

Home Health Prospective Payment System (HH-PPS)

Java code Design

Author:

Tim Gallagher at 3M Health Information Systems

Date:

July 1, 2009

Table of Contents

Introduction 1

Required Components 1

Connecting to Java 2

Connecting to the HH-PPS Java Bridge 4

Visual Basic example 4

C/C++ example 7

Introduction

In 2008, 3M was tasked with providing a new version of the HH-PPS system while maintaining the same logic as version 2.03. While the previous versions were written in the C-language with an accompanying pseudo code document, the new version is written using the Java language. This changes the way the 3rd party vendors who relied on the DLL in the past will call the new version. This document explains the requirements for connecting to the new Java version of the HH-PPS.

Since DLL is a Windows operating specific module, this document will not reference configuration of the DLL on any other operating system.

Required Components

There are several components required in order to use the Java version. All of these components must be installed in a set of pre-defined folders in order to properly connect to the HH-PPS.

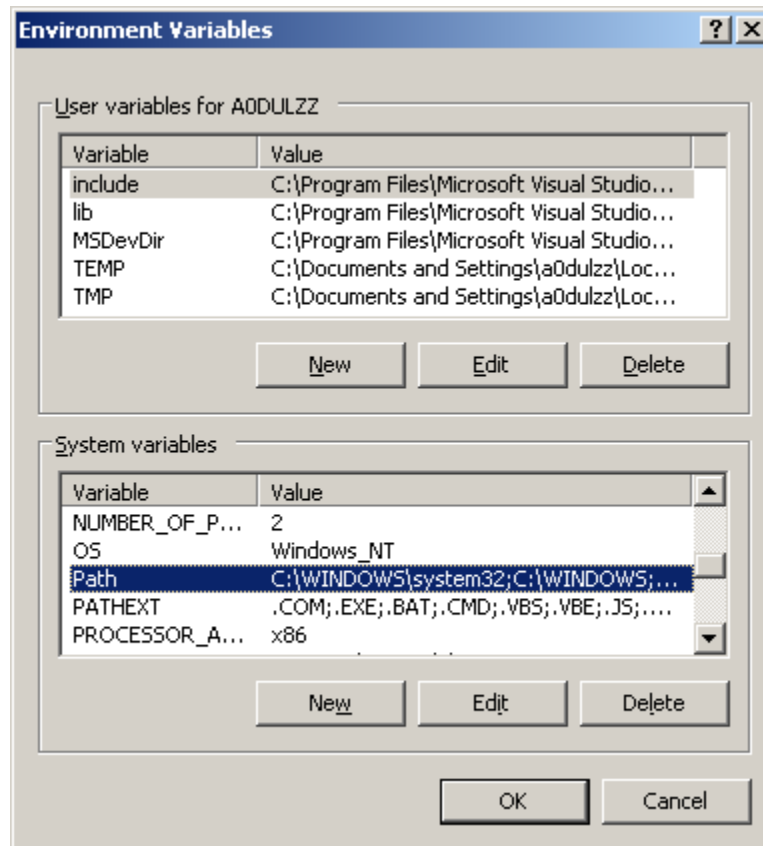
Component or File	What is it?	Location
Java 1.5	This is the Java run time environment which can be downloaded for free at http://java.sun.com/javase/downloads (The Bridge DLL also worked with Java 1.6) This component has its own installation application.	The installation program will decide where to place this. Use the defaults.
HomeHealthJavaBridge.dll	This file is the DLL which a Visual Basic, C/C++ or other non-Java application will use to connect to the HH-PPS Java version.	This should be placed in the C:\PROGRAM FILES\HOMEHEALTHGROUPER folder
HomeHealthJava.jar	This is a Java library file which contains the HH-PPS program	This should be placed in the C:\PROGRAM FILES\HOMEHEALTHGROUPER folder
HomeHealthGrouper.properties	This is the configuration file for the HH-PPS program	This should be placed in the C:\PROGRAM FILES\HOMEHEALTHGROUPER\config folder
Logging.properties	This is a standard Java Logging configuration file which has a default logging file defined as C:\PROGRAM FILES\HOMEHEALTHGROUPER\logs\HomeHealthJava0.log. Note that the C:\PROGRAM FILES\HOMEHEALTHGROUPER\logs must exist in order to record the log files.	This should be placed in the C:\PROGRAM FILES\HOMEHEALTHGROUPER\config folder.
Grouper data components	This is a folder containing a set of file with data related to the current set of HH-PPS Groupers installed.	These folders will be sub folders of C:\PROGRAM FILES\HOMEHEALTHGROUPER

Connecting to Java

After the Java installation, the Java relate DLLs may not be accessible to the Bridge DLL. You may have to put some extra folders onto the system PATH. This can be done from:

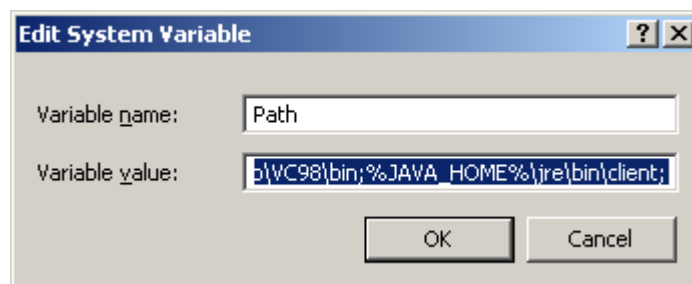
Start → Settings → Control Panel → System → the “Advanced Tab” → “Environment Variables”

This should bring up the following dialog box:



Graphic: Environment Variables

Select the PATH in the “System variables” and click Edit. In the following dialog,



Graphic: Editing Path variable

make sure the following types of paths are listed, and if not then add them:

```
C:\Program Files\Java\jre1.6.0_03\bin;  
C:\Program Files\Java\jre1.6.0_03\bin\client;
```

The item in blue above may be different for your installation, depending on the Java version you are using. If your installation creates an environment variable JAVA_HOME then you can add the paths as:

```
%JAVA_HOME%\bin;  
%JAVA_HOME%\bin\client;
```

If you installed a Java Development Kit (JDK) you may have to put the paths in like this:

```
C:\Program Files\Java\jre1.6.0_03\bin;  
C:\Program Files\Java\jre1.6.0_03\jre\bin\client;
```

These paths help Windows find the “jvm.dll” and supporting files.

Connecting to the HH-PPS Java Bridge

The HH-PPS Java Bridge is a DLL that allows non-Java applications to invoke the HH-PPS Java version. It requires the above components to be installed prior to its use. Its purpose is to:

- 1) accept data from the non-Java application
- 2) start the Java virtual machine (JVM) if needed
- 3) pass the data (untouched) to HH-PPS Java Grouper
- 4) collect the returned scoring information
- 5) place the scoring information into the supplied data elements provided during from the non-Java application.

Important: it is the responsibility of the calling application to allocate the space for the returned scoring data. This bridge will not provide such allocation, and, although it may score a record properly, it may not provide the scoring results properly if there is not enough memory allocated to store the results in.

For debugging purposes, setting the HIPPS code to “DEBUG” prior to calling the Home Health Grouper, will cause many pop-up message to be displayed by the Grouper. These pop ups will require user interaction, but are helpful in determining why the Grouper may not be functioning.

Visual Basic example

This example of connecting to the bridge uses Visual Basic (VB) as the programming language (knowledge of is assumed), and uses the OASIS B record format for the episode data. In order to connect, the VB application must identify the DLL and function with it that will

process the episode record. This is done with the following (note that multi-line markers are removed for clarity):

```
Private Declare Sub ProcessRecord_v203 Lib "C:\Program  
Files\HomeHealthGrouper\HomeHealthJavaBridge.DLL" Alias  
"?process_record_v203@@YGXPAD0000@Z"  
    (ByVal outData As String,  
    ByVal oasisRecord As String,  
    ByVal oasisTreatment As String,  
    ByVal version As String,  
    ByVal invalidFlag As String)
```

If you have used the previous version of the DLL, which was written in C, you will notice that the items in blue are the only things different. The first item, **ProcessRecord_v203**, is an arbitrary name, and is left to programming preference in order to distinguish it from the original DLL connection. The second item, **C:\Program Files\HomeHealthGrouper\HomeHealthJavaBridge.DLL**, is the location of the DLL bridge which should be similar to identifying the previous version which was named "Grouper.dll". The last item, **?process_record_v203@@YGXPAD0000@Z**, is the identifier internal to the DLL. This name can not be changed.

Next allocate some return data variables and space to hold the scoring information. This is similar to the previous version of the DLL. This data is considered in-out variables which are sent to the Grouper and populated with the scoring information. If there is not enough space to properly store the information, the scoring results may not be complete.

```
Dim hippsCode As String * 5  
Dim oasis As String * 18  
Dim version As String * 5  
Dim invflag As String * 1
```

Next the application must provide the OASIS B record as a String of 1448 characters. This should be allocated and filled by the VB application as in the following example that reads the data from an external file:

```
Dim record as String  
...  
Line Input #1, record
```

Next the application makes a call to the HH-PPS Java Bridge in order to score the record.

```
ProcessRecord_v203 hippsCode, record, oasis, version, invflag
```

The first time this function is called a progress pop-up will display that the Home Health Java Grouper is initializing. This helps to ensure that the bridge is working properly. After the first call, the initializing pop-up will not display until you restart the VB application.



Graphic: Home Health Grouper initialization pop-up

C/C++ example

This example uses C/C++ to connect to the HH-PPS Java Bridge and score a record. Note that all "output variables" require that the method caller allocate the appropriate amount of space in order to have this method populate them with the information.

parameters:

hippsCode - 5 character, plus null terminated string of the HIPPS code.
An output variable, total size 6 chars

record - 1448 character null terminated string of the OASIS B record used to score.

oasisTreatment - 18 character, plus null terminated string to hold the OASIS Treatment/Assessment value. An output variable, total size 19 chars.

version - 6 character, plus null terminated string of the Grouper's logic version identifier. An output variable, total size 7 chars

invflag - 1 character, plus null terminated string of the invalidity flag. An output variable, total size 2 chars

For debugging purposes, if the initial value of hippsCode = "DEBUG" then the debug messages will pop up and must be responded to by a user.

Environment variables:

home_health_lib = the Java jar file. Default is: "C:\PROGRAM FILES\HOMEHEALTHGROUPE\HomeHealthJava.jar"

home_health_logging = the Java system logging config file.
Default is: "C:\PROGRAM FILES\HOMEHEALTHGROUPE\config\logging.properties"

home_health_jni_error = the default Java JNI (connection between this DLL and JVM) error file. Default is: "C:\PROGRAM FILES\HOMEHEALTHGROUPE\HomeHealthJavaErrors.log"

home_health_quiet = indicates whether or not to have pop-up dialogs during the processing for critical events such as initialization, grouping errors, etc. values are true or false - the default is false,

home_health_debug = indicates whether to display debug pop-ups. This overrides the DEBUG flag used in the hippsCode but only if the value is present.

More detail is explained in the source code.