NIH System to System (S2S) “Client” Installation Guide

Version 1.2.0.0

TABLE OF CONTENTS

INTRODUCTION

Objectives

The goal of this document is to provide a quick-start installation guide to enable technical staff to deploy then configure and verify the operation of the eRA Exchange System-to-System Client (S2Sclient) software package. The intent is to establish a basic working configuration that can be used to familiarize staff with the operation and characteristics of the S2Sclient and the exchanges it enables between their systems and the NIH eReceipts servers (eRA exchange).

The S2Sclient is designed to allow external parties (such as grantee service implementers or intra-government sites) to securely send and receive messages with the NIH eRA exchange servers as part of the grants application processing. The S2Sclient uses open standards-based message exchanges that are pre-configured to work with the NIH eRA exchange. The S2Sclient is named to differentiate its function from the main eRA NIH exchange server, but the S2Sclient is also able to function as a messaging server to receive NIH initiated messages.

For extended technical details and configuration options users are referred to the companion "Developers Guide" documentation. Also to change the configuration and to integrate the S2Sclient with existing production systems then the Developers Guide should be referenced.

The S2Sclient is provided and released "as is" by NIH. The components it contains are themselves maintained and supported by open public software projects. NIH maintains its own additions and extensions to those base components but not the components themselves. The NIH extensions are specifically associated with handling the NIH XML transactions and in their delivery, control and security.

The baseline functionality will continue to be adapted and extended by eRA NIH as required to support the grants application process. New S2Sclient installation packages will then be made available with those additional features and capabilities. However NIH expects that the core methods and API approach taken that conforms to the open public standards will remain stable and consistent.

Together there are two system-to-system interchange methods in the overall grants applications processing – the Grants.gov WSDL based submission process, and then the Hermes based S2Sclient facilities for all other interchanges relating to the subsequent grant application processing status and related organizational information. Existing and planned transactions include: Status request, Person information request, Person information update, Validations service, and Notice of Grant Award transaction. See the separate S2Sclient transactions document for complete details of current transactions and their layout structure schemas and content.

Operational Overview

This S2Sclient package is provided as a convenience to allow implementers to rapidly deploy a baseline system and test that it operates correctly. Implementers are responsible for configuring and adapting this baseline package to then operate in their own actual production environments (in-depth operational configuration details are provided in the companion "S2Sclient Developers Guide" document).

Basic installation and setup instructions are provided here for a typical envisioned Windows server system. However the components provided are open platform and are not just limited to Windows deployment; the software is known to run in typical UNIX environments that meet the deployment configuration pre-requisites.

The package includes sample test messages to establish that the S2Sclient is operating correctly using the default in/out folder managed interface with the eRA NIH server systems. These test messages can be used in both the eRA NIH pre-production external user acceptance testing area (Ext-UAT) and the production environment (see overview in figure 1 below).

There is a simple "ping" test that is designed to be a baseline test. Other tests will require availability of valid application test accounts for the NIH Commons system – see the main eRA NIH website for more details (http://era.nih.org) on sign-up and account creation.

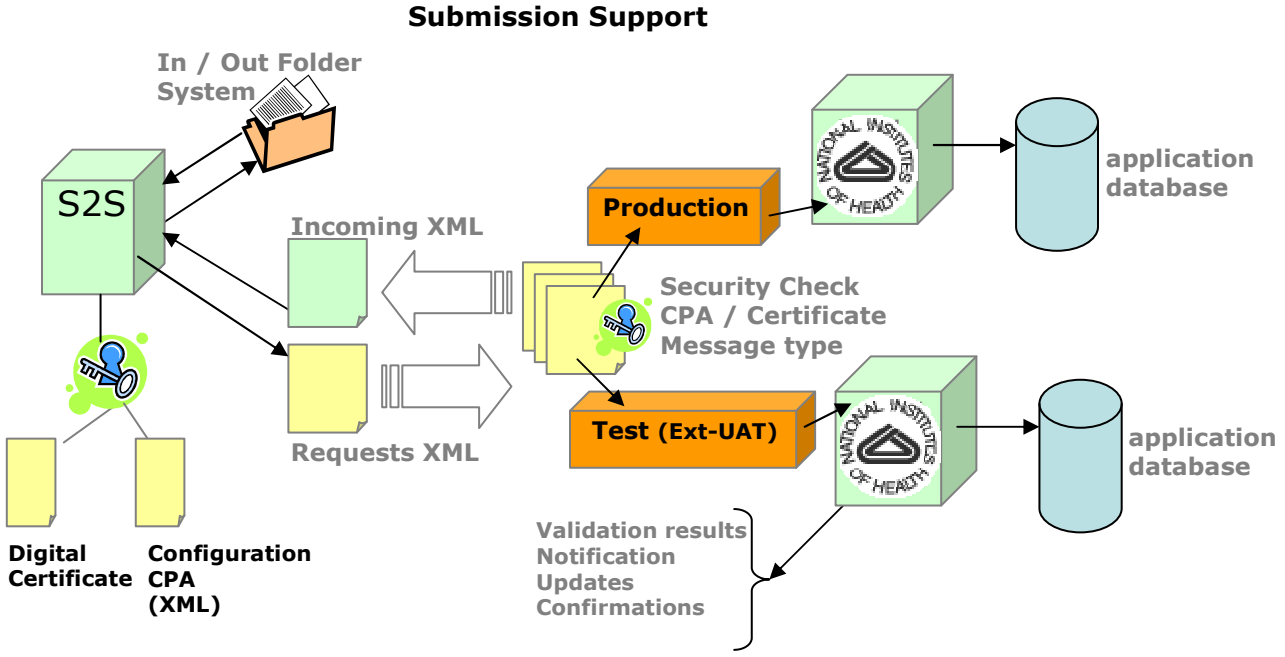


Figure 1 – Schematic overview of submissions support for application information exchanges.

Procedures and Overview

The installation and setup procedures for the S2Sclient software package are covered in this section. The first part here provides an overview of the steps, prerequisites and components. The second part then provides explicit instructions and details.

The installation and setup process consists of four steps:

1) Configuring the base system Java environment ready to install the S2Sclient (*Prerequisites*)
2) Downloading and unpacking the S2Sclient ZIP package and running the initial setup (*System*)
3) Configuring the security environment for SSL using your own digital certificate and CPA[1] (*Security*)
4) Starting the S2Sclient and testing the exchange of messages with NIH (*Testing*).

The software package includes the components that are required to be installed on a Windows 2000/XP platform in order to run the S2Sclient along with instructions on setting up the base environment that those components require (the specific Java Development Kit [JDK]). There are also software and hardware prerequisites that are discussed in the next section.

Prerequisites

The following is a summary of the intended deployment environment for the S2Sclient package. These components should be deployed prior to running the S2Sclient installation:

- Pre-installation environment:
    - Operating Systems: Windows 2000/XP
    - Java Versions: 1.5 - Development Kit (SDK) version only (Note: later versions are currently not supported and do not work with the package as configured; SDK v1.5+ support is anticipated but not yet available).
    - See detailed instructions for configuring the Java Environment setup below.

Installation Package

Once the pre-installation is complete then the S2Sclient install script can be executed. The following are then deployed from the S2Sclient package.

- Installation package components:
    - Application Servers: Tomcat 5.x is the default provided in the install package.
    - Database Engines: DerbyDB is the default embedded SQL database used.
    - Hermes V1.01 messaging server
    - NIH extensions

When that installation is complete – then the next step is configuration.

[1] CPA – ebXML Collaboration Protocol Agreement – this is an XML file provided by NIH that controls the business processes that are permitted between a S2Sclient system and NIH eRA eReceipts system.

- Executable standard test case components with S2Sclient installation:
    - Configuration and setup files to control the sending and receiving of transactions via outgoing and incoming file system structure.
    - Configuration files to control the profile for interfacing to the eRA NIH servers.
    - Sample java components for handling incoming and outgoing transactions.

Security

Transport layer security is provided by the S2Sclient using SSL (Secure Socket Layer) enabled communications with NIH. A default certificate configuration is provided to enable initial installation only of the S2Sclient. This default setup will not work and requires customization. Each installation is therefore expected to replace the default certificate setup with their particular private certificate, and to register that new certificate with NIH accordingly. More information on using and configuring SSL is provided in the security configuration section below.

Application level security is provided by using the ebXML CPPA (Collaboration Partner Protocol Agreement) also referred to as CPA (Collaboration Partner Agreement). This is a configuration file (XML) that controls the S2Sclient communications setup and the business level checking on the type of message exchanges permitted. NIH provides unique CPA.xml configuration files to registered partners. The CPA.xml is private and confidential between NIH and the partner and uniquely identifies the partner's message exchanges with NIH. Also the default certificate configuration is controlled from the CPA as well.

Customization

This completes the default environment that is installed as is. All the components such as Tomcat and Derby are pre-configured. However this packaging is not the sole deployment environment.

Equivalent components may be substituted by implementers, based on their own local system operational requirements (see Developers Guide for more information). Examples include using different servlet services instead of Tomcat, or different SQL compatible databases such as Oracle. Implementers are responsible for configuring these changes themselves, however a wide variety of commonly available COTS solutions are known to be compatible.

Starting the S2Sclient and Testing

The first test recommended is the "ping" test to ensure that communications are working correctly. See details of "Starting and Testing the S2Sclient" below.

Commence Installation

In the second part of these instructions the specific details of the installation procedures are now given.

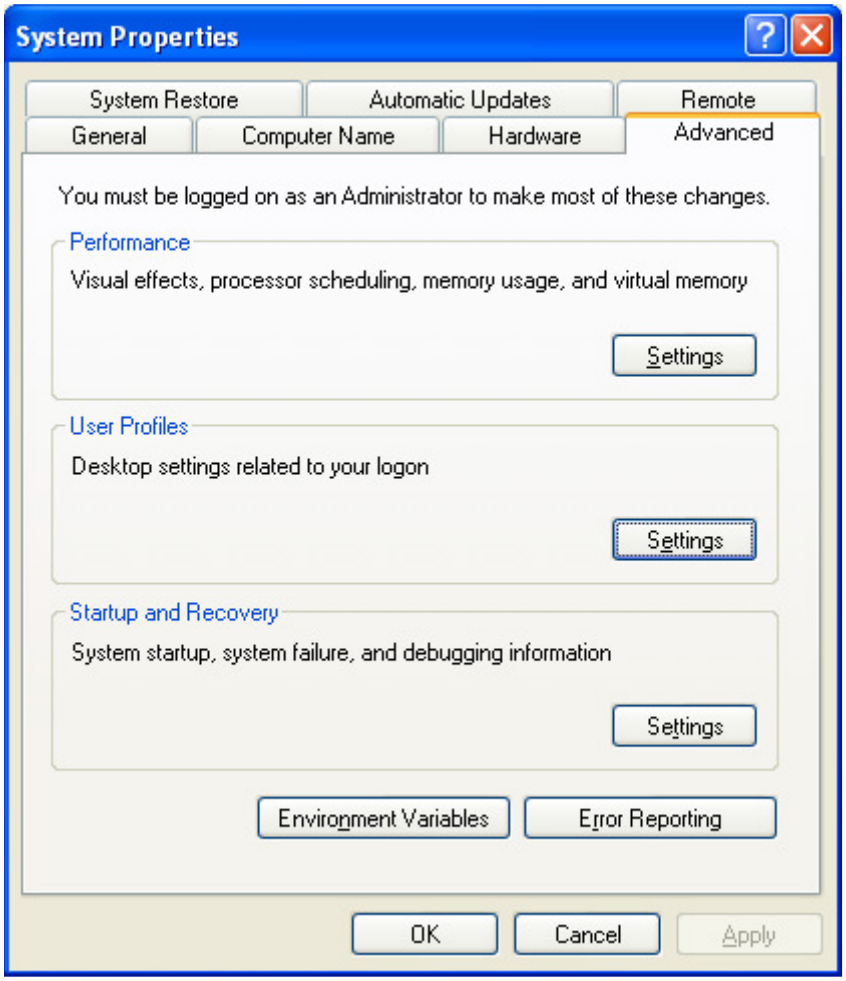
Java Environment

You will need to download and install the Java 1.5.x SDK system from Sun Microsystems (http://java.sun.com/javase/downloads/index_jdk5.jsp). The JDK development environment must be present to run the server environment (the JRE alone will ***not*** work), also older JDKs will not work because the newer version of TomCat is included in this NIH distribution. (Note: We expect to upgrade at some point once we have completed baseline system verification).

Then you must ensure that the JAVA_HOME environment variable is set for your Windows account to point to your installed JDK directory (i.e. c:\jdk1.5).

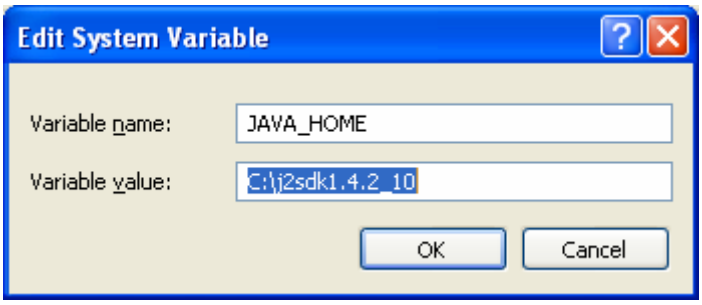
To verify this do the following:

First, from the "My Computer" icon – click on the "View System Information" link that is on the "System Tasks" panel, and this should open the system dialogue.



Next, click on the Environment Variables button at the bottom of the dialogue.

If the JAVA_HOME variable is already set in System variables – then no further actions is needed – otherwise click on the "Create New" system variable – and enter the name JAVA_HOME and the value for your java JDK directory also (by default this is off the root directory).



This is then saved and completes the configuration of the Java environment.

Downloading and Installing S2Sclient

The main S2Sclient can be downloaded from the eRA NIH provided URL link.

The package is provided as a ZIP file. The following steps will then unpack this and install it.

1) After downloading the ZIP to a local directory on the Windows system, then right click on the ZIP file and select "Extract All".
2) Once extraction is complete, open the directory that is created and locate the SETUP batch script file. By default the setup will place the S2Sclient in a directory > c:\s2sclient
3) If you need to change this default directory see editing instructions note below.
4) Next – to initiate the installation, click on the SETUP batch file and then press enter to continue.
5) The process will run automatically.
6) Once this completes, then you can locate the directory (default is c:\s2sclient – or whatever you specified as in instruction #3 above) – open that – and copy your CPA.xml file provided by NIH to this directory. Next you need to configure your SSL certificate setup before attempting to run the S2Sclient itself (startServer batch file).

Note: changing the default directory - if you wish to place the client in a different location, simply right click on the SETUP script file – select "Edit", and then change the location and save and exit the editor. This will change the directory where the Tomcat and Derby components are deployed. To change the location where the delivery folders sub-directory structure resides (the default is c:\S2Sclient\ ) – then the watch directory parameter in the exchangeclient.properties file should be modified as well. (Any text file editor such as NotePad can be used).

Configuring SSL security

This section provides instructions and guidelines for the setup and use of SSL. The S2Sclient is pre-configured to use SSL as installed however you will need to configure the base installation for your own machine environment. The SSL setup stage requires you to provide your own uniquely identifying certificate and configure the S2Sclient environment to use that. You must also send a copy of the certificate to be used to NIH via the HelpDesk support team. Instructions are now provided below.

You will need *administrator* privileges to complete this portion of the setup and configuration.

For the NIH eRA System itself the certificates used are issued by VeriSign. Normally these require no action on your part. Most application platforms come configured to trust VeriSign as a certificate authority and the default S2Sclient trust store setup contains a NIH certificate. If you still need a copy of the NIH eRA System certificates, please contact the NIH support HelpDesk.

There are three most common scenarios that we will consider here. The table gives a summary and the detailed instructions for each scenario then follow after the table.

#	Scenario	Comments	Actions
1	You will use the key store configuration shipped with S2Sclient "as is" and create your own certificate to use this.	You will need to setup the certificate information, and identify your host server by entering your host machine name	1. Create your own locally signed certificate and the associated key store 2. Export certificate from key store 3. Configure certificate alias name parameters 4. Send copy of certificate to NIH CGAP support 5. Edit host machine name parameters
2	You will create a key store - but then you will re-locate the S2Sclient and/or key store to another network location (other than the default installation)	You will need to edit some configuration files in addition to configuring the certificate details as in scenario #1	1. Create your own locally signed certificate and the associated key store 2. Export certificate from key store 3. Configure certificate alias name parameters 4. Send copy of certificate to NIH CGAP support 5. Edit host machine name parameters 6. Change configuration location parameters
3	You already have your own existing key store and certificate that is on your network servers and you will use that instead of the S2Sclient store	This requires 2 or 3 steps to complete setting up this configuration	1. Change configuration location parameters 2. Make sure NIH CGAP support has a copy of your certificate if needed (e.g. if it's not from a trusted certificate authority) 3. Optionally insert NIH certificate into your trust store 4. Edit host machine name parameters

If you do not already have a digital certificate, then you need to either purchase a digital certificate from a supplier company, or you can use the Java keytool utility program to create your own self-signed certificate.

If you use a self-signed certificate in your test environment, you will need to provide the eRA eXchange team with that certificate public key so it can be registered with our application server (contact the HelpDesk for more instructions on sending certificates to NIH).

Key store parameters reference

This table is provided as a reference for the options to the key store tool usage. Instructions for using that are provided below.

Table: Options for the `keytool` Command

Option	Sample Value	Meaning	Constraint
`-keyalg`	`RSA`	Format of the private key is RSA	Do not change
`-keysize`	`1024`	Key size is 1024 bits	Default setting – do not change
`-alias`	`Tomcat`	Key alias is Tomcat	Select value – default is "Tomcat"
`-dname`	`CN=Grants Usage`	Identification of the key is `CN=Grants Usage`	Optional – not required / used
`-keypass`	`apasswordvalue`	Password for the private key is apasswordvalue	Any value
`-storepass`	`apasswordvalue`	Password for the keystore file is apasswordvalue	Any value – can be different to keypass
`-keystore`	`filenameof.ks`	Keystore filename is `filenameof.ks`	Any physical file name
`-keystoretype`	`JKS`	Keystore filetype is Java Key Store (JKS)	Default setting - do not change

The scenarios for setting and using the key store and its values are now reviewed below.

Scenario #1 – Configuring the S2Sclient key store

The following is to setup and install your own SSL certificate into the S2Sclient environment.

If you already have your own certificate, then skip this and use the instructions in Scenario #3.

Creating the SSL certificate and associated key store details

1) From the Windows command prompt[2] – change directory to the default S2Sclient sub-directory: `(c:\S2Sclient)`

2) Then you need to establish the machine name of the server containing the S2Sclient. From the Windows command prompt – enter the command > hostname
and note the value that is displayed.

3) Next create a digital certificate using the keytool utility to create one for you and then export it to send to NIH. Use these two commands and change the items in bold to match your own preferences (note: when prompted for "First and Last name" provide the hostname value from 2) above).

```
keytool -genkey -alias tomcat -keyalg RSA -keystore s2sclient.ks -keypass changeit -storepass
changeit

keytool -export -alias tomcat -keystore s2sclient.ks -storepass changeit -file s2sclient.cer
```

the s2sclient.cer file generated needs to then be delivered to NIH CGAP support staff via email.

Items marked in **purple** may be changed to match the default Tomcat alias (if that was changed during installation) and then the `keypass` and `storepass` values in red may be set to your own preferred password values. The name of the **`s2sclient.cer`** file may also be changed too.

This then completes the SSL certificate management, next we look at setting up the machine level details for Scenario #1.

[2] To access the Windows command prompt – from the Windows Start menu choose the "Run" option and enter "cmd"

Configuring your machine name and key store in the message server

Next you need to change the machine name in the messaging server and configuration address URL. This setting is located in the msh.properties.xml file in the C:\S2SClient\tomcat5\bin sub-directory.

```
<Config>
      <!-- The URL of MSH (The URL of MSH to external systems) -->
      <!-- Keep trailing slash after context path to avoid unexpected problems
           with some application servers -->
      <URL>https://{hostname}:9443/msh/</URL>
```

(Note: to establish the machine host name of the server containing the S2Sclient. From the Windows command prompt – enter the command > hostname and note the value displayed).

Then this same file also contains configuration of the key store and alias settings. You will need to also enter those values here for the certificate you are using:

```
  <DigitalSignature>
      <TrustedAnchor>
        <KeyStore>
          <Path></Path><File></File><Password></Password>
        </KeyStore>
      </TrustedAnchor>
      <AckSign>
        <KeyStore>
          <Path>C:/S2SClient</Path>      <!-- this only changes for scenario #2 and #3 -->
          <File>s2sclient.ks</File>      <!-- this only changes for scenario #2 and #3 -->
          <Algorithm>RSA</Algorithm>
          <Alias>yourAliasName</Alias>   <!-- this is the alias to your certificate -->
          <Password>changeit</Password>  <!-- this is the password to the default key store -->
        </KeyStore>
      </AckSign>
    </DigitalSignature>
```

This completes the steps for Scenario #1; if you are configuring the default setup, then you can now skip to the following section that contains the instructions for starting and testing the S2Sclient. Otherwise refer to the next two sections for further configuring the S2Sclient.

Scenario #2 – Relocating the S2Sclient key store

In this scenario you perform all the steps from Scenario #1 – but then in addition you wish to change the location of the S2Sclient and / or the keystore locations.

The Java keytool utility is used to manage the key store for S2Sclient. The default key store for the S2Sclient is the s2sclient.ks file in the c:\s2sclient directory.

If you need to change this location path then edit the server.xml file located in C:\S2SClient\tomcat5\conf sub-directory and change the keystoreFile value as indicated below:

```
<Connector port="9443" maxHttpHeaderSize="8192"
               maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
               enableLookups="false" disableUploadTimeout="true"
               acceptCount="100" scheme="https" secure="true"
                              clientAuth="false" sslProtocol="TLS"
               keystoreFile="C:/S2SClient/s2sclient.ks" keystorePass="mykeypass"/>
```

Also the cacerts file located in the Java development sub-directory:

```
<java_home directory>\jre\lib\security
```

is the default controlling trust store for the certificates if you wish to make modifications to that (optional).

Scenario #3 – Using your own network key store

In this scenario you only need to perform the steps from the "*Configuring your machine name and key store in the message server*" section at the bottom of Scenario #1.

Optionally you may also need to perform one or more of these steps depending on your certificate and security policy details.

If you need to insert a digital certificate use this command to insert your digital certificate into the trust store:

1) From the DOS command prompt in Windows – change directory to the Java sub-directory: {i.e. cd c:\jdk1.5.0\jre\lib\security)
2) Then: `keytool -import -alias yourAlias -keystore ./cacerts -file c:\s2sclient\s2sclient.cer -keypass changeit -storepass changeit`

The following commands should be used to optionally import the NIH certificate digital certificate into your own keystore.

3) To insert the NIH provided digital certificate into your default key store: **(The items noted in bold should be changed to suit your deployment location and parameters).**

```
keytool -import -alias nihs2s -keystore ./cacerts -file c:\s2sclient\nihs2s.cer -keypass changeit -
storepass changeit
```

Table Summarizing the SSL Configuration directories and parameter files

Filename	Location	Description
CPA.xml	install root directory	Controls interface with NIH S2S server and identifiers the sender. Allows setting of tp:Endpoint tp:uri to machine hostname. Controls validation of transactions permitted to be used between NIH and S2Sclient.
server.xml	..\ tomcat5\conf	Controls location and name of the digital certificate key store.
msh.properties.xml	..\ tomcat5\bin	Machine hostname setting and key store name and alias for the messaging server use.
cacerts	<java_home directory>\jre\lib\security	Certificate management of trust store with the keytool utility

Starting and Testing the S2Sclient

Once the parameter changes for your particular scenario above have been completed for the certificate configuration then the startServer batch command can be used from the .c:\s2sclient sub-directory to begin actual testing of the message exchanging with NIH.

The installation package contains sample messages and a "ping test" directory that can be referenced – see the default folder structure– samples directory (S2Sclient/sampleMsg).

To send a test message copy and paste the folder (not just the message) from the sampleMsg sub-directory into the watchDirectory area for that type of message and inside the outgoing folder within that. We suggest the pingTest be used first to verify communications connection.

Trouble Shooting

Things to Check:

1) Issue - Startup gives an error message "Fatal Error – Unable to open and parse the CPA file in {directory location}": Do you have the CPA.xml located in your s2sclient directory? NIH should provide you with an XML file that contains your unique connection information as part of the sign-up process. There is a form to request this available from the NIH developer web site – this is called the CPA – Collaboration Profile Agreement.
2) Startup failure messages – did you correctly configure your SSL digital certificate?
3) Issue – Nothing sends: To run the ping test, did you copy the whole systemPing_msg directory – (not just the payload itself) – to the C:\s2sclient\watchdirectory\systemPingTest\outgoing directory?
4) Issue – No reply received back: Is your server information registered with the NIH server (Ext-UAT / Production?) – If not complete the registration form and send in details.
5) Startup batch process fails: Is port 8089 already in use? The default Tomcat port is set to be address 8089. If this is already being used by another process – edit the 3 Tomcat configuration files – server.xml, msh_client.properties and msh_properties . (See configuration tables locations details in table at end of this document)
6) Sending and Receiving messages - did you register your digital certificate with NIH?

Remediation:

1) Try stopping the TomCat server, removing the DerbyDB directory (just delete it) and the ebxmlms directory (delete too), and then restart the TomCat server. *Note: the msh.txt file contains the session log detail and c an be deleted or examined as needed.*
2) Message appears in console log "Error - cannot build message envelope": This can be due to one of two things – either the JDK you are using is not 1.4.x, or the ebxmlms tracking tables have become corrupted – perform remediation 1) above.
3) Re-check the SSL configuration setup instructions.
4) Make sure your digital certificate information is registered by NIH.

Tuning and Configuration

There are several things that can be configured fairly simply, while others require more detailed knowledge of the underlying software components.

1) Changing the delivery polling interval: This can be simply done by editing the exchangeclient.properties file and setting the parameter- polling_interval.
2) To change the port that TomCat is using (the default is port:8089): Please see the TomCat configuration documentation – item 5 in troubleshooting above.
3) Suppressing the verbose server status messages (running in quiet mode): We will be implementing support for this in a future release.

For other more extensive configuration options – please see the Developer Guide documentation.

Configuration Files.

The S2Sclient operations are controlled by various configuration files – these are itemized in the table here.

Advanced users may manually edit these tables as needed to change the setup and operations of the S2Sclient.

Filename	Location	Description
`CPA.xml`	install root directory	Controls interface with NIH S2S server and identifiers the sender.
`msh_passwd`	install root directory	Hermes login password
`tomcat-user.xml`	..\tomcat5\conf\.	Default user accounts
`Web.xml`	..\tomcat5\webapps\refimpl\WEB-INF\	CGAP startup servlet

Filename	Location	Description

exchangeclient.properties	..\tomcat5\webapps\refimpl\WEB-INF\classes\	#CPA File Location
		cpa.file.location=c:/S2SClient/cpa.xml
		
		# What directory are the files Picked up from
		watch.directory=c:/S2SClient/watchdirectory
		
		# What is the interval in which to check that directory (milliseconds)
		polling.interval=5000
		
		# Destination CPA - hard wired for LRP CPA
		destination.cpa=LRPCPA
		
		supported.message.types=
		systemPingTest, statusQuery, personQuery, personUpdate

gcrc.properties		Not used currently for S2S
messageHandlerFactory.properties		Associates the transaction type with the handler class
processoragents.properties		Class / processor mapping for message handlers
startup.properties		Invokes startup handler classes
stageddelivery.properties		Specifies the staged delivery message handler (pulled message handling)
log4j.properties		Controls generation of log files
server.xml	..\tomcat5\conf	Controls port address (default is 9443 for SSL)
msh_client.properties	..\tomcat5\bin	Controls port address (default is 8089)
msh.properties	..\tomcat5\bin	Controls port address (default is 9443 for SSL)