Chase Paymentech SDK
Merge Module Developer's Guide

V 1.2.1

2.20.13

TABLE OF CONTENTS

1. Summary

There are two ways to include the Orbital Gateway SDK with your product. The easiest way is to use the Merge Modules (discussed below). The other way is to add the files to your product and perform some install-time operations.

It is highly recommended to use the merge modules, if you can. They are designed to deploy the SDK correctly without requiring much effort for you. If your product uses Windows Installer, then read section “2. Merge Modules.” If your product does not use Windows Installer, then read section “3. Manual Redistribution.”

2. Merge Modules

Chase Paymentech has developed merge modules designed to assist third party software developers with the SDK integration into their Windows based install project. The developer will simply add the merge module to their project, set one property and the merge module will automatically install the SDK when the third party software is installed.

2.1. Requirements

- Chase Paymentech SDK
- InstallShield (or another Windows Installer development package)

Note: This module was built and tested with WiX, but is expected to work with InstallShield, WISE, or any other product that produces Windows Installer kits.

This document does not replace the standard SDK documentation. You will need the appropriate documentation for the SDK you plan to include.

2.2. Installation

Download the appropriate merge module from the Chase Paymentech web site. Copy the downloaded .msm file into your install development software’s merge module folder. With InstallShield 12, you can copy it to either:

- C:\InstallShield 12 Projects\MergeModules or
- C:\Program Files\Macrovision\IS12\Modules\i386

The former is recommended, as the latter folder is reserved for modules that were included with InstallShield.

Note: *The examples shown here are made using InstallShield 12. The procedure may differ slightly with different products.*

2.3. Adding the Merge Module to a Project

You must assign the install path to the merge module, so that it will install itself in a subdirectory off your product’s installation.

The following two steps describe how to add an SDK merge module to an InstallShield 12 installer. There are only two steps required for most of the merge modules. Refer to the section for your specific merge module for additional steps and details.

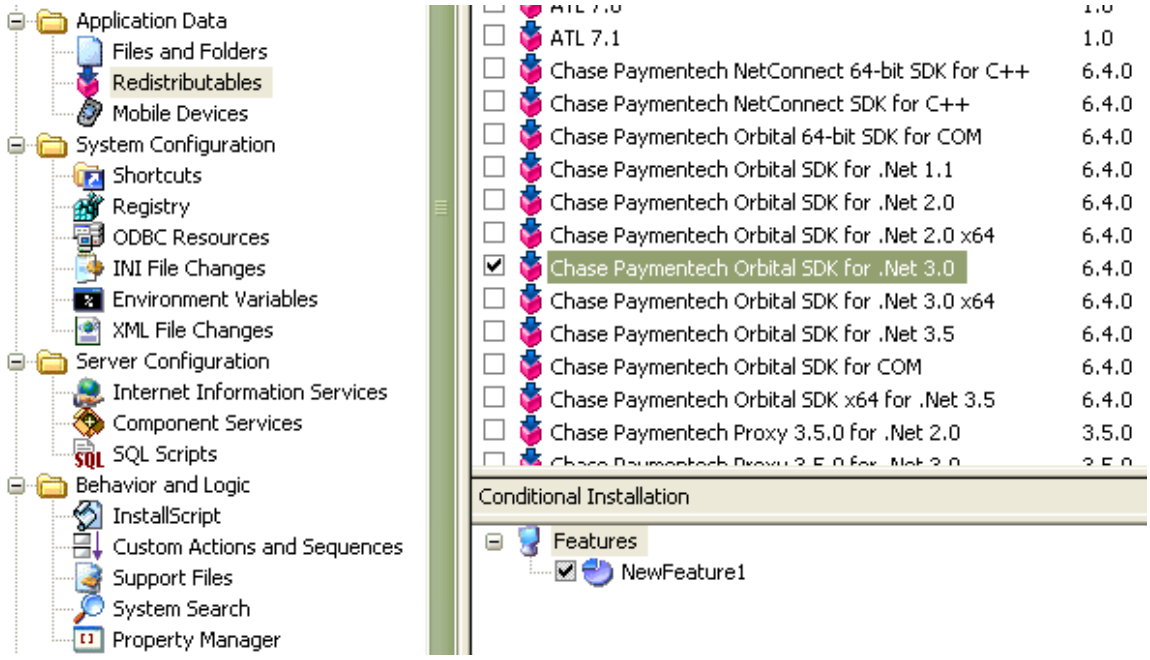
Step One: Add the Merge Module

To add the merge module click on the “Redistributables” item under Application Data and the right-hand pane will change to show the list of available merge modules.

Note: *If you had copied the SDK merge module to one of InstallShield’s merge module directories, it will appear in the list automatically*.

Put a check in the box beside the merge module you want to add.

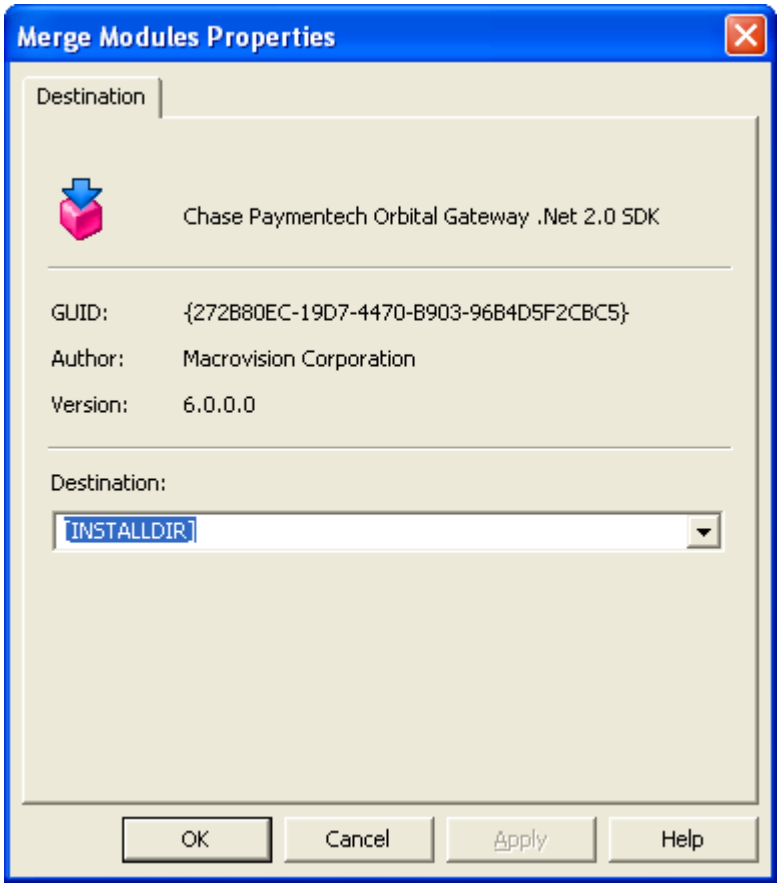
Under the “Conditional Installation” section check the Feature(s) where you want the merge module included.



Step Two: Set the Base Install Path

Right click on the merge module in the list, and then pick Properties on the pop-up menu. Pull down the Destination list and pick [INSTALLDIR]. This tells the install compiler to map your installer’s INSTALLDIR property to the INSTALLDIR property of the merge module.

Note: *This is required for the SDK to work properly. If you fail to set this, then the SDK is likely to be non-functional after installing.*



2.4. C++ Merge Modules (Orbital and NetConnect)

There are no special instructions or notes regarding this SDK.

A sample program is installed in the SDK's bin directory for your use to verify the SDK works properly.

2.5. .Net Merge Modules (Frameworks 2.0, 3.0 & 3.5)

The .Net merge modules come with a custom action that is required for installing the SDK's service. If you do not run this custom action, the SDK's service mode will not be available however DLL mode will function normally.

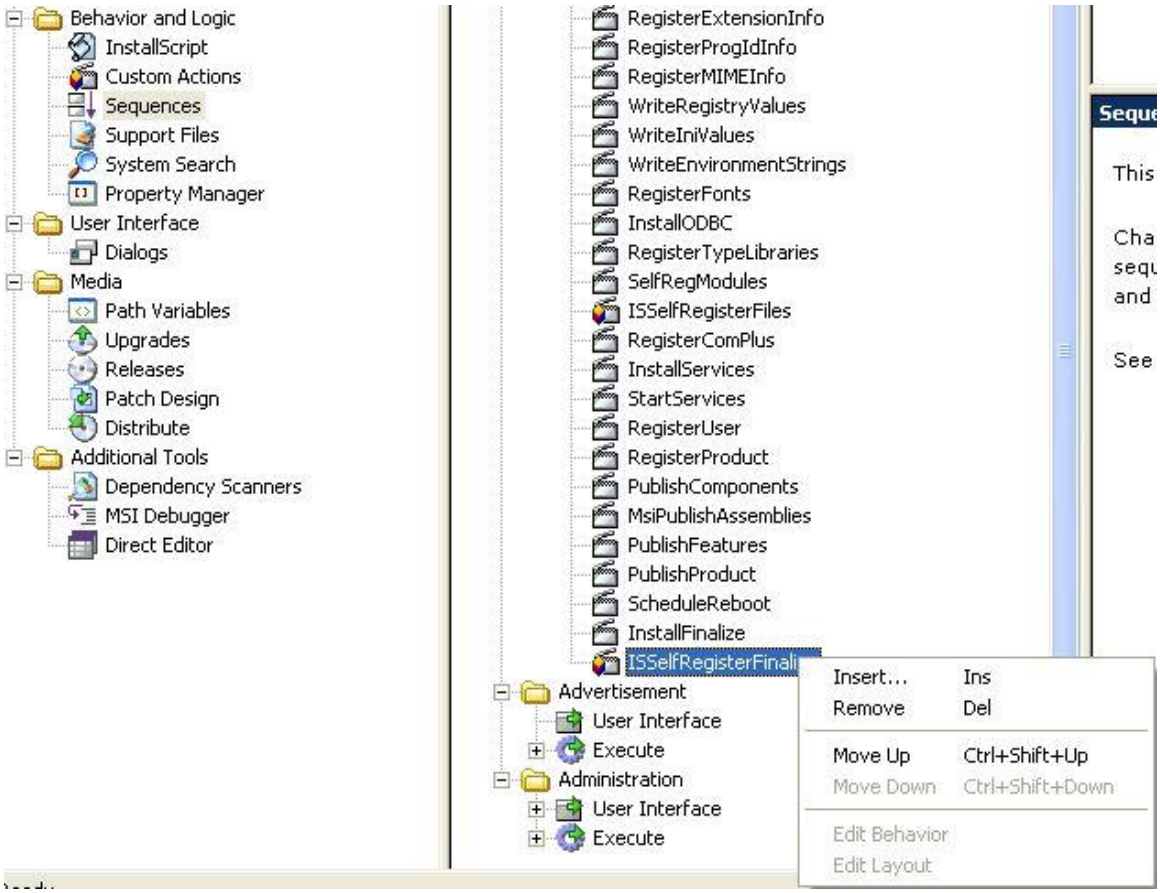
A sample program is installed in the SDK's bin directory for your use to verify the SDK works properly.

2.6. Configuration

Once the merge module has been installed and added to the project, you must decide how you want to handle the configuration of the SDK as it deploys. The SDK is configured by editing the linehandler.properties file. You can either create your own

custom action in the installer to automatically update the file with settings you define, or you can leave it alone, and require your client to manually edit the file.

If you elect to create a custom action to modify the file, it must be added to the install sequence so that it runs after the file has been successfully deployed. In the example below there is a Custom Action is being added after the “InstallFinalize” action has been executed.



3. Manual Redistribution

There are several steps that must be taken to properly install the SDK on a target system. For this reason, it is highly recommended that you use the Merge Modules, as described in section 2. However, if your product does not use Windows Installer as its deployment method, then you must use this means of installing the SDK.

Your installation process should follow these steps in order to ensure that the SDK works properly.

3.1. Preparation

Install the Orbital Gateway SDK on your system. You will need to take directories from that installation to build your own.

3.2. Copy Files

Your installation must create a directory tree that resembles the one created by the Orbital Gateway SDK installer. The following directories are required for the SDK to function:

etc
lib
logs
xml

You must create a directory to put these directories inside. This directory tree must have read/write permissions for the user that the application runs as.

Do **not** install these directories inside the InetPub directory on the target computer. IIS maintains rather tight security on that directory, and it may cause the SDK to fail to work.

3.3. Create Environment Variables

Create the following environment variables:

PAYMENTECH_HOME=C:\Paymentech\sdk\6.7.0
PAYMENTECH_LOGDIR=C:\Paymentech\sdk\6.7.0\logs
PATH= C:\Paymentech\sdk\6.7.0;%PATH%

Of course, you should replace the directory path to match the one you installed the SDK to.

NOTE: Setting the PATH is not necessary for .Net merge modules.

3.4. Add Assemblies to the GAC

Add all of the DLL files in the lib directory to the Global Assembly Cache (GAC). Do this using gacutil.exe, which comes with the .Net Framework.

The command line should look like this:

gacutil.exe -i Paymentech.dll

Do this for each DLL in the directory.

3.5. Register the Service

You must create the Paymentech Linehandler Service by executing the following:

installutil -i lib\PTDotNetService.exe

3.6. Reboot the Computer

This is usually necessary to ensure that the various registrations are accessible to all running applications, and to make sure that IIS and other applications can see the environment variables.