

## MyDelivery Source Code – Version 0.9.31 June 2010

The MyDelivery source code was created by the United States National Library of Medicine except for portions that were privately created and permission granted for distribution by the respective authors.

### DISCLAIMER

THIS SOFTWARE, CALLED MYDELIVERY, IS RELEASED TO THE PUBLIC DOMAIN BY THE NATIONAL LIBRARY OF MEDICINE, DEPARTMENT OF HEALTH AND HUMAN SERVICES, HEREBY REFERRED TO AS “NLM”. ANY PERSON OR ORGANIZATION RECEIVING MYDELIVERY SHALL BE HEREBY REFERRED TO AS “RECIPIENT”. MYDELIVERY IS PROVIDED “AS IS” AND NLM MAKES NO WARRANTIES EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. NLM MAKES NO REPRESENTATION THAT USE OF MYDELIVERY WILL NOT INFRINGE ON ANY PATENT OR PROPRIETARY RIGHTS OF THIRD PARTIES. IN NO EVENT SHALL NLM, THE UNITED STATES GOVERNMENT, OR ITS EMPLOYEES BE HELD LIABLE TO RECIPIENT IN ANY MANNER WHATSOEVER FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

### **Instructions for compiling and installing the client**

1. Install the SOAP Toolkit 3.0 on your computer. This is freely available from [msdn.microsoft.com](http://msdn.microsoft.com).
2. Unzip the mydelivery.zip file and extract the files to a suitable location on your hard disk. The MyDelivery source code is designed to be compiled with Visual Studio 2010. The distribution files are located in three folders:

CLIENT - Contains all source code to compile the MyDelivery client.

DATABASE - Contains a sample database named 'mydelivery1' to be installed on SQL Server.

SERVER - Contains all source code for the MyDelivery server.

3. The CLIENT source code is located in six folders:

ApiTest, ComMgr, ComMgr+, MDServer, MyDelivery, and Usage.

To compile the client using Visual Studio 2010, open the ComMgr+.sln file in the ComMgr+ folder.

There will be six projects with the same names as the six folders.

4. It may be necessary to modify two source files prior to compilation:
  - a. The common.h file in the commgr project. Modify the path for the import soap3.dll if necessary. This depends on where the SOAP 3 toolkit was installed on your computer.
  - b. The wsdl.cpp file in the commgr project. Modify #WEBSERVER to point to your MyDelivery website URL.

5. Select Rebuild Solution.

After compilation, there will be six executables located in the bin folder:

ApiTest.exe, ComMgr.dll, ComMgr+.exe, MDServer.exe, MyDelivery.exe, and usage.dll.

6. Copy these six files to a suitable folder, such as c:\program files\mydelivery.
7. At the command prompt in a command window, register the MyDelivery executable:  
`mydelivery.exe /regserver`
8. At the command prompt in a command window, register the commgr+ executable:  
`commgr+.exe /regserver`
9. Run the MyDelivery client: mydelivery.exe. Refer to the MyDelivery client documentation for using this.
10. The Apitest.exe is a sample application for using the MyDelivery API. Refer to the MyDelivery client documentation for using this.

## **Instructions for installing the database, 'mydelivery1'**

After unzipping mydelivery.zip, a sample database is located in the DATABASE folder. Two files are required for the mydelivery1 database:

MyDelivery1\_Data.mdf (the database)

MyDelivery1\_Log.ldf (the log file)

Refer to the MyDelivery Server documentation for installing this on SQLServer Express, SQL Server 2005, or SQL Server 2008. Instructions are included for using database mirroring on SQL Server 2005.

## **Instructions for installing the server software**

After unzipping mydelivery.zip, the MyDelivery server source code is located in the SERVER folder.

Please refer to the MyDelivery Server documentation for complete instructions for configuring and setting up a MyDelivery server. There are various types of servers (Front-End Processor, Client Manager,

and Database Server). It is possible to have all these inside one machine, or they can be configured as separate machines. The instructions here only explain how to compile the server software. Five folders of files are required for each Front-End processor:

1. **bin** - contains the initialization file for MyDelivery services: mydelivery.ini. These services usually hard-code the path to the bin file as c:\bin, so if you want to put this file on a different disk drive, you should change the code in the services appropriately.
2. **MDServices** – contains services used by MyDelivery servers. The following projects should be compiled in the order listed. Each has its own separate solution file.
  - **CacheData** – Open the solution file, CacheData.sln with Visual Studio 2010 and compile.
  - **mdFlusher** – Open the solution file, mdFlusher.sln with Visual Studio 2010. Open Properties for mdFlusher, and under References set up the correct path to the CacheData DLL. Compile.
  - **ClientList** – Open the solution file, ClientList.sln with Visual Studio 2010 and compile.
  - **mdClientListService** – Open the solution file, mdClientListService.sln with Visual Studio 2010. Open Properties for mdClientListService, and under References set up the correct path to the ClientList DLL. Compile.
  - **FileXFER** – Open the solution file, mdFileXFER.sln with Visual Studio 2010 and compile.
  - **mdFileXFERService** – Open the solution file, mdFileXFERService.sln with Visual Studio 2010. Open Properties for mdFileXFERService, and under References set up the correct path to the FileXFER DLL. Compile.
  - **mdMonitor** – Open the solution file, mdMonitor.sln with Visual Studio 2010 and compile.
  - **Pinger** – Open the solution file, Pinger.sln with Visual Studio 2010 and compile.
  - **mdPinger** – Open the solution file, mdPinger.sln with Visual Studio 2010. Open Properties for mdPinger, and under References set up the correct path to the Pinger DLL. Compile.
  - **mdClientManager** – If you are configuring the MyDelivery server to run as a single machine, then mdClientManager shall be installed and running on that machine. It is possible to put mdClientManager on its own machine in a multiprocessor server configuration. Whatever the case, open the solution file, mdClientManager.sln with Visual Studio 2010. Open Properties for mdClientManager, and under References set up the correct path to the Pinger, CacheData, ClientList, and FileXFER DLLs. Compile.

To properly install the services, copy the Microsoft installutil.exe utility to the c:\bin folder. Edit the mdsetup.bat file to point to the correct paths for the services. Then execute the mdsetup.bat file to install the services. After this, use the Windows Services utility to set up the proper login ID credentials for each service. Each service should be configured to run automatically upon bootup, and should always be running.

3. **NonSSL** – This is a folder containing a small website, used only for MyDelivery Front-End Processors that use SSL or TLS (HTTPS communications). This folder is not needed for MyDelivery Front-End Processors that use only HTTP communications. The purpose of NonSSL is to force any attempt to use HTTP communications to HTTPS. Copy this folder to an appropriate location on hard disk and share it as a website named NonSSL. Using IIS Manager, allow anonymous access to this website. Edit the SSLRedirect.aspx.vb file to point to the proper location of the secure mydeliveryweb folder on your hard disk. Using Visual Studio 2010, open the NonSSL folder as a local IIS website. Compile.
4. **MyDeliveryWeb** – This is a sample MyDelivery website for user registration, help, system control, and distribution of MyDelivery client software. This website is installed on each Front-End Processor. Modify it to suit your own website preferences. In NLM’s MyDelivery beta test, we made MyDeliveryWeb the default website on the Front-End Processor, which was accessed through the URL <http://mydelivery.nlm.nih.gov>. This website uses the Microsoft AJAX control toolkit, so you may need to install that on your computer. Copy the MyDeliveryWeb folder to an appropriate location on hard disk and share it as a website named MyDeliveryWeb. Using IIS Manager, allow anonymous access to this website.

**For secure (HTTPS) use only:** If you are using SSL or TLS, you can configure MyDeliveryWeb to use secure (HTTPS) communications. Also using IIS Manager, go to the Custom Errors tab. Edit error 403;4 so that it uses the URL /NonSSL/SSLRedirect.aspx. If the website is using SSL, and a user attempts to access it using HTTP, then the NonSSL website redirects the user query to your secure website. If you are running a Government website, then you may need to run it under FIPS140 mode (TLS). To do this (only for a release, non-debug website), open Local Security Policy under Administrative Tools. Under Local Policies/Security Options, select “System cryptography: use FIPS compliant algorithms ...”. Your website will not run properly if it is set up for debug mode under FIPS cryptography; use FIPS mode only for release mode websites.

Using Visual Studio 2010, open the MyDeliveryWeb folder as a local IIS website. Compile.

5. **MyDelivery** – This is a website containing all SOAP functionality for a MyDelivery Front-End Processor. Copy this folder to an appropriate location on hard disk and share it as a website named MyDelivery. Using IIS Manager, allow anonymous access to this website.

**For secure (HTTPS) use only:** If you are using SSL or TLS, you can configure MyDelivery to use secure (HTTPS) communications. Using IIS Manager, go to the Custom Errors tab. Edit error 403;4 so that it uses the URL /NonSSL/SSLRedirect.aspx. If the website is using SSL, and a user

attempts to access it using HTTP, then the NonSSL website redirects the user query to your secure website. On the Secure Communications tab, select Require Secure Channel. Also, select Require 128 bit SSL. If you are running a Government website, then you may need to run it under FIPS140 mode (TLS). To do this (only for a release, non-debug website), open Local Security Policy under Administrative Tools. Under Local Policies/Security Options, select "System cryptography: use FIPS compliant algorithms ...". Your website will not run properly if it is set up for debug mode under FIPS cryptography; use FIPS mode only for release mode websites.

Using Visual Studio 2010, open the MyDelivery folder as a local IIS website. Compile.

Finally, if you will be running your Front-End processors using SSL or TLS, you will need to separately procure and install the appropriate server certificate on each Front-End Processor. A description of how to do that is out of the scope of this document.