

IBM Platform Symphony 6.1.1 .NET4.0 API Patch with .NET4.5.2 runtime Readme File

About .NET4.0 API Patch with .NET4.5.2 runtime

This package contains the .NET4.0 API Patch with .NET4.5.2 runtime, which enables Platform Symphony clients and services to work with the assembly on the .NET4.5.2 runtime without any compatibility issue by default. Use this feature when you want to develop a .NET4.5.2 C# application for Platform Symphony.

Readme file for: IBM® Platform Symphony

Product/Component Release: Symphony 6.1.1

Update Name: Enhancement pack

Fix ID: sym-6.1.1-build374362

Publication date: 31 December 2015

Last modified date: 31 December 2015

1. Scope.....	3
2. Installation.....	4
1) Prerequisites.....	4
2) Installation files.....	4
3) Installation procedure.....	6
a. Before installation	6
b. Installation steps	6
c. After installation	8
d. Verification procedure	10
e. Uninstallation steps	12
3. Usage.....	14
1) How this feature works.....	14
2) Examples.....	14
4. Copyright and trademark information.....	16

1. Scope

Applicability	
Operating system	Windows Server 2012 64-bit, Windows 7 SP1 32-bit.
Symphony version	6.1.1
Cluster types	The feature applies to Platform Symphony cluster, DE and Client.
Other	Symphony .NET API
Dependencies	
File system	N/A
Limitations	
.NET targeted version	The C# clients and services should both target .NET 4.5.2
Known Issues	N/A

2. Installation

1) Prerequisites

- To apply this fix, you must have a Platform Symphony 6.1.1 Windows cluster, DE, or Client installed.
- This package must be installed on all development hosts, compute hosts, and client hosts.
- Visual Studio 2013/.NET 4.5.2 must be installed on development hosts when building the client and service that target .NET4.5.2.

Microsoft .NET Framework 4.5.2 and Microsoft Visual C++ 2013 Redistributable X86(for 32-bit client and service) or X64(for 64-bit client and service) must be installed on compute hosts and client hosts that do not have Visual Studio 2013 installed.

2) Installation files

The package includes the following files.

File name	Description
soam6.1.1_win-x86-374362.msp	The package that contains the new feature for Windows 32-bit Platform Symphony 6.1.1 cluster.
soam6.1.1_win-x64-374362.msp	The package that contains the new feature for Windows 64-bit Platform Symphony 6.1.1 cluster.
SymClnt6.1.1_win-x86-374362.msp	The package that contains the new feature for Windows 32-bit Platform Symphony 6.1.1 Client.
SymClnt6.1.1_win-x64-374362.msp	The package that contains the new feature for Windows 64-bit Platform Symphony 6.1.1 Client.
SymDE6.1.1_win-x86-374362.msp	The package that contains the new feature for Windows 32-bit Platform Symphony 6.1.1 DE.
SymDE6.1.1_win-x64-374362.msp	The package that contains the new feature for Windows 64-bit Platform Symphony 6.1.1 DE.
checksum.md5	The file that contains the MD5 checksum for soam6.1.1_win-x86-374362.msp, soam6.1.1_win-x64-374362.msp, SymClnt6.1.1_win-x86-374362.msp, SymClnt6.1.1_win-x64-374362.msp, SymDE6.1.1_win-x86-374362.msp, SymDE6.1.1_win-x64-374362.msp.

3) Installation procedure

a. Before installation

Disable the application using the following command:

```
> soamcontrol app disable all -f
```

b. Installation steps.

Note: Refer to the installation steps of the 64-bit packages when installing the 32-bit packages.

1) Copy the appropriate MSP packages of soam6.1.1_win-x64-374362.msp to compute hosts.

- **Interactive installation**

Double-click the Symphony MSP package to run the Symphony installer.

- **Silent installation:**

Run the Symphony installers from the command line using the following commands:

```
C:\>msiexec /update <Sym_package_name_path> /l*v <sym_install_log>  
/norestart /quiet REINSTALLMODE=omus
```

Where:

- sym_package_name_path is the fully qualified file name of the MSP package in this release.

- sym_install_log is the log file for the Symphony upgrade.

For example, to update a Windows 64-bit compute host, run the following commands:

```
C:\>msiexec /update C:\soam6.1.1_win-x64-374362.msp /l*v  
updateSym.log /norestart /quiet REINSTALLMODE=omus
```

2) Copy the appropriate MSP packages of SymDE6.1.1_win-x64-374362.msp to DE hosts.

- **Interactive installation**

Double-click the Symphony MSP package to run the Symphony installer.

- **Silent installation:**

Run the Symphony installers from the command line using the following commands:

```
C:\>msiexec /update <Sym_package_name_path> /l*v <sym_install_log>
/norestart /quiet REINSTALLMODE=omus
```

Where:

- sym_package_name_path is the fully qualified file name of the MSP package in this release.

- sym_install_log is the log file for the Symphony upgrade.

For example, to update a Windows 64-bit DE host, run the following commands:

```
C:\>msiexec /update C:\SymDE6.1.1_win-x64-374362.msp /l*v
updateSym.log /norestart /quiet REINSTALLMODE=omus
```

3) Copy the appropriate MSP packages of SymCInt6.1.1_win-x64-374362.msp to Client hosts.

- **Interactive installation**

Double-click the Symphony MSP package to run the Symphony installer.

- **Silent installation:**

Run the Symphony installers from the command line using the following commands:

```
C:\>msiexec /update <Sym_package_name_path> /l*v <sym_install_log>
/norestart /quiet REINSTALLMODE=omus
```

Where:

- sym_package_name_path is the fully qualified file name of the MSP package in this

release.

- sym_install_log is the log file for the Symphony upgrade.

For example, to update a Windows 64-bit client host, run the following commands:

```
C:\>msiexec /update C:\SymClnt6.1.1_win-x64-374362.msp /l*v  
updateSym.log /norestart /quiet REINSTALLMODE=omus
```

c. After installation

Deployment of assemblies on the development hosts, compute hosts, and Client hosts is required to enable the C# client and service applications to locate and reference them during runtime.

1) Register the .NET4.0 assemblies into the GAC

Register the new assemblies by running the 'gacutil /i' command.

Platform Symphony 6.1.1 compute host:

- **Windows 32-bit**

```
> gacutil.exe  
/i %SOAM_HOME%\6.1.1\win32-vc7\lib\Platform.Symphony.Soam.Net4.0.dll
```

- **Windows 64-bit**

```
> gacutil.exe  
  
/i %SOAM_HOME%\6.1.1\w2k3_x64-vc7-psdk\lib\Platform.Symphony.Soam.Net4.0.dll  
  
> gacutil.exe  
  
/i %SOAM_HOME%\6.1.1\w2k3_x64-vc7-psdk\lib64\Platform.Symphony.Soam.
```


Net4.0_64.dll

Platform Symphony 6.1.1 DE host:

- **Windows 32-bit**

```
> gacutil.exe
```

```
/i %SOAM_HOME%\6.1.1\win32-vc7\lib\Platform.Symphony.Soam.Net4.0.dll
```

- **Windows 64-bit**

```
> gacutil.exe
```

```
/i %SOAM_HOME%\6.1.1\w2k3_x64-vc7-psdk\lib\Platform.Symphony.Soam.Net4.0.dll
```

```
> gacutil.exe
```

```
/i %SOAM_HOME%\6.1.1\w2k3_x64-vc7-psdk\lib64\Platform.Symphony.Soam.Net4.0_64.dll
```

Platform Symphony 6.1.1 client host:

- **Windows 32-bit**

```
> gacutil.exe /i %SOAM_HOME%\lib\Platform.Symphony.Soam.Net4.0.dll
```

- **Windows 64-bit**

```
> gacutil.exe /i %SOAM_HOME%\lib\Platform.Symphony.Soam.Net4.0.dll
```

```
> gacutil.exe
```

```
/i %SOAM_HOME%\lib64\Platform.Symphony.Soam.Net4.0_64.dll
```

Note: The windows tool gacutil.exe must be prepared for registering assemblies.

2) Copy the assembly into the service package if you do not register the assembly with the GAC tool.

If you do not have the administrator role to install the assembly in the GAC, or you have no gacutil.exe tool installed, you can copy the assembly into the same directory with your client and deploy it in your service package as another dependent library of your service. You can refer to the C# tutorial in “Developer Edition Symphony 6.1.1 - Application Samples” for detailed steps on how to package the assembly with your service and deploy the package through Symphony CLI or GUI.

d. Verification procedure

1) File existence

Ensure the following files are in the following directories, as applicable, on all development, compute, and client hosts.

Platform Symphony 6.1.1 compute host:

- **Windows 32-bit**

Platform.Symphony.Soam.Net4.0.dll and Platform.Symphony.Soam.Net4.0.pdb should exist in %SOAM_HOME%\6.1.1\win32-vc7\lib\.

- **Windows 64-bit**

Platform.Symphony.Soam.Net4.0.dll and Platform.Symphony.Soam.Net4.0.pdb should exist in %SOAM_HOME%\6.1.1\w2k3_x64-vc7-psdk\lib\.

Platform.Symphony.Soam.Net4.0_64.dll and Platform.Symphony.Soam.Net4.0_64.pdb should exist in %SOAM_HOME%\6.1.1\w2k3_x64-vc7-psdk\lib64\.

Platform Symphony 6.1.1 DE host

- **Windows 32-bit**

Platform.Symphony.Soam.Net4.0.dll and Platform.Symphony.Soam.Net4.0.pdb should exist in %SOAM_HOME%\6.1.1\win32-vc7\lib\.

- **Windows 64-bit**

Platform.Symphony.Soam.Net4.0.dll and Platform.Symphony.Soam.Net4.0.pdb should exist in %SOAM_HOME%\6.1.1\w2k3_x64-vc7-psdk\lib\.

Platform.Symphony.Soam.Net4.0_64.dll and Platform.Symphony.Soam.Net4.0_64.pdb should exist in %SOAM_HOME%\6.1.1\w2k3_x64-vc7-psdk\lib64\.

Platform Symphony 6.1.1 Client host

- **Windows 32-bit**

Platform.Symphony.Soam.Net4.0.dll and Platform.Symphony.Soam.Net4.0.pdb should exist in %SOAM_HOME%\lib\.

- **Windows 64-bit**

Platform.Symphony.Soam.Net4.0.dll and Platform.Symphony.Soam.Net4.0.pdb should exist in %SOAM_HOME%\lib\.

Platform.Symphony.Soam.Net4.0_64.dll and Platform.Symphony.Soam.Net4.0_64.pdb should exist in %SOAM_HOME%\lib64\.

2) **Register the assembly with the GAC tool.**

If you have registered the assemblies in the GAC in the previous step, you should find the DLLs

installed in the following directories:

- **Windows 32-bit**

x:\Windows\Microsoft.NET\assembly\GAC_32\Platform.Symphony.Soam.Net4.0

- **Windows 64-bit**

x:\Windows\Microsoft.NET\assembly\GAC_32\Platform.Symphony.Soam.Net4.0

x:\Windows\Microsoft.NET\assembly\GAC_64\Platform.Symphony.Soam.Net4.0_64

e. Uninstallation steps

1) Disable related applications that are using the .NET4.0 DLL

```
> egosh user logon -u Admin -x Admin
```

```
> soamcontrol app disable XXX
```

2) Unregister the .NET4.0 assemblies

If you have registered the assemblies in the previous step, you should unregister them by running the following commands.

Platform Symphony 6.1.1 compute host:

- **Windows 32-bit**

```
> gacutil.exe /u Platform.Symphony.Soam.Net4.0
```

- **Windows 64-bit**

```
> gacutil.exe /u Platform.Symphony.Soam.Net4.0
```

```
> gacutil.exe /u Platform.Symphony.Soam.Net4.0_64
```

Platform Symphony 6.1.1 DE host:

- **Windows 32-bit**

```
> gacutil.exe /u Platform.Symphony.Soam.Net4.0
```

- **Windows 64-bit**

```
> gacutil.exe /u Platform.Symphony.Soam.Net4.0
```

```
> gacutil.exe /u Platform.Symphony.Soam.Net4.0_64
```

Platform Symphony 6.1.1 client host:

- **Windows 32-bit**

```
> gacutil.exe /u Platform.Symphony.Soam.Net4.0
```

- **Windows 64-bit**

```
> gacutil.exe /u Platform.Symphony.Soam.Net4.0
```

```
> gacutil.exe /u Platform.Symphony.Soam.Net4.0_64
```

3) **Uninstall the enhancement pack.**

You can use Microsoft Windows “Control Panel->Programs and Features->View installed updates” feature to roll back to a previous Symphony package on a Windows host.

You can also use the following command to roll back Symphony.

- Get the symphony package for sym6.1.1(for example, soam6.1.1_win-x86-374362.msp in this case).

- Use following command to rollback Symphony.

```
C:\>msiexec /uninstall <Sym_SP_path> /package <Sym_6.1.1_path>  
/norestart /quiet /l*v <Sym_rollback_log>
```

Where:

- Sym_SP_name_path is the fully qualified file name of the MSP package in this release.
- Sym_6.1.1_path is the fully qualified file name of the Symphony 6.1.1 msi package.
- Sym_rollback_log is the log file for the Symphony rollback.

3. Usage

1) How this feature works

The enhancement pack enables Platform Symphony clients and services to work with the assembly on the .NET4.5.2 runtime without any compatibility issue by default.

2) Examples

Case 1: Build C# client and service with the new assembly and run the application with .NET 4.5.2 runtime.

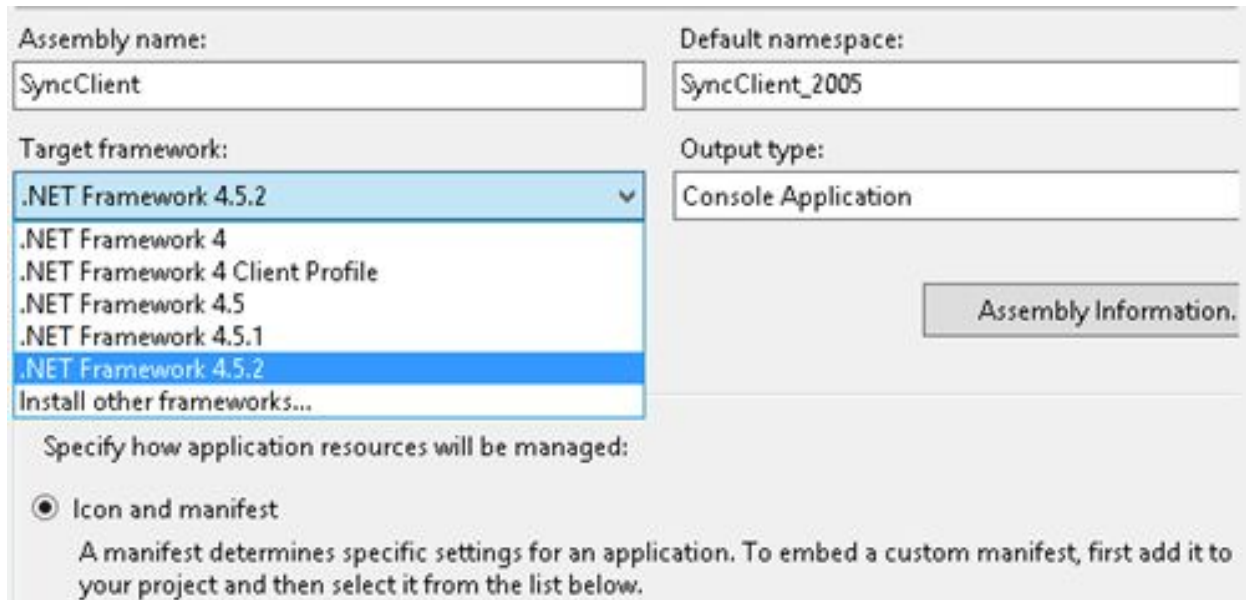
Summary

Build C# client and service with the new assembly and run the application with .NET 4.5.2 runtime.

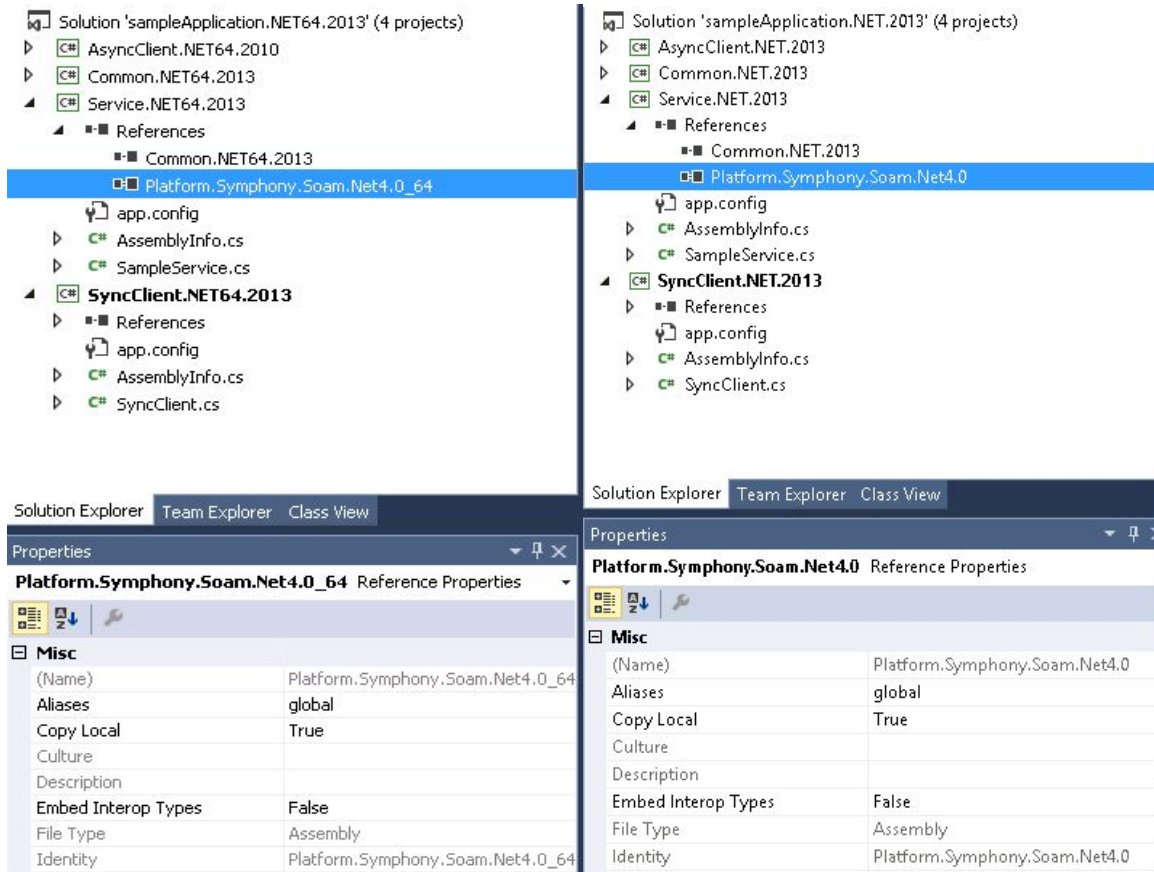
Pre-conditions:

1. The enhancement pack is installed on Compute/DE/Client hosts; .NET4.0 assemblies are registered by the GAC tool "gacutil.exe" on Compute/DE/Client hosts.
2. Build the Sample client and service with Visual Studio 2013.

- a. Open the Visual C# .NET solution file with Visual Studio 2013, Right click your project in Visual Studio Solution Explorer, and select the 'Properties' to enter the 'Application' property page. Set the 'Target framework' to '.NET Framework 4' or '.NET Framework 4.5.2'. The following screen capture shows where to select the target framework.



- b. Remove the references of the old assembly from the project, and add the new reference of the new assembly. For 32-bit applications shown on the right, you should add the reference to "Platform.Symphony.Soam.Net4.0.dll". For 64-bit applications shown on the left, you need to add the reference to 'Platform.Symphony.Soam.Net4.0_64.dll'. The following screen capture show where to remove and add the reference to 32-bit and 64-bit DLLs.



c. Build the .NET solution.

3. Deploy the service package.

Scenario:

Run the C# client program.

Post-Conditions:

Client finished successfully.

4. Copyright and trademark information

© Copyright IBM Corporation 2015

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP
Schedule Contract with IBM Corp.

IBM®, the IBM logo and ibm.com® are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.