter's StartOnProjectStart property to False makes sure the adapter does not start when the project is started. When set to False, the Adapter.Start method is required in the automation logic to start the adapter.

It is a good practice to set StartOnProjectStart to True only for the adapter applications that are required when the project starts.

When you set the StartOnProjectStart property to False, the Path and/or TargetPath applications cannot be launched when the project starts, regardless of the associated application StartMethods

Working Directory

OpenSpan Studio populates this field with the location of the working directory. In this case, the location is the same as the Path property.

Arguments

Use the Arguments property to enter command-line arguments required for starting your adapter. For example, you could enter a file name here so when a program launched via a shortcut, it opens that file. Also, use this property when working with Java applications to specify the Java class name.

HideApplicationAtRuntime

When set to True, the application is hidden during project runtime. To show the application, call the Show method on the adapter.

Note Set the HideApplicationAtRuntime property to True if you do not want the application shown when using the Runtime solution.