



eHISC v6.0.0

Build Guide

31 May 2016

Approved for external use

National E-Health Transition Authority Ltd

Level 25, 56 Pitt Street

Sydney, NSW 2000

Australia

www.nehta.gov.au

Acknowledgements**Council of Australian Governments**

The National E-Health Transition Authority 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.

Quality Systems, Inc

Mirth is a trademark of QSI Management, LLC, a subsidiary of Quality Systems, Inc. and is registered with the United States Patent and Trademark Office.

Disclaimer

The National E-Health Transition Authority Ltd (NEHTA) 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. NEHTA 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 National E-Health Transition Authority Ltd

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 NEHTA. All copies of this document must include the copyright and other information contained on this page.

Document information

Key information

Owner	Head of Delivery
Contact for enquiries	NEHTA Help Centre
t:	1300 901 001
e:	help@nehta.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.
6.0.0	May 2016	See release note (NEHTA-2263:2016) 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	6
2.1	Core Web Services.....	6
2.2	User Interface Web Application.....	6
3.	Build Process	8
3.1	Build Tools and Environment.....	8
3.1.1	Core Web Services	8
3.1.2	User Interface Web Application	8
3.2	Build Target	9
3.2.1	Visual Studio	9
3.2.2	Mirth Connect.....	9
3.3	Dependencies.....	9
3.4	Build Steps	9
3.4.1	Debugging Uninstall Requirement	9
3.4.2	eHISC Core	10
3.4.3	eHISC Web	12
4.	Build Test	14

1. Introduction

1.1 Purpose

The purpose of this document is to provide straightforward instructions for building of the eHISC suite of products from the provided source code solutions

The intended audience for this document is technical developers and/or implementers of the eHISC 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 eHISC products. All information that is essential to the build process of for version 6.0.0 of the eHISC 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 eHISC 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 eHISC 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
eHISC	eHealth Integration Source Code
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.

2. Products Overview

The eHISC source code and related artefacts are contained in the eHISC Release version 6.0.0 zip folder. Under the eHISC Release version 6.0.0 zip folder the source code is contained in the following sub folders:

- NEHTA-XXXX_2016_eHealthIntegrationSampleCode_CoreSoftwarePackage_v6.0.0

This folder contains the source code and related artefacts for the eHISC Core Web Services

- NEHTA-XXXX_2016_eHealthIntegrationSampleCode_UISoftwarePackage_v6.0.0

This folder contains the source code and related artefacts for the eHISC User Interface Web Application.

2.1 Core Web Services

The NEHTA-XXXX_2016_eHealthIntegrationSampleCode_CoreSoftwarePackage_v6.0.0 folder contents are as follows:

- *eHISC Core Database zip*: This zip folder contains the database scripts to create the database tables, stored procedures, reference data, permissions. See the *eHISC Release 6.0.0 - Initial and Clean Installation Guide (Core)* document for the database set up details.
- *eHISC Core Documentation zip*: This zip folder contains all the documentation (including this document) associated with the eHISC Core Web Services.
- *eHISC 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 eHISC Core solution.
 - *HIPS*: contains the application solution file and associated projects source code.
 - *Hipstest*: contains the Demo Harness solution source code, used for testing and evaluation of the eHISC Core functions.
 - *Mirth*: The source code and associated library files required to build the Mirth java application.
- *eHISC Core WSDL zip*: The WSDL files for the web services in eHISC Core.

2.2 User Interface Web Application

The NEHTA-XXXX_2016_eHealthIntegrationSampleCode_UISoftwarePackage_v6.0.0 folder contents are as follows:

- *eHISC Web UI Database zip*: This zip folder contains the database scripts to create the database tables, stored procedures, reference data, permissions. See the *eHISC Release 6.0.0 – Web Initial and Clean Installation Guide* document for the database set up details.
- *eHISC Web UI Documentation zip*: This zip folder contains all the documentation (including this document) associated with the eHISC UI Web Application.
- *eHISC 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 eHISC Core solution.
 - HIPS_Web: contains the application solution file and associated projects source code.

3. Build Process

The build process is designed to be self-contained. The build tasks can be run on any machine using Visual Studio, and for Mirth the Java Development Kit (JDK) is 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 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 eHISC.

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

3.1.1 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	eHISC_AppServer solution, DemoHarness solution
Visual Studio Update 2+	Only required if using Visual Studio 2013	Installed from Visual Studio notifications hub.	eHISC_AppServer solution, DemoHarness solution
NuGet Package Manager	2.0+	https://www.nuget.org/	eHISC_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.

3.1.2 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	eHISC_Web solution
Visual Studio Update 2+	Only required if using Visual Studio 2013	Installed from Visual Studio notifications hub.	eHISC_Web solution
NuGet Package Manager	2.0+	https://www.nuget.org/	eHISC_Web solution

3.2 Build Target

3.2.1 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
eHISC_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 eHISC
	Puma_Release	Not used for eHISC
eHISC_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.

3.2.2 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 eHISC web services. A build.bat file is included in the *eHISC Core Source* zip file under the Mirth folder to assist with the build of the java class.

3.3 Dependencies

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

3.4 Build Steps

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

3.4.1 Debugging Uninstall Requirement

Visual Studio adds in an XML node when debugging called "VsDebuggerCausalityData" to the SOAP messages. This is used by Visual Studio to help it debug the WCF calls using system diagnostics.

This XML node cannot be added to the SOAP requests when sent to the National Repository (My Health Record System), thus it must be removed. This is done by running the WCF Diagnostics Registration Tool - `vsdiag_regwcf.exe`. This is located (on a 64 bit installation) in "C:\Program Files\Microsoft Visual Studio [10.0|11.0|12.0]\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 following command needs to be run from a command prompt as an administrator (located as part of the standard Visual Studio 2013/2015 Tools):

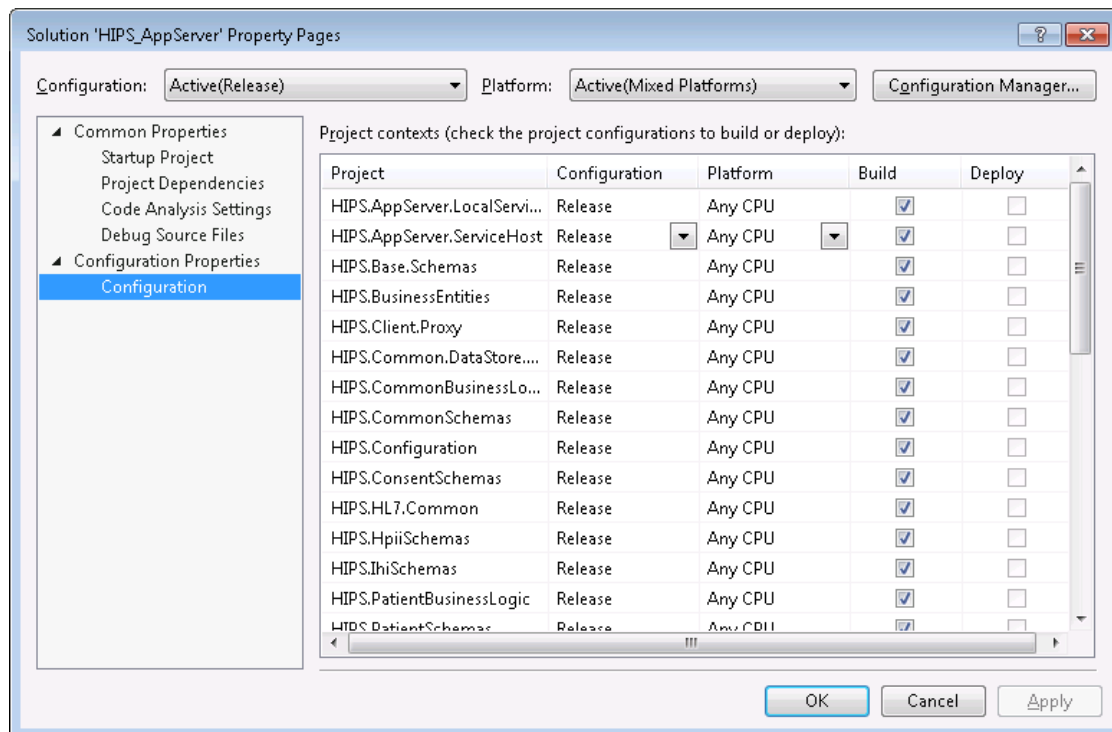
```
vsdiag_regwcf.exe -u
```

3.4.2 eHISC Core

3.4.2.1 Web Services

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

1. Open the solution (Double click on the *eHISC_AppServer.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" dialog, 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 eHISC_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 eHISC installation guide:

1. Create a new 'eHISC Core Release v6.0.0' folder.
2. Create an 'appServer' sub folder
3. Under the 'appServer' folder, create a 'binaries' subfolder.
4. Under the 'appServer' folder, create a 'ps scripts' subfolder.
5. Copy the contents of the 'HIPS_AppServer/HIPS.AppServer.LocalServiceHost/BuildPackage' folder to the 'binaries' subfolder.
6. Copy the contents of the 'HIPS_AppServer/HIPS.AppServer.LocalServiceHost/Deploy/NEHTA Release' folder to the 'binaries' folder. This is the required configuration files for a test or production environment.
7. Copy the contents of the 'HIPS_AppServer/buildscripts/appServer' folder the 'ps scripts' folder. This contains the powershell scripts used to create the web service site within IIS during the installation steps.

3.4.2.2 Mirth

To build the custom library (jar file) required for eHISC 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 eHISC Web Services then the steps below are not required.

1. Under the "/trunk/Mirth" folder in the *eHISC 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 3.1.1 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

Copy the new jar file to the release folder:

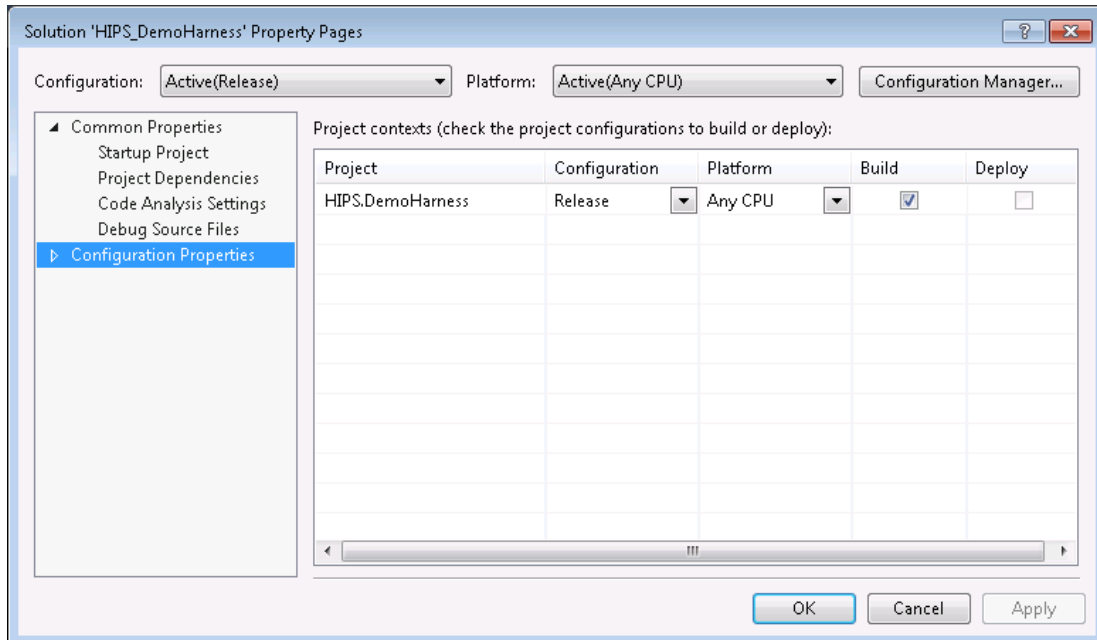
1. Under the 'eHISC Core Release v6.0.0' folder, create a 'Mirth' sub folder.
2. Under the 'Mirth' folder, create a 'custom-lib' folder.
3. Copy the newly created accept-ack-message.jar to the Mirth/custom-lib folder. This accept-ack-message.jar is required when installing the Mirth components.

3.4.2.3 Demo Harness

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

1. Open the solution (Double click on the eHISC_DemoHarness.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" dialog, 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.

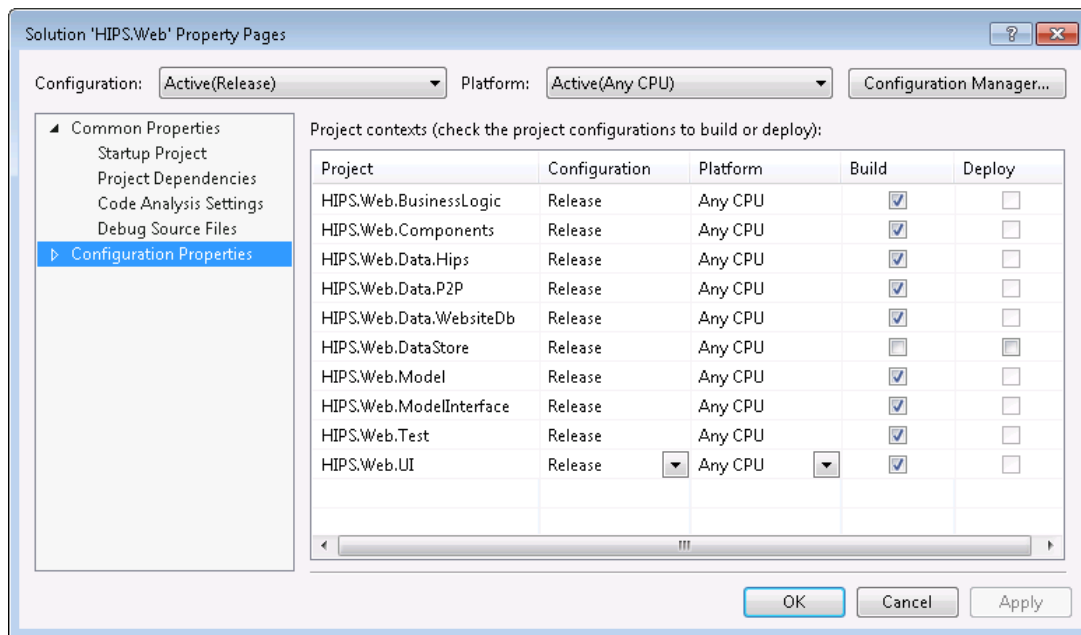


4. Build the solution. (Within Visual Studio either press the F6 key or Click on the Build->Build Solution menu item)

3.4.3 eHISC Web

Perform the following steps to build and run the *eHISC-UI* Web solution for the first time:

1. Open the *eHISC.Web* solution (Double click on the 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" dialog, 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 eHISC.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 eHISC.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 eHISC\UI\eHISC 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.

6. Locate the HIPS.Web.UI project within the solution.
7. Right click on the HIPS.Web.UI project and select 'Publish' from the menu.
8. Select Local Filesystem .\Publish folder.
9. Click 'Publish'.

To create the web application for installation:

1. Create a new 'eHISC UI Release v6.0.0' folder.
2. Create a 'Web' subfolder.
3. Copy the contents of the 'HIPS_Web/HIPS.Web.UI/Publish' folder to the 'Web' subfolder.

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:

eHISC Core Release v6.0.0

 appServer

 binaries

 ps scripts

 Mirth

 custom-lib

eHISC UI Release v6.0.0

 Web

To test the build process was successful, deploy the components and artefacts built in the preceding steps to a test environment by following the *eHISC Release 6.0.0 – Initial and Clean Installation Guide (Core)* document and *eHISC Release 6.0.0 – Initial and Clean Installation Guide (UI)* document.

Further evaluation of the components can be done by following the *eHISC Release 6.0.0 – Evaluation Guide* document samples.