



Australian Government

Australian Digital Health Agency



HIPS

Build Guide

7 November 2016

V6.1

Approved for external use

Australian Digital Health Agency

Level 25, 56 Pitt Street

Sydney, NSW 2000

Australia

www.digitalhealth.gov.au

Acknowledgements**Council of Australian Governments**

The Australian Digital Health Agency is jointly funded by the Australian Government and all state and territory governments.

HL7 International

This document includes excerpts of HL7™ International standards and other HL7 International material. HL7 International is the publisher and holder of copyright in the excerpts. The publication, reproduction and use of such excerpts is governed by the HL7 IP Policy (see <http://www.hl7.org/legal/ippolicy.cfm>) and the HL7 International License Agreement. HL7 and CDA are trademarks of Health Level Seven International and are registered with the United States Patent and Trademark Office.

Disclaimer

The Australian Digital Health Agency ("the Agency") makes the information and other material ("Information") in this document available in good faith but without any representation or warranty as to its accuracy or completeness. The Agency cannot accept any responsibility for the consequences of any use of the Information. As the Information is of a general nature only, it is up to any person using or relying on the Information to ensure that it is accurate, complete and suitable for the circumstances of its use.

Document control

This document is maintained in electronic form and is uncontrolled in printed form. It is the responsibility of the user to verify that this copy is the latest revision.

Copyright © 2016 Australian Digital Health Agency

This document contains information which is protected by copyright. All Rights Reserved. No part of this work may be reproduced or used in any form or by any means – graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems – without the permission of the Australian Digital Health Agency. All copies of this document must include the copyright and other information contained on this page.

Document information

Key information

Owner	Executive General Manager, Innovation & Development
Contact for enquiries	Australian Digital Health Agency Help Centre
t:	1300 901 001
e:	help@digitalhealth.gov.au

Product version history

Product version	Date	Release comments
1.0	February 2014	Initial release (HIPS 4.1.0).
2.0	February 2015	See release note (NEHTA-2040:2015) for details of changes and bug fixes.
2.0.1 2.0.2		Unpublished updates.
2.0.3	February 2016	See release note (NEHTA-2185:2016) for details of changes and bug fixes.
6.0	March 2016	See release note (NEHTA-2263:2016) for details of changes and bug fixes.
6.1	November 2016	See release note (DH-2445:2016) for details of changes and bug fixes.

Table of contents

1.	Introduction	5
1.1	Purpose	5
1.2	Scope.....	5
1.3	Assumptions	5
1.4	Definitions and Acronyms	5
2.	Products Overview	7
2.1	Source Code Software Package.....	7
3.	Build Process	9
3.1	Build Target Products	9
3.2	Build Tools and Environment	10
3.2.1	Core Web Services	10
3.2.2	User Interface Web Application	10
3.3	Build Target	11
3.3.1	Visual Studio	11
3.3.2	Mirth Connect.....	11
3.4	Dependencies.....	11
3.5	Build Steps	12
3.5.1	Debugging Uninstall Requirement	12
3.5.2	HIPS Core	12
3.5.3	P2P SMD.....	15
3.5.4	HIPS Web UI	18
4.	Build Test	21

1. Introduction

1.1 Purpose

The purpose of this document is to provide straightforward instructions for building the HIPS suite of products from the provided source code. If using the HIPS products suite the steps in this build guide must be completed prior to the steps within any of the installation guides.

The intended audience for this document is software developers and/or implementers of the HIPS product suite.

1.2 Scope

This document describes the steps and tools required to build the binaries and associated files required for each of the HIPS products. All information that is essential to the build process of version 6.1 of the HIPS product suite is provided. This includes how to build and verify the various products, library dependencies and tools required.

This profile does not describe any functional requirements, or the installation of the products into a test environment; these are covered in separate documentation.

1.3 Assumptions

The following assumptions have been made in the development of this profile:

- There is an understanding of the HIPS products and their use.
- The reader is familiar with Microsoft Visual Studio and its associated solution files.
- The term 'solution' refers to the Microsoft Visual Studio solution file.
- The developer/implementer has the correct software and versions to build the HIPS products.

1.4 Definitions and Acronyms

Item	Definition
UI	User Interface
SOAP	Simple Object Access Protocol, used to exchange structured information through web services.
WSDL	Web Service Description Language, used to describe the web services
JDK	Java Development Kit
WCF	Windows Communication Framework, used in the building of the web services.
MLLP	Minimal Lower Layer Protocol, transport protocol used in the Mirth Connect component to transport HL7 messages.
DMZ	Demilitarised zone, exposes external facing services.
CPU	Central Processing Unit of the computer.
XML	Extensible Markup Language.

Item	Definition
P2P	Provider to Provider, a HIPS component that allows documents to be sent from one Health Provider Organisation to another Health Provider Organisation.
SMD	Secure Message Delivery, a HIPS component that ensures documents are securely sent when using the P2P component.

2. Products Overview

The HIPS source code and related artefacts are contained in the HIPS Release version 6.1 zip folder. Under the HIPS Release version 6.1 zip folder the source code is contained in the following subfolder:

- SourceCodeSoftwarePackage_v6.1

2.1 Source Code Software Package

The SourceCodeSoftwarePackage_v6.1 folder contents are as follows:

- Documentation Folder: This folder contains all the documentation (including this document) associated with the HIPS Core Web Services and User interface web application.
- Core Folder:
 - HIPS Core Database zip: This zip folder contains the database scripts to create the database tables, stored procedures, reference data, and permissions. See the *HIPS 6.1 - Initial and Clean Installation Guide (Core)* document for the database setup details.
 - HIPS Core Source zip: This zip folder contains all the source code required to build the Core Web Services. The folder contents are as follows:
 - *trunk*: contains all source code required, this folder contents are as follows:
 - *extern-lib*: contains the external libraries that are referenced within the HIPS Core solution.
 - *HIPS*: contains the Visual Studio solution files and associated projects source code, including HIPS_AppServer and P2P_AppServer.
 - *Hipstest*: contains the Demo Harness solution source code, used for testing and evaluation of the HIPS Core functions.
 - *Mirth*: The source code and associated library files required to build the acknowledgement web service library for the Mirth Connect application.
 - *DMZ*: contains the Visual Studio solution files and associated projects source code, including SMD_AppServer and Decrypter_AppServer.
 - HIPS Core WSDL zip: The WSDL files for the web services in HIPS Core.

UI folder:

- HIPS Web UI Database zip: This zip folder contains the database scripts to create the database tables, stored procedures, reference data, and permissions. See the *HIPS 6.1 - Initial and Clean Installation Guide (UI)* document for the database setup details.
- HIPS Web UI Source zip: This zip folder contains all the source code required to build the UI Web Application. The folder contents are as follows:

- *trunk*: contains all source code required, this folder contents are as follows:
 - *extern-lib*: contains the external libraries that are referenced within the HIPS Core solution.
 - *HIPS_Web*: contains the application solution file and source code for the associated projects.

3. Build Process

The build process is designed to be self-contained. The build tasks can be run on any machine where Visual Studio and the Java Development Kit (JDK) are available. There should be no need to resolve library dependencies as all of the required libraries are included in the release package.

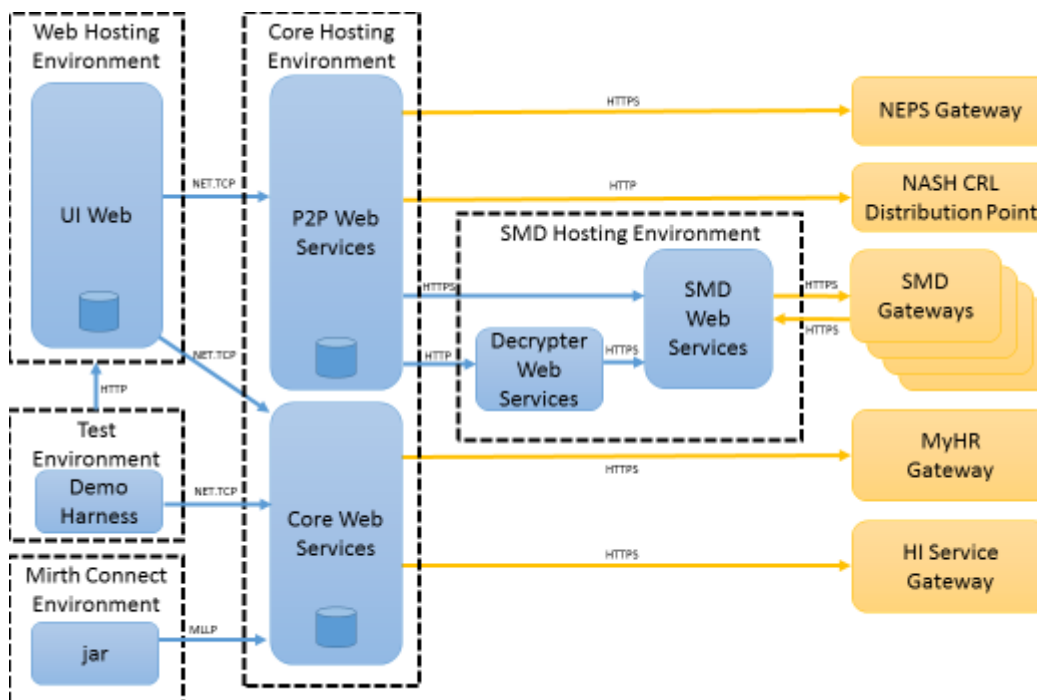
The remainder of the section will provide details on the tools, targets, dependencies and information related to the build process.

3.1 Build Target Products

The build process should result in the following products being built ready for release into a test environment:

- HIPS Core
 - Web Services
 - Mirth Connect jar file
 - Demo Harness
- HIPS P2P
 - P2P Web Services
 - Decrypter Web Services
 - SMD Web Services
- HIPS UI

The diagram below displays the above products and how they interact:



3.2 Build Tools and Environment

The following are the prerequisites that a developer will need to compile the source code into executable binaries and deployment packages for HIPS.

For HIPS the development environment will require the following software packages to be able to build the HIPS source code.

3.2.1 Core Web Services

Software	Version	Source	Component
Visual Studio Professional	2013 or 2015	https://www.visualstudio.com/en-us/products/visual-studio-professional-with-msdn-vs.aspx	HIPS_AppServer solution, DemoHarness solution
Visual Studio Update 2+	Only required if using Visual Studio 2013	Installed from Visual Studio notifications hub.	HIPS_AppServer solution, DemoHarness solution
NuGet Package Manager	2.0+	https://www.nuget.org/	HIPS_AppServer solution
Java SE Development Kit	8+	http://www.oracle.com/technetwork/java/javase/overview/index.html	Mirth Connect, a build.bat file is provided within the Mirth source package.

3.2.2 User Interface Web Application

Software	Version	Source	Component
Visual Studio Professional	2013 or 2015	https://www.visualstudio.com/en-us/products/visual-studio-professional-with-msdn-vs.aspx	HIPS_Web solution
Visual Studio Update 2+	Only required if using Visual Studio 2013	Installed from Visual Studio notifications hub.	HIPS_Web solution
NuGet Package Manager	2.0+	https://www.nuget.org/	HIPS_Web solution

3.3 Build Target

3.3.1 Visual Studio

Visual Studio includes a Configuration Manager tool that specifies how the projects within the solution are to be built and deployed. The following configurations have been set for each of the solutions.

Solution	Configuration	Description
HIPS_Core	Release	Builds a version of the application that can be deployed to a test or production environment. Each project within the solution has the <i>Target Platform</i> set to 'Any CPU'.
	Debug	Supports the debugging of an application. Used in a development environment.
	Puma_Debug	Not used for this HIPS release.
	Puma_Release	Not used for this HIPS release.
HIPS_P2P	Release	Builds a version of the application that can be deployed to a test or production environment. Each project within the solution has the <i>Target Platform</i> set to 'Any CPU'.
	Debug	Supports the debugging of an application. Used in a development environment.
HIPS_Web	Release	Builds a version of the application that can be deployed to a test or production environment. Each project within the solution has the <i>Target Platform</i> set to 'Any CPU'.
	Debug	Supports the debugging of an application. Used in a development environment.
DemoHarness	Release	Builds a version of the application that can be deployed to a test environment. <i>Target Platform</i> set to 'Any CPU'.
	Debug	Supports the debugging of an application. Used in a development environment.

3.3.2 Mirth Connect

The target for the Mirth Connect component is a java class (.jar file) required for the Mirth Connect component to accept messages from the HIPS Web Services. A build.bat file is included in *the HIPS Core Source* zip file under the Mirth folder to assist with the build of the java class.

3.4 Dependencies

All dependencies will be included as part of the HIPS Source Code package or aforementioned in this document.

3.5 Build Steps

The following section will outline the steps for the compilation and output of the HIPS Source Package components.

3.5.1 Debugging Uninstall Requirement

Visual Studio adds in an XML element <VsDebuggerCausalityData> to the SOAP messages it generates when debugging. This is used by Visual Studio to help it debug the WCF calls using system diagnostics.

This XML element must not be added to the SOAP requests sent to the My Health Record system. This function must be disabled by running the WCF Diagnostics Registration Tool - vsdiag_regwcf.exe to uninstall WCF diagnostics. This is located (on a 64 bit installation) in "C:\Program Files\Microsoft Visual Studio [13.0|14.0]\Common7\IDE" depending on the version you are using, thus it may not be included in the path of the VS Developer Command Prompt.

To uninstall the component, the following command needs to be run from a command prompt as an administrator:

```
vsdiag_regwcf.exe -u
```

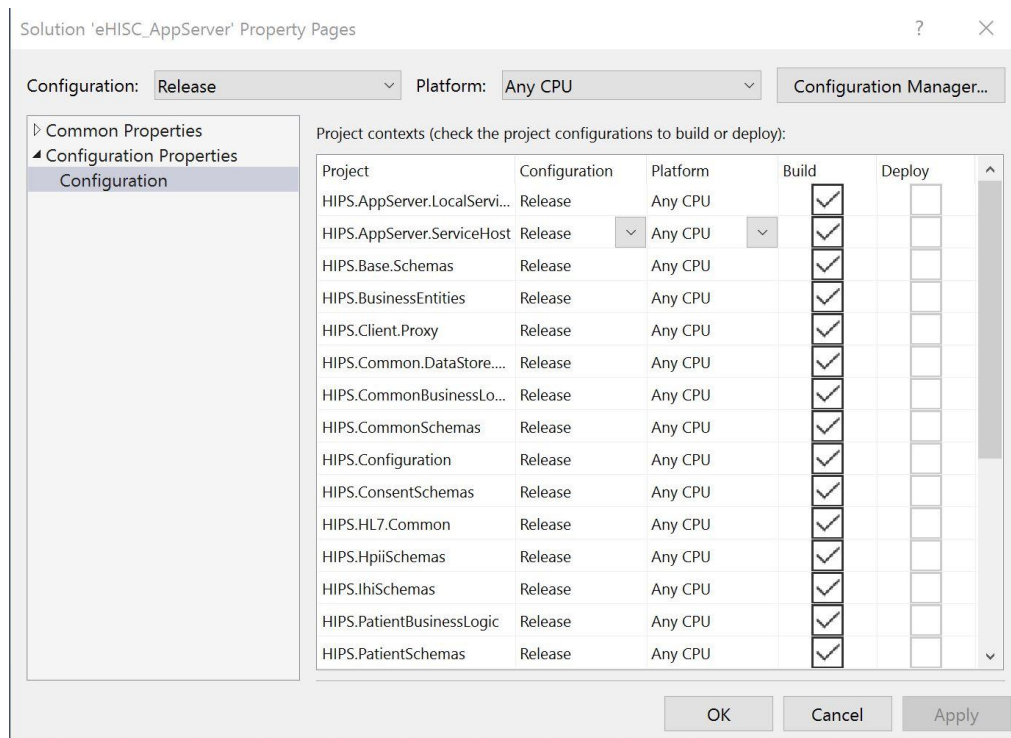
3.5.2 HIPS Core

The following steps relate to the source code found in the Core directory within the Source Code Software Package.

3.5.2.1 Web Services

Perform the following steps to build a release target of the *HIPS_AppServer* solution (.sln file) for the first time:

1. Open the solution (Double-click on the /trunk/HIPS/HIPS_AppServer/eHISC_AppServer.sln file) in Visual Studio. This will launch Visual Studio and load the HIPS_AppServer solution.
2. Right-click on the solution node in the Solution Explorer tab and select "Properties" from the context menu.
3. On the "Solution Property Pages" dialogue, select the "Configuration Properties" node.
 - a. Ensure the active solution configuration is set to the "Release" configuration.
 - b. Ensure the platform for each project is set to "Any CPU".
 - c. Click the "OK" button.



4. Build the solution. (Within Visual Studio either press the F6 key or click on the Build Solution menu item in the Build menu).
5. If any errors occur, right-click on the HIPS_AppServer solution in Solution Explorer and select 'Manage Nuget Packages'. The Nuget Package Manager window should be displayed, if any packages are missing a notification at the top will be displayed with a 'Restore' button. Click the 'Restore' button to download the packages required.

Once the build is complete, the produced web services and required libraries will be created in the 'HIPS_AppServer/HIPS.AppServer.LocalServiceHost/BuildPackage' folder.

To create a release folder with all of the required contents referenced in the HIPS installation guide:

6. Create a new 'HIPS Core Release v6.1' folder.
7. Create a 'HIPS' subfolder under the 'HIPS Core Release v6.1' folder.
8. Create a 'HIPS_AppServer' subfolder under the 'HIPS' folder.
9. Under the 'HIPS_AppServer' folder, create a 'binaries' subfolder.
10. Under the 'HIPS_AppServer' folder, create a 'ps_scripts' subfolder.
11. Copy the contents of the 'HIPS_AppServer/HIPS.AppServer.LocalServiceHost/BuildPackage' folder to the 'HIPS Core Release v6.1/HIPS/HIPS_AppServer/binaries' subfolder.
12. Copy the contents of the 'HIPS_AppServer/HIPS.AppServer.LocalServiceHost/Deploy/NEHTA Release' folder to the 'HIPS Core Release v6.1/appServer/binaries' folder. This is the required configuration files for a test or production environment.
13. Copy the contents of the 'HIPS_AppServer/buildscripts/appServer' folder to the 'HIPS Core Release v6.1/HIPS/HIPS_AppServer/ps_scripts' folder. This contains the PowerShell scripts used to create the web service site within IIS during the installation steps.

3.5.2.2 Mirth

To build the custom library (jar file) required for HIPS Web Services to connect to Mirth Connect for an MLLP Acknowledgement follow the steps below. If you are not using Mirth Connect to connect to the HIPS Web Services then the steps below are not required.

1. Under the "/trunk/Mirth" folder in the *HIPS Core Source* zip folder edit the *build.bat* file.
2. Check the JAVA_HOME directory points to the correct version (8+) of the Java SDK bin directory on your machine. (See Section 3.2.1 on where to download the Java SDK).
3. Check the MIRTH_HOME directory points to the Mirth Connect Home Directory.
4. Open a command prompt and run build.bat. This will create the jar file required.

Copy the new jar file to the release folder created below:

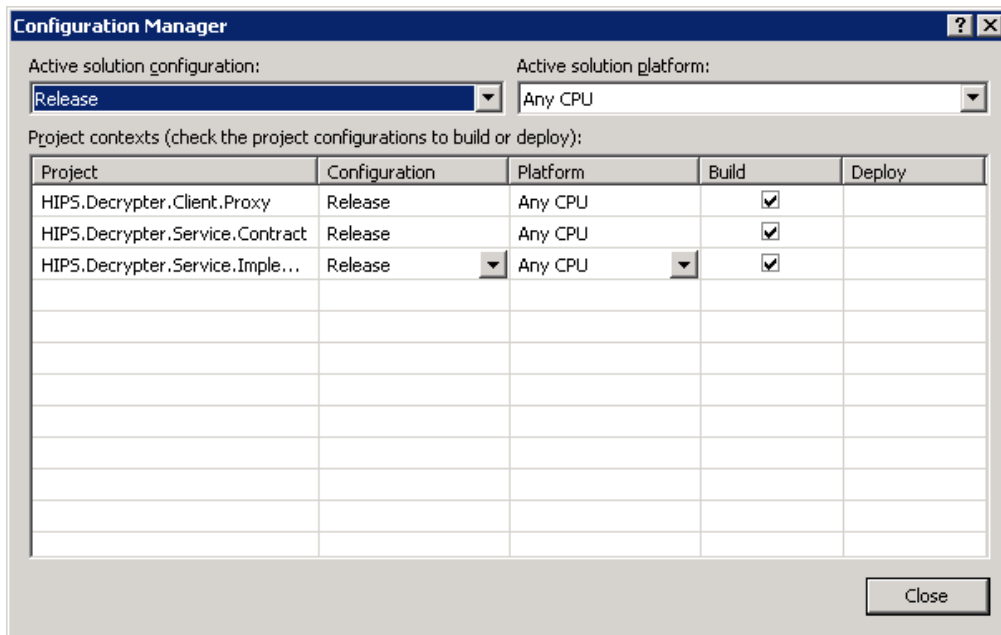
1. Under the 'HIPS Core Release v6.1' folder, create a 'Mirth' subfolder.
2. Under the 'Mirth' folder, create a 'custom-lib' folder.
3. Copy the newly created accept-ack-message.jar to the Mirth/custom-lib folder. This accept-ack-message.jar is required when installing the Mirth components.
4. Under the 'Mirth' folder, create a 'channels' subfolder
5. Copy the contents of the '/trunk/Mirth/channels' to the newly created channels folder.

3.5.2.3 P2P Decrypter

The Decrypter component is part of the P2P module of the HIPS product suite.

Perform the following steps to build a release target of the Decrypter_AppServer solution (.sln file) for the first time:

1. Open the solution (Double-click on the *Decrypter_AppServer.sln* file) in Visual Studio. The file is located under DMZ folder in trunk. This will launch Visual Studio and load the Decrypter_AppServer solution.
2. Right-click on the solution node in Solution Explorer and select "Properties" from the context menu.
3. On the "Solution Property Pages" dialogue, select the "Configuration Properties" node.
 - a. Ensure the active solution configuration is set to the "Release" configuration.
 - b. Ensure the platform for each project is set to "Any CPU".
 - c. Click the "OK" button.



4. Build the solution. (Within Visual Studio either press the F6 key or click on the Build->Build Solution menu item.)
5. Locate the HIPS.Decrypter.Service.Implementation project within the solution.
6. Right-click on the HIPS.Decrypter.Service.Implementation project and select 'Publish' from the menu.
7. Select Local Filesystem .\Publish folder.
8. Click 'Publish'.

Note: If an error is returned stating the path is too long then copy the solution to the root of the current drive (i.e. C:\) and attempt the build again.

Once the publish is complete, the produced web services and required libraries will be created in the 'Decrypter_AppServer/HIPS.Decrypter.Service.Implementation/Publish' folder.

To create a release folder with all of the required contents referenced in the HIPS installation guide:

1. Under the 'HIPS Core Release v6.1' folder, create a SMD_Decrypter subfolder.
2. Under the SMD_Decrypter folder, create a 'binaries' subfolder.
3. Copy the contents of the 'Decrypter_AppServer/HIPS.Decrypter.Service.Implementation/Publish' folder to the 'HIPS Core Release v6.1/SMD_Decrypter/binaries' subfolder.

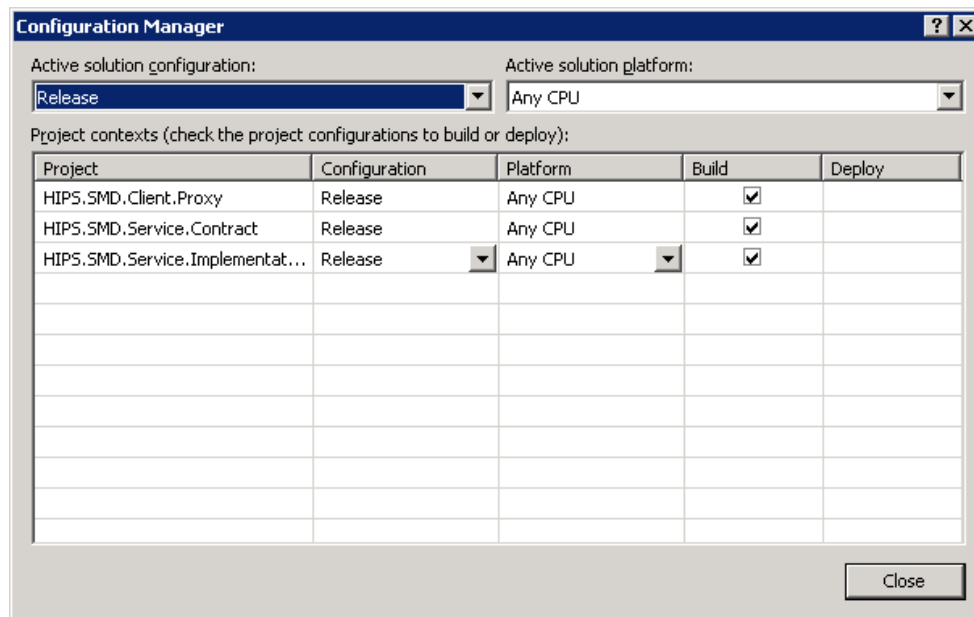
3.5.3 P2P SMD

The SMD component is part of the P2P module within the HIPS Core component.

Perform the following steps to build a release target of the *SMD_AppServer* solution (.sln file) for the first time:

1. Open the solution (Double-click on the *SMD_AppServer.sln* file) in Visual Studio. This will launch Visual Studio and load the *SMD_AppServer* solution.
2. Right-click on the solution node in Solution Explorer and select "Properties" from the context menu.
3. On the "Solution Property Pages" dialogue, select the "Configuration Properties" node.
 - a. Ensure the active solution configuration is set to the "Release" configuration.

- b. Ensure the platform for each project is set to "Any CPU".
- c. Click the "OK" button.



4. Build the solution. (Within Visual Studio either press the F6 key or click on the Build->Build Solution menu item.)
5. Locate the HIPS.SMD.Service.Implementation project within the solution.
6. Right-click on the HIPS.SMD.Service.Implementation project and select 'Publish' from the menu.
7. Select Local Filesystem .\Publish folder.
8. Click 'Publish'.

Once the publish is complete, the produced web services and required libraries will be created in the 'SMD_AppServer/HIPS.SMD.Service.Implementation/Publish' folder.

To create a release folder with all of the required contents referenced in the HIPS installation guide:

1. Under the 'HIPS Core Release v6.1/HIPS' folder, create an 'SMD_AppServer' subfolder.
2. Under the newly created 'SMD_AppServer' folder, create a 'binaries' subfolder.
3. Copy the contents of the 'SMD_AppServer/HIPS.SMD.Service.Implementation/Publish' folder to the 'HIPS Core Release v6.1/HIPS/SMD_AppServer/binaries' subfolder.

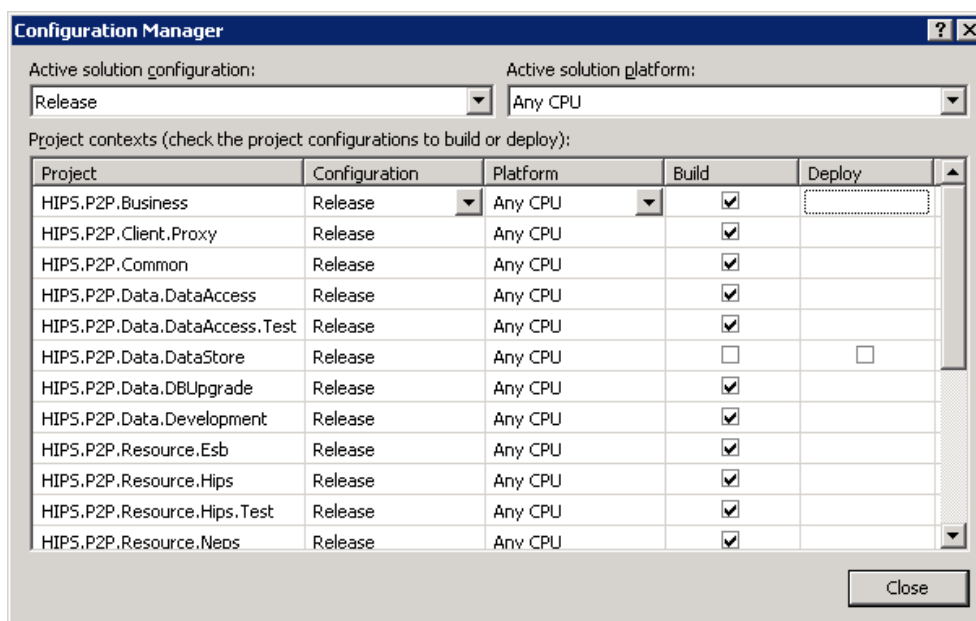
3.5.3.1 P2P Web Services

The P2P Web services is part of the P2P module of the HIPS product suite.

Perform the following steps to build a release target of the *HIPS_P2P_AppServer* solution (.sln file) for the first time:

1. Open the solution (Double-click on the *HIPS_P2P_AppServer.sln* file) in Visual Studio. This will launch Visual Studio and load the HIPS_P2P_AppServer solution.
2. Right-click on the solution node in Solution Explorer and select "Properties" from the context menu.
3. On the "Solution Property Pages" dialogue, select the "Configuration Properties" node.
 - a. Ensure the active solution configuration is set to the "Release" configuration.

- b. Ensure the platform for each project is set to "Any CPU".
- c. Click the "OK" button.



4. Build the solution. (Within Visual Studio either press the F6 key or click on the Build->Build Solution menu item.)
5. If any errors occur, right-click on the HIPS_P2P_AppServer solution in Solution Explorer and select 'Manage Nuget Packages'. The Nuget Package Manager window should be displayed, if any packages are missing a notification at the top will be displayed with a 'Restore' button. Click the 'Restore' button to download the packages required.
6. Locate the HIPS.P2P.Service.IIS project within the solution.
7. Right-click on the HIPS.P2P.Service.IIS project and select 'Publish' from the menu.
8. Select Local Filesystem .\Publish folder.
9. Click 'Publish'.

Once the publish is complete, the produced web services and required libraries will be created in the 'P2P_AppServer/ HIPS.P2P.Service.IIS /Publish' folder.

To create a release folder with all of the required contents referenced in the HIPS installation guide:

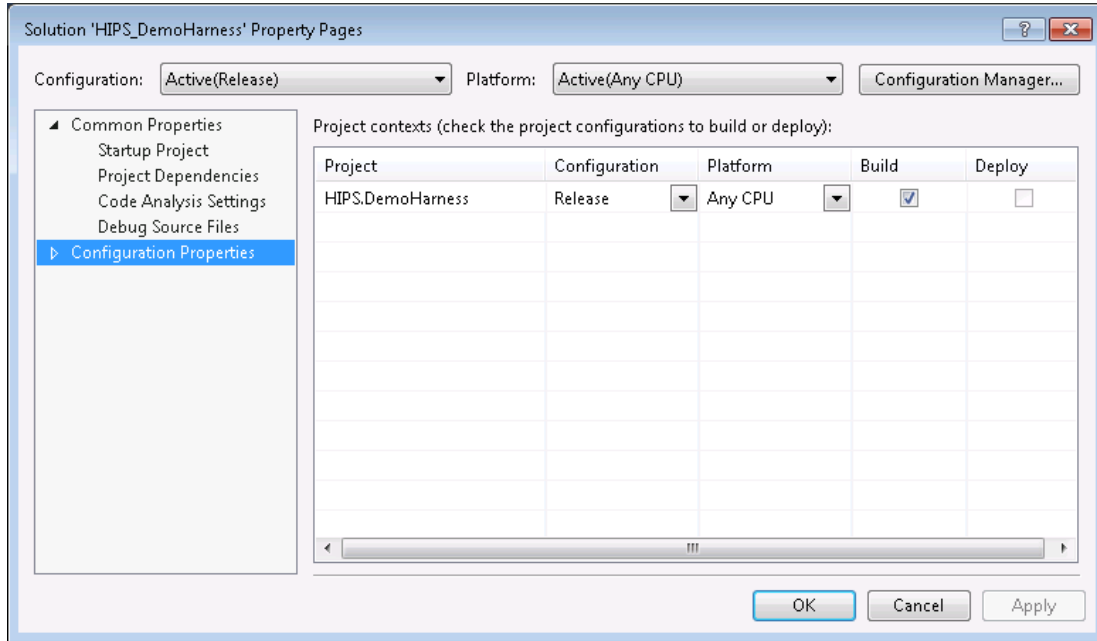
1. Under the 'HIPS Core Release v6.1/HIPS' folder, create a 'P2P_AppServer' subfolder.
2. Under the newly created 'HIPS Core Release v6.1/HIPS/P2P_AppServer' folder, create a 'binaries' subfolder.
3. Copy the contents of the 'P2P_AppServer/HIPS.P2P.Service.IIS/Publish' folder to the 'HIPS Core Release v6.1/HIPS/P2P_AppServer/binaries' subfolder.
4. Under the newly created 'HIPS Core Release v6.1/HIPS/P2P_AppServer' folder, create a 'ps_scripts' subfolder.
5. Copy the contents of the 'P2P_AppServer/buildscripts/appServer' folder to the 'HIPS Core Release v6.1/HIPS/P2P_AppServer/ps_scripts' subfolder.

3.5.3.2 Demo Harness

Based on dependencies, this part needs to be performed after HIPS P2P (see Section 3.5.3)

Perform the following steps to build a release target of the *HIPS_DemoHarness* solution (.sln file) for the first time:

1. Open the solution (Double-click on the /trunk/hipstest/HIPS_DemoHarness.sln file) in Visual Studio. This will launch Visual Studio and load the HIPS_DemoHarness solution.
2. Right-click on the solution node in the Solution Explorer tab and select "Properties" from the context menu.
3. On the "Solution Property Pages" dialogue, select the "Configuration Properties" node.
 - a. Ensure the active solution configuration is set to the "Release" configuration.
 - b. Ensure the HIPS.DemoHarness project is set to "Any CPU".
 - c. Click the "OK" button.



4. Build the solution. (Within Visual Studio either press the F6 key or click on the Build Solution menu item in the Build menu.)

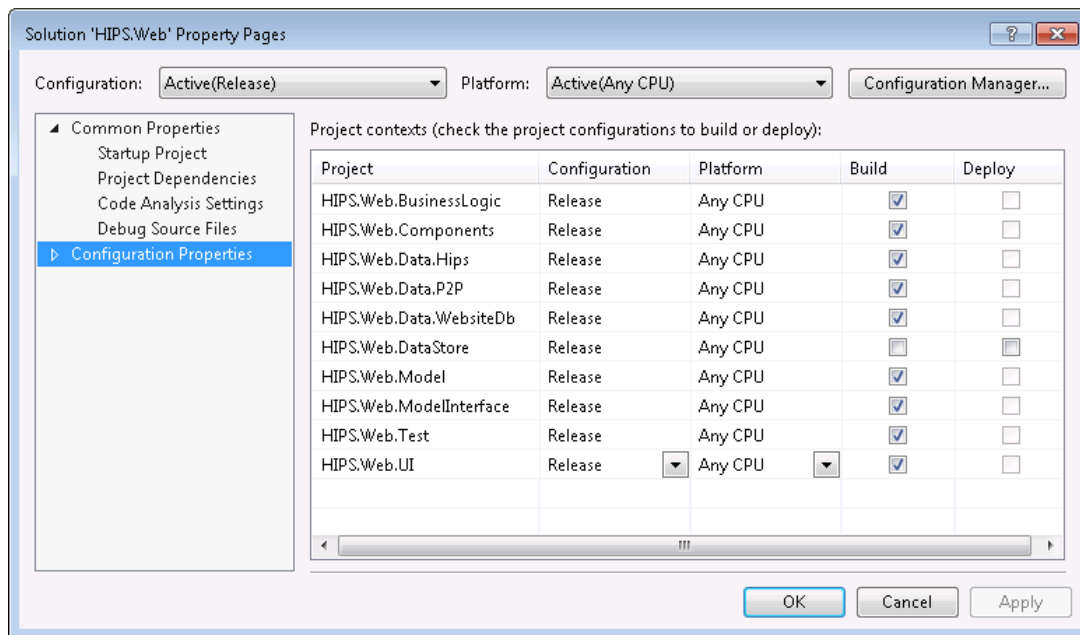
3.5.4 HIPS Web UI

The following steps relate to the source code found in the UI directory within the Source Code Software Package.

3.5.4.1 Web site

Perform the following steps to build a release target of the *HIPS UI* Web solution for the first time:

1. Open the HIPS.Web solution (Double-click on the /trunk/HIPS_Web/eHISC.Web.sln file).
2. Right-click on the solution node in the Solution Explorer tab and select "Properties" from the context menu.
3. On the "Solution Property Pages" dialogue, select the "Configuration Properties" node.
 - a. Ensure the active solution configuration is set to the "Release" configuration.
 - b. Ensure the platform for each project is set to "Any CPU".
 - c. Click the "OK" button.



4. Build the solution. (Within Visual Studio either press the F6 key or click on the Build->Build Solution menu item.)
5. If any error occurs, right-click on the HIPS.Web solution in Solution Explorer and select 'Manage Nuget Packages'. The Nuget Package Manager window should be displayed, if any packages are missing a notification at the top will be displayed with a 'Restore' button. Click the 'Restore' button to download the packages required.

Note: The HIPS.Web solution has a local reference to the compiled HIPS.AppServer dlls. If the HIPS.AppServer source code is modified and rebuilt then the referenced dlls in the HIPS \UI\HIPS Web UI Source\trunk\HIPS_Web\References folder must be updated. This can be either done manually or via a Post-Build Event from the HIPS.AppServer.LocalServiceHost project. There is an existing set of "IF EXIST" statements within this project properties Build Events that can be used to automate the copying process of the referenced dlls for the Web UI during a build cycle.

1. Locate the HIPS.Web.UI project within the solution.
2. Right-click on the HIPS.Web.UI project and select 'Publish' from the menu.
3. Select Local Filesystem .\Publish folder.
4. Click 'Publish'.

To create the web application for installation:

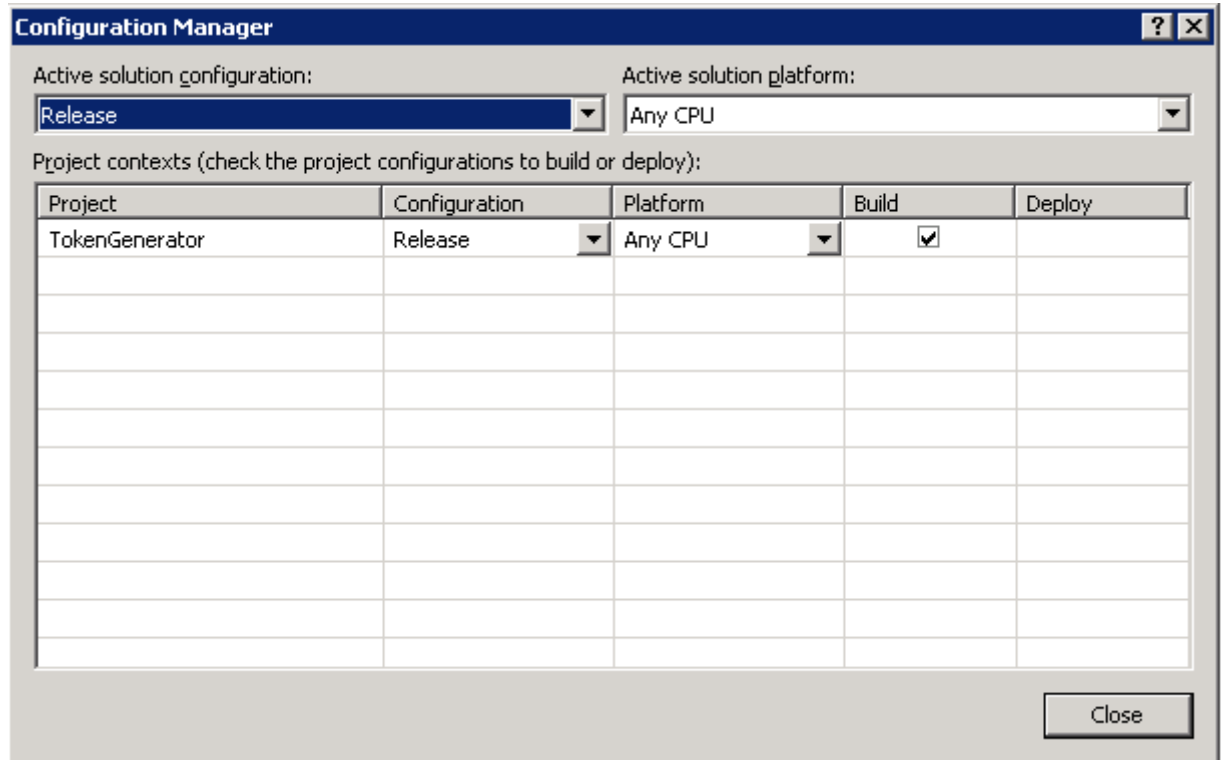
1. Create a new 'HIPS UI Release v6.1' folder.
2. Create a 'Web' subfolder.
3. Copy the contents of the 'HIPS_Web/HIPS.Web.UI/Publish' folder to the 'Web' subfolder.

3.5.4.2 Token Generator

Perform the following steps to build a release target of the *TokenGenerator* solution for the first time:

1. Open the TokenGenerator solution (Double-click on the /trunk/HIPS_Web/TokenGenerator.sln file).

2. Right-click on the solution node in the Solution Explorer tab and select "Properties" from the context menu.
3. On the "Solution Property Pages" dialogue, select the "Configuration Properties" node.
 - a. Ensure the active solution configuration is set to the "Release" configuration.
 - b. Ensure the platform for each project is set to "Any CPU".
 - c. Click the "OK" button.



4. Build the solution. (Within Visual Studio either press the F6 key or click on the Build->Build Solution menu item.)

4. Build Test

On successful completion of the build process you should have the following folder structure which will be referenced in the installation guides:

```
HIPS Core Release v6.1
  HIPS
    HIPS_AppServer
      binaries
      ps_scripts
    P2P_AppServer
      binaries
      ps_scripts
    SMD_AppServer
      binaries
    SMD_Decrypter
      binaries
  Mirth
    channels
    jar
  hipstest
HIPS UI Release v6.1
  Web
  TokenGenerator
```

To test the build process was successful, deploy the components and artefacts built in the preceding steps to a test environment by following the *HIPS 6.1 – Initial and Clean Installation Guide (Core)* document, *HIPS 6.1 – Initial and Clean Installation Guide (UI)* document and *HIPS 6.1 – Initial and Clean Installation Guide (P2P)*.

Further evaluation of the components can be done by following the *HIPS 6.1 – Evaluation Guide* document samples.