

AGNISSM – A Growable Network Information SystemSM

AGNIS Quick Start

Table of Contents

AGNIS Quick Start.....	1
1 Software Prerequisites.....	2
1.1 Info-ZIP.....	2
1.2 Java.....	2
1.3 Apache Ant.....	2
1.4 Globus Toolkit (Java WS Core).....	2
1.5 AGNIS Code.....	3
2 Installation.....	4
2.1 Install Apache Ant.....	4
2.2 Install Globus Toolkit (Java WS Core).....	4
2.3 Install AGNIS Code.....	5
3 Compilation.....	6
3.1 Set Environment Variables.....	6
3.2 Compile AGNIS Code.....	6
4 Certificates and Authentication.....	8
4.1 Generating a Certificate Request.....	8
5 Execution.....	13
5.1 Initialize Grid Proxy.....	13
5.2 Run the Client.....	13
6 Staging Client.....	16
6.1 Configure Staging Client.....	16
6.2 Add JDBC Driver.....	17
6.3 Create Staging Database.....	17
6.4 Database Options.....	18
6.5 Run Staging Client.....	18
7 Developing a Customized Client.....	19
8 Modifying the AGNIS Service Code.....	21
9 Further Information.....	22

AGNIS Quick Start

This document describes basic steps for using the freely provided AGNIS software to connect with the AGNIS development server hosted by the National Marrow Donor Program® (NMDP®). The AGNIS development server at the NMDP enables forms data to be electronically submitted to the FormsNet test system, and also enables retrieval of forms data published by FormsNet.

Trademark Information

A Growable Network Information System and AGNIS are service marks of the National Marrow Donor Program.

National Marrow Donor Program and NMDP are registered trademarks of the National Marrow Donor Program.

FormsNet is a trademark of the National Marrow Donor Program.

Other product or company names mentioned herein are the trademarks of their respective owners.

1 Software Prerequisites

Third-party software required by the provided AGNIS software includes Java, Apache Ant, and the portion of the Globus Toolkit known as Java WS Core. The software is implemented in Java, and should be capable of running under any operating system for which a Java 5.0 JDK is available. This includes UNIX-like operating systems such as Solaris or Linux, and Windows variants such as Windows XP.

1.1 Info-ZIP

Much of the software required by AGNIS is distributed as ZIP archive files. Most UNIX-like operating systems include an unzip program capable of extracting the contents of a ZIP archive file. Similarly, the Windows Explorer program included with recent Windows versions is also capable of extracting the contents of a ZIP archive file.

Users of an operating system which does not provide a built-in unzip function would need to install software such as Info-ZIP, a free set of tools for working with ZIP archives. The main Info-ZIP web site is here:

<http://www.info-zip.org/>

To install Info-ZIP tools, please refer to the web site for platform-specific installation information.

1.2 Java

Sun Java 5.0, also known as Java 1.5, is recommended. Later Java releases may also work. The Globus Toolkit, an important AGNIS component, is known to have problems with GJC (<http://gcc.gnu.org/java/>), but is compatible with Java implementations provided by Sun, IBM, HP, and BEA. Sun Java 5.0 is available here:

http://java.sun.com/javase/downloads/index_jdk5.jsp

To install Java, please refer to the provider's instructions.

1.3 Apache Ant

Ant 1.6 is recommended. Later Ant releases may also work. Ant 1.6 is available here:

<http://archive.apache.org/dist/ant/binaries/>

An example showing how to install Apache Ant is provided later in this document.

1.4 Globus Toolkit (Java WS Core)

To use the AGNIS Staging Client, only the Java portion of Globus Toolkit, known as Java WS Core, is required. Java WS Core 4.0 is recommended. Later Globus Toolkit versions may also work. At the time

AGNIS Quick Start

of this writing, the latest stable release of the Globus Toolkit is 4.0.5. Java WS Core 4.0.5 is available here:

http://www.globus.org/toolkit/downloads/4.0.5/#wscore_bin

An example showing how to install Java WS Core is provided later in this document.

1.5 AGNIS Code

The AGNIS code is distributed in two zip files. The `agnis-formhandler-{version}.zip` file contains the client and service code, and the `agnis-common-{version}.zip` file contains source code for the binary `agnis-common.jar` Java archive (JAR) file, which is included inside `agnis-formhandler-{version}.zip`.

Only the `agnis-formhandler-{version}.zip` file is required in order to run the standard AGNIS client programs. The `agnis-common-{version}.zip` file would be useful for working with the business logic code in `agnis-common.jar`, but is otherwise not required. The AGNIS code is available here:

<http://www.agnis.net/>

Information on installing, compiling, and configuring an AGNIS client is provided later in this document.

2 Installation

This section uses examples to illustrate software installation and setup of an environment for working with the AGNIS code.

The Windows examples show entry of commands into a command prompt window. To open a command prompt window, click the Windows Start button, select Run..., type "cmd.exe", and click OK.

For brevity, the Windows commands shown depict use of the Info-ZIP unzip program to extract the contents of zip archives. Depending on the version of Windows being used, it may also be possible to extract the contents of a zip archive file by using Windows Explorer. To access the Extraction Wizard via Windows Explorer, right-click the Windows Start button, click Explore, locate the zip file, right-click on the file and select Extract All..., then follow the wizard steps to select an extraction location and extract the zip file's contents.

2.1 Install Apache Ant

To install Ant, unzip the `apache-ant-1.6.5-bin.zip` zip file to a convenient location, such as `$HOME/agnis` (unix), or `C:\agnis` (Windows). Examples shown below.

Unix:

```
~> mkdir $HOME/agnis
~> cd $HOME/agnis
~/agnis> unzip ../apache-ant-1.6.5-bin.zip
...
```

Windows:

```
C:\> mkdir \agnis
C:\> cd \agnis
C:\> unzip ..\apache-ant-1.6.5-bin.zip
...
```

2.2 Install Globus Toolkit (Java WS Core)

To install Java WS Core, unzip the `ws-core-4.0.5-bin.zip` zip file to a convenient location, such as `$HOME/agnis` (unix), or `C:\agnis` (Windows). Additionally, for unix, manually modify the executable flags of extracted files in the `bin` subdirectory. Examples shown below.

Unix:

```
~> mkdir $HOME/agnis
~> cd $HOME/agnis
~/agnis> unzip ../ws-core-4.0.5-bin.zip
...
~/agnis> chmod 775 ws-core-4.0.5/bin/*
~/agnis> chmod 664 ws-core-4.0.5/bin/*.bat
```

AGNIS Quick Start

Windows:

```
C:\> mkdir \agnis
C:\> cd \agnis
C:\> unzip ..\ws-core-4.0.5-bin.zip
...
```

2.3 Install AGNIS Code

To install the AGNIS code, unzip the `agnis-formhandler-{version}.zip` zip file to a convenient location, such as `$HOME/agnis` (unix), or `C:\agnis` (Windows). Examples shown below.

Unix:

```
~> mkdir $HOME/agnis
~> cd $HOME/agnis
~/agnis> unzip ../agnis-formhandler-1.0.1.zip
...
```

Windows:

```
C:\> mkdir \agnis
C:\> cd \agnis
C:\> unzip ..\agnis-formhandler-1.0.1.zip
...
```

The `agnis-common-{version}.zip` zip file could also be unzipped to the same location. The contents of the "common" zip file are not required, but would be useful if there is a need to work with the source code for the `agnis-common.jar` file.

3 Compilation

The AGNIS software is distributed in source code format, and needs to be compiled before it can be used.

3.1 Set Environment Variables

The Ant script for compiling the AGNIS code requires the `ANT_HOME`, `JAVA_HOME`, and `GLOBUS_LOCATION` environment variables to be set to appropriate values. `ANT_HOME` specifies the location where Apache Ant is installed, `JAVA_HOME` specifies the location where Java is installed, and `GLOBUS_LOCATION` specifies the location where the Globus Toolkit (Java WS Core) is installed. Modifying the `PATH` environment variable to include Ant, Java, and Globus executables is also helpful. Examples shown below.

Unix (csh):

```
~/agnis> setenv ANT_HOME $HOME/agnis/apache-ant-1.6.5
~/agnis> setenv JAVA_HOME $HOME/agnis/jdk1.5.0_12
~/agnis> setenv GLOBUS_LOCATION $HOME/agnis/ws-core-4.0.5
~/agnis> setenv PATH $ANT_HOME/bin:$JAVA_HOME/bin:$GLOBUS_LOCATION/bin:$PATH
```

Windows:

```
C:\agnis> set ANT_HOME=C:\agnis\apache-ant-1.6.5
C:\agnis> set JAVA_HOME=C:\Program Files\Java\jdk1.5.0_12
C:\agnis> set GLOBUS_LOCATION=C:\agnis\ws-core-4.0.5
C:\agnis> set PATH=%ANT_HOME%\bin;%JAVA_HOME%\bin;%GLOBUS_LOCATION%\bin;%PATH%
```

Make careful note of the environment variable settings shown above. These same settings are also used when executing AGNIS client programs and supporting Globus software. Creating a script, or otherwise modifying the operating environment to automatically set these environment variables is recommended.

3.2 Compile AGNIS Code

Move to the directory containing the AGNIS code and use Ant to compile it ("ant clean createDeploymentGar"). Examples below.

Unix:

```
~> cd $HOME/agnis/grid-service/FormHandler
~/agnis/grid-service/FormHandler> ant clean createDeploymentGar
...
```

Windows:

```
C:\> cd \agnis\grid-service\FormHandler
C:\agnis\grid-service\FormHandler> ant clean createDeploymentGar
...
```

AGNIS Quick Start

The Ant script should run for a while and end with a message that says, "BUILD SUCCESSFUL". This indicates the AGNIS client program is compiled and ready for use. However additional configuration and setup is still needed in order to access the NMDP's secure AGNIS server.

4 Certificates and Authentication

AGNIS security is based on the Globus Security Infrastructure (GSI), which uses client-side X.509 certificates to authenticate users. To access secure services running on the AGNIS development server at the NMDP, the client needs to obtain a valid client-side certificate. The client also needs a copy of the AGNIS Certificate Authority (CA) certificate.

The basic procedure for obtaining an AGNIS client-side certificate, and the AGNIS CA certificate, is:

1. User generates certificate request.
2. User transmits public key component of certificate request (`usercert_request.pem` file) to AGNIS administrators.
3. AGNIS administrator manually verifies user's identity and digitally signs public key, thereby creating signed user certificate.
4. AGNIS administrator transmits signed user certificate (`usercert.pem` file) to user. AGNIS administrator also transmits CA certificate (`{ca_hash}.0` file) to user.
5. User copies certificate files to appropriate locations for use with AGNIS client software.

IMPORTANT

The generated certificate request includes both a public key and a private key. Typically, the public key is stored in a file named `usercert_request.pem`, and the private key is stored in a file named `userkey.pem`. As implied by its name, the public key is intended to be public knowledge. However, the private key *must* be kept secret.

Because **anyone who obtains a copy of the private key could use it to impersonate you or your organization**, the private key file (`userkey.pem`) must be kept in a safe place, only accessible by trusted personnel. If an untrusted party ever obtains a copy of your private key, please notify AGNIS administrators immediately to have the certificate revoked.

4.1 Generating a Certificate Request

This section describes various methods for generating an AGNIS user certificate request and using that to obtain a signed certificate.

4.1.1 Globus Toolkit

The `grid-cert-request` command, provided by the Globus Toolkit, is capable of generating an AGNIS user certificate request. When generating a user certificate request, it creates three output files: `usercert_request.pem`, `userkey.pem`, and (zero-length) `usercert.pem`.

Those three files are normally stored in the `.globus` subdirectory of the location specified by the Java `user.home` system property. Under unix, this typically is the `$HOME/.globus` directory, and under Windows, it typically is the `C:\Documents and Settings\%USERNAME%\globus` directory.

AGNIS Quick Start

To use the `grid-cert-request` command, first open a command prompt session and set environment variables as shown in the section 3.1, above. Then, execute the `grid-cert-request` command, using command line options to control various aspects of its execution, as shown below. Usage of `grid-cert-request` is nearly identical under either unix or Windows.

The following example shows the `grid-cert-request` command being used to generate an certificate request for the development AGNIS server hosted by the NMDP.

```
C:\> grid-cert-request -caEmail agnis@nmdp.org
      -orgBaseDN "o=Grid,ou=GlobusTest,ou=simpleCA-yew"
      -cn "My Name"
A certificate request and private key will be created.
You will be asked to enter a PEM pass phrase.
This pass phrase is akin to your account password,
and is used to protect your key file.
If you forget your pass phrase, you will need to
obtain a new certificate.

Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
Generating a 1024 bit RSA private key
A private key and a certificate request has been generated with the subject:

/O=Grid/OU=GlobusTest/OU=simpleCA-yew/CN=My Name

The private key is stored in C:\Documents and Settings\username\globus\userkey.pem
The request is stored in C:\Documents and Settings\username\globus\usercert_request.pem
```

The `grid-cert-request` command uses a pass phrase to encrypt the certificate's private key. It prompts for the pass phrase twice, to verify it was entered accurately.

The certificate subject is a name which uniquely identifies the certificate. The subject is a concatenation of the organization's base distinguished name (DN) and the user's common name (CN).

In the above example, the organization base DN and subject CN are specified via the `-orgBaseDN` and `-cn` command line parameters, respectively. Additionally, the email address for the CA is specified via the `-caEmail` command line parameter. The base DN recommended for use with the NMDP's development AGNIS server is:

```
"o=Grid,ou=GlobusTest,ou=simpleCA-yew"
```

The CN should be set to a name which identifies the user of the certificate.

Additional information on `grid-cert-request` is available from the Globus web site:

<http://www.globus.org/toolkit/docs/4.0/security/prewsaa/rn01re02.html>.

After generating the certificate request, email the `usercert_request.pem` file to `agnis@nmdp.org` and, upon approval of the request, store the signed certificate and CA certificate as described in the *Obtain Signed Certificate* section, below.

AGNIS Quick Start

4.1.2 OpenSSL

This section describes a procedure for using OpenSSL to generate an X.509 certificate request compatible with the AGNIS development environment.

The OpenSSL command used here references a configuration file. An OpenSSL configuration file suitable for use with the AGNIS development environment is shown below.

```
[ req ]
default_bits          = 1024
distinguished_name    = req_distinguished_name
req_extensions        = v3_req

[ req_distinguished_name ]
0.organizationName    = Level 0 Organization
0.organizationName_default = Grid
0.organizationalUnitName = Level 0 Organizational Unit
0.organizationalUnitName_default = GlobusTest
1.organizationalUnitName = Level 1 Organizational Unit
1.organizationalUnitName_default = simpleCA-yew
commonName            = Name (e.g., John M. Smith)
commonName_max        = 64

[ v3_req ]
nsCertType            = objsign,email,server,client
basicConstraints      = critical,CA:false
```

The nsCertType extension used by the above configuration has been deprecated by OpenSSL. It may instead be preferable to specify keyUsage and extendedKeyUsage, as shown here.

```
[ v3_req ]
basicConstraints = critical,CA:false
keyUsage         = keyAgreement,dataEncipherment,keyEncipherment,digitalSignature
extendedKeyUsage = serverAuth,clientAuth,codeSigning,emailProtection,timeStamping
```

The following example illustrates generation of a user certificate request by directly using OpenSSL. The openssl command shown here uses the globus-ssl.conf configuration file described above. At the Name prompt, enter a name which identifies the user of the certificate.

```
C:\> openssl req -new -keyout userkey.pem -out usercert_request.pem
    -config globus-ssl.conf
Generating a 1024 bit RSA private key
...+++++
.....+++++
writing new private key to 'userkey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
```

AGNIS Quick Start

```
Level 0 Organization [Grid]:
Level 0 Organizational Unit [GlobusTest]:
Level 1 Organizational Unit [simpleCA-yew]:
Name (e.g., John M. Smith) []:My Name
C:\>
```

The `openssl` command uses a pass phrase to encrypt the certificate's private key. It prompts for the pass phrase twice, to verify it was entered accurately.

Additional information on OpenSSL is available from the OpenSSL web site:

<http://www.openssl.org/>

If possible, set file permissions to limit access to the `userkey.pem` file, which contains your private key.

Unix:

```
~/globus > chmod 600 userkey.pem
```

Create a `.globus` directory and move the `userkey.pem` and `usercert_request.pem` files to that location.

Unix:

```
~> mkdir $HOME/.globus
~> mv userkey.pem $HOME/.globus
~> mv usercert_request.pem $HOME/.globus
```

Windows:

```
C:\> mkdir "%Documents and Settings\kbengtss\.globus"
C:\> move userkey.pem "%Documents and Settings\kbengtss\.globus"
C:\> move usercert_request.pem "%Documents and Settings\kbengtss\.globus"
```

After generating the certificate request, email the `usercert_request.pem` file to `agnis@nmdp.org` and, upon approval of the request, store the signed certificate and CA certificate as described in the *Obtain Signed Certificate* section, below.

4.1.3 AGNIS Admin Interface

Usage of the AGNIS Admin Interface to generate an AGNIS user certificate request is not recommended at this time.

4.1.4 Pass Phrase Removal

Optionally, some sites may prefer to access the private key without using a pass phrase. In that case, it is necessary to obtain an unencrypted copy of the private key file. This can be done using OpenSSL, with a command such as the following.

AGNIS Quick Start

```
> openssl rsa -in userkey.pem -inform PEM -out userkey_nopass.pem -outform PEM
Enter pass phrase for userkey.pem:
writing RSA key
```

4.1.5 Obtain Signed Certificate

After generating the user certificate request, email a copy of the `usercert_request.pem` file to the AGNIS administrators at `agnis@nmdp.org`. When a request is approved, the AGNIS administrator sends back two files: a `usercert.pem` file containing the signed user certificate, and a `{ca_hash}.0` file containing a copy of the CA certificate. These files should be copied to the `.globus` and `.globus/certificates` directories, respectively, as shown below.

Unix:

```
~> cp usercert.pem $HOME/.globus
~> mkdir $HOME/.globus/certificates
~> cp 5f925b36.r0 $HOME/.globus/certificates
```

Windows:

```
C:\> copy usercert.pem "%Documents and Settings\kbengtss\.globus"
C:\> mkdir "%Documents and Settings\%USERNAME%\\.globus\certificates"
C:\> copy 5f925b36.r0 "%Documents and Settings\kbengtss\.globus\certificates"
```

5 Execution

The provided software includes an AGNIS client program which is capable of connecting to the development AGNIS server hosted at the NMDP. The basic procedure for running the client program is:

1. Set environment variables (see section 3.1, above).
2. Compile the code (see section 3.2, above).
3. Initialize a grid proxy (see section 5.1, below).
4. Run the client (see section 5.2, below).

The software also includes a `README.txt` document, which provides additional details and examples.

5.1 Initialize Grid Proxy

Use the `grid-proxy-init` program to generate a temporary grid credential, known as a "proxy certificate". The client needs this credential to authenticate with the AGNIS service. Example:

```
C:\> grid-proxy-init
Your identity: O=Grid,OU=GlobusTest,OU=simpleCA-yew,CN=My Name
Enter GRID pass phrase for this identity:
```

```
C:\> grid-proxy-init
Your identity: O=Grid,OU=GlobusTest,OU=simpleCA-yew,CN=My Name
Enter GRID pass phrase for this identity:
  Creating proxy, please wait...
Proxy verify OK
Your proxy is valid until Fri Oct 26 04:28:52 CDT 2007
Warning: Please check file permissions for your proxy file.
```

The `grid-proxy-init` command uses credentials from the `.globus` subdirectory, so those need to be set up as described in section 4 of this document before `grid-proxy-init` can be used.

5.2 Run the Client

Move to the directory containing the `FormHandler` code, and use the `"ant runClient"` command to run the client. The client exercises each operation offered by the AGNIS service. To change the service URL, edit the `run-tools.xml` file and modify the `service.url` property in the `runClient` target.

Service operations called by the client include: `requestSiteCertificate`, `acknowledgeRetrieval`, `acknowledgeSingleRetrievedFormRevision`, `getNewlyCompletedFormRevisionQuantity`, `publishCompletedFormRevision`, `retrieveNewlyCompletedFormRevisions`, `retrieveSingleCompletedFormRevision`, and `submitFormRevision`. Most operations are called twice; the first call is normally successful, and the second normally generates a "not authorized" exception.

AGNIS Quick Start

The source code for this client program is in the `FormHandlerClient.java` file, located in the `src\net\agnis\grid\client` subdirectory. That Java file contains generated code, so the preferred method for creating a customized client program is by extending the `FormHandlerClient` class, instead of directly modifying that file.

Example client output:

```
C:\agnis\grid-service\FormHandler>ant runClient
Buildfile: build.xml

setGlobus:

checkGlobus:
    [echo] Globus: C:\ws-core-4.0.5

defineClasspaths:

defineExtendedClasspaths:

runClient:
    [echo] Connecting to service:
    [echo] https://yew.nmdp.org:8443/wsrf/services/agnis/FormHandler
    [java] JVM args ignored when same JVM is used.
    [java] Running the Grid Service Client

    [java] Operation:  requestSiteCertificate

    [java] Operation:  acknowledgeRetrieval
    [java] Operation successful.
    [java] Now triggering RemoteException.
    [java] RemoteException received.

    [java] Operation:  acknowledgeSingleRetrievedFormRevision
    [java] Operation successful.
    [java] Now triggering RemoteException.
    [java] RemoteException received.

    [java] Operation:  getNewlyCompletedFormRevisionQuantity
    [java] Return value: 1
    [java] Operation successful.
    [java] Now triggering RemoteException.
    [java] RemoteException received.

    [java] Operation:  publishCompletedFormRevision
    [java] Operation successful.
    [java] Return value: 1
    [java] Now triggering RemoteException.
    [java] RemoteException received.

    [java] Operation:  retrieveNewlyCompletedFormRevisions
    [java] Writing serialized object to output file: Retrieval.xml
    [java] Operation successful.
    [java] Now triggering RemoteException.
    [java] RemoteException received.

    [java] Operation:  retrieveSingleCompletedFormRevision
    [java] Writing serialized object to output file: FormRevision.xml
```

AGNIS Quick Start

```
[java] Operation successful.  
[java] Now triggering RemoteException.  
[java] RemoteException received.
```

```
[java] Operation: submitFormRevision  
[java] Writing serialized object to output file: SubmitFormRevision.xml  
[java] Operation successful.  
[java] Now triggering RemoteException.  
[java] RemoteException received.
```

```
BUILD SUCCESSFUL  
Total time: 33 seconds  
C:\agnis\grid-service\FormHandler>
```

6 Staging Client

The staging client is a client program capable of replicating forms data retrieved from AGNIS and in a local database. Source code for the staging client is in the `StagingClient.java` file, located in the `src\net\agnis\grid\client` subdirectory. The `StagingClient` class extends from the base `FormHandlerClient` class.

The procedure for running the staging client is similar to that for running basic client:

1. Set environment variables (see section 3.1, above).
2. Compile the code (see section 3.2, above).
3. Initialize a grid proxy (see section 5.1, above).
4. Run the client (see section 6.5, below).

However, the staging client uses a local database, and requires some additional setup and configuration, as described in the following sections.

6.1 Configure Staging Client

Edit the `StagingClient.properties` file to configure the staging client. This is a plain text file, and can be edited using a regular text editor such as `vi` (unix) or Notepad (Windows).

The `stagingClient...` properties in this file determine basic operating parameters of the staging client. `stagingClientMaximumFormQuantity` specifies the maximum number of updated forms retrieved at once, `stagingClientServiceURL` specifies the URL of the AGNIS service, and `stagingClientSubscriberUniqueName` specifies the AGNIS subscriber name associated with the retrieval request. Example:

```
stagingClientMaximumFormQuantity=100
stagingClientServiceURL=https://yew.nmdp.org:8443/wsrf/services/agnis/FormHandler
stagingClientSubscriberUniqueName=mySubscriberUniqueName
```

The `jdbc...` properties define database connectivity parameters. `jdbcDriver` specifies the JDBC database driver to be used, `jdbcURL` specifies the URL for the staging database, `jdbcUser` specifies the database user name, and `jdbcPassword` specifies the database password for that user.

Example (MySQL):

```
jdbcDriver=com.mysql.jdbc.Driver
jdbcPassword=agnistest
jdbcURL=jdbc:mysql://localhost:3306/AGNIS_FORMS
jdbcUser=AGNIS_FORMS
```

Example (Oracle):

```
jdbcDriver=oracle.jdbc.OracleDriver
jdbcPassword=agnistest
jdbcURL=jdbc:oracle:thin:@localhost:1521:XE
jdbcUser=AGNIS_FORMS
```

AGNIS Quick Start

Other configuration properties in the `StagingClient.properties` file can be left unmodified.

6.2 Add JDBC Driver

The staging client uses a Java Database Connectivity (JDBC) driver to connect with the staging database. The JDBC driver for a specific database, such as MySQL or Oracle, normally is available from the database vendor, and comes packaged in a single JAR file.

To make the JDBC driver available for use by the staging client, copy the JAR file containing the JDBC driver to the `lib` subdirectory. Examples below.

MySQL:

```
C:\agnis\grid-service\FormHandler> copy \mysql\mysql-connector-java-5.0.3-bin.jar lib
```

Oracle:

```
C:\agnis\grid-service\FormHandler> copy \oracle\ojdbc14.jar lib
```

6.3 Create Staging Database

A separate Data Dictionary document describes the AGNIS repository and staging database schema. Please refer to that document for details regarding database table definitions.

The `db` subdirectory contains scripts for creating the staging database tables under MySQL or Oracle. Usage examples:

MySQL:

```
C:\agnis\grid-service\FormHandler> cd db
C:\agnis\grid-service\FormHandler\db> mysql
mysql> create database AGNIS_FORMS;
mysql> use AGNIS_FORMS;
mysql> source AGNIS_FORMS-createdb-mysql.sql;
...
```

Oracle:

```
C:\agnis\grid-service\FormHandler> cd db
C:\agnis\grid-service\FormHandler\db> sqlplus /nolog
SQL> connect AGNIS_FORMS
Enter password:
Connected.
SQL> start AGNIS_FORMS-createdb-oracle.sql
...
```

6.4 Database Options

MySQL 5 or Oracle 10g is recommended. Later MySQL or Oracle versions may also work.

MySQL 5.0 can be downloaded here:

<http://dev.mysql.com/downloads/mysql/5.0.html>

Oracle 10g Express Edition can be downloaded here:

<http://www.oracle.com/technology/software/products/database/xe/index.html>

To install database software, please refer to the vendor's platform-specific installation instructions.

6.5 Run Staging Client

Use Ant ("ant runStagingClient") to execute the AGNIS Staging Client program. The staging client retrieves newly updated form revisions from the AGNIS service, and stores them in the staging database, as configured by the `StagingClient.properties` file. Example:

```
C:\agnis\grid-service\FormHandler> ant runStagingClient
Buildfile: build.xml

setGlobus:

checkGlobus:
    [echo] Globus: C:\ws-core-4.0.5

defineClasspaths:

defineExtendedClasspaths:

runStagingClient:
    [java] Reading configuration file.
    [java] Initializing web service client.
    [java] Service URL: https://yew.nmdp.org:8443/wsrf/services/agnis/FormHand
ler
    [java] Retrieving updates from web service.
    [java] 1 updated FormRevision(s) retrieved.
    [java] Storing updates in staging database.
    [java] 1 updated FormRevision(s) stored.
    [java] Acknowledging retrieval.
    [java] Done.

BUILD SUCCESSFUL
Total time: 11 seconds
```

7 Developing a Customized Client

The most straightforward way to develop a customized AGNIS client is by extending the basic `FormHandlerClient` class, as illustrated by the following Java code snippet.

```
package com.aaa.bbb;

import net.agnis.grid.client.FormHandlerClient;

public class CustomClient
extends FormHandlerClient
{
    // custom code ...
}
```

A custom client such as the above inherits all the `FormHandlerClient` methods for communicating with the AGNIS service, which simplifies the task of developing custom code which communicates with the AGNIS server. `StagingClient` is an example of a custom client which extends from `FormHandlerClient` in this way. To make the custom client executable by Ant, edit the `run-tools.xml` file and add a new target such as the following.

```
<target name="runCustomClient" depends="checkGlobus, defineClasspaths"
description="Run the custom client">
    <java classname="com.aaa.bbb.CustomClient" classpathref="run.classpath"
fork="yes" failonerror="true">
        <jvmarg value="-DGLOBUS_LOCATION=${ext.globus.dir}" />
    </java>
</target>
```

The AGNIS service operations can be divided into four general categories.

Category	Operation(s)
Set up user account	<code>requestSiteCertificate</code>
Publish completed forms	<code>publishCompletedFormRevision</code>
Submit new forms	<code>submitFormRevision</code>
Retrieve completed forms	<code>acknowledgeRetrieval</code> , <code>acknowledgeSingleRetrievedFormRevision</code> , <code>getNewlyCompletedFormRevisionQuantity</code> , <code>retrieveNewlyCompletedFormRevisions</code> , <code>retrieveSingleCompletedFormRevision</code>

The `requestSiteCertificate`, `publishCompletedFormRevision`, `submitFormRevision` operations are relatively self explanatory, but the AGNIS forms retrieval process requires some further explanation. The provided `StagingClient` illustrates the typical process for retrieving completed forms from AGNIS:

AGNIS Quick Start

1. Invoke `retrieveNewlyCompletedFormRevisions` to retrieve a group of new or updated forms.
2. Process the retrieved forms.
3. Invoke `acknowledgeRetrieval` to acknowledge that the retrieval was successful.

Other retrieval operations would only be used under special circumstances.

Forms data transmitted via AGNIS uses metadata (Common Data Elements, or CDEs) defined in the Cancer Data Standards Repository (caDSR). FormsNet forms are defined in caDSR using the Form Builder tool. Further information on caDSR is available from the caDSR web site.

http://ncicb.nci.nih.gov/NCICB/infrastructure/cacore_overview/cadsr

When communicating with AGNIS, users will typically need to translate data from local formats to caDSR value domains, and vice-versa.

The AGNIS service follows web service standards, and is interoperable with non-Java programming environments such as Microsoft .NET and Ruby. Full coverage of those environments is beyond the scope of this document, but some additional information can be found in the following entry from the gt-user mailing list archive:

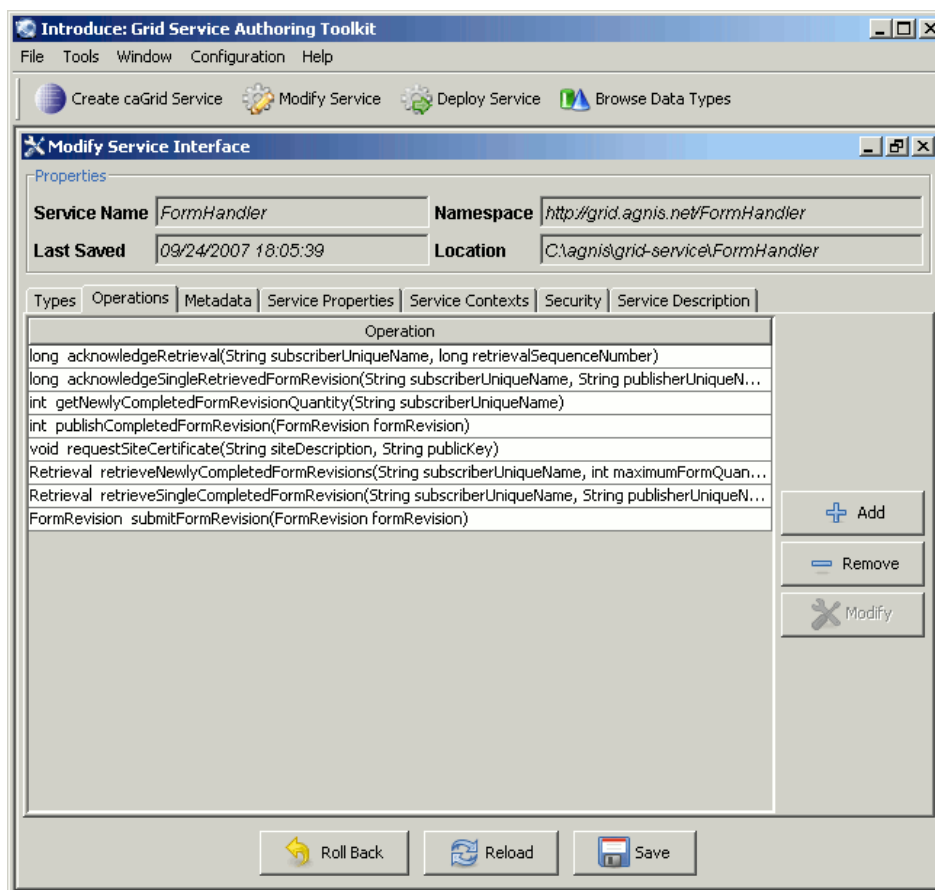
http://www.globus.org/mail_archive/gt-user/2007/05/msg00034.html

8 Modifying the AGNIS Service Code

Much of the AGNIS code in the "formhandler" zip file was generated and is being maintained using the Introduce tool from caGrid 1.1. The caGrid home page has further information on Introduce and caGrid:

<https://cabig.nci.nih.gov/workspaces/Architecture/caGrid>

Those developing AGNIS client software will not normally need to use Introduce to modify the code. However, for informational purposes, here is a screen shot of the Introduce interface for defining service operations.



9 Further Information

AGNIS Web Site
<http://www.agnis.net/>

Apache Ant
<http://ant.apache.org/>

caDSR
http://ncicb.nci.nih.gov/NCICB/infrastructure/cacore_overview/cadsr

caDSR CDE Browser
<http://cdebrowser.nci.nih.gov/CDEBrowser/>

caDSR Form Builder
<http://cdebrowser.nci.nih.gov/CDEBrowser/formSearchAction.do>

caGrid
<https://cabig.nci.nih.gov/workspaces/Architecture/caGrid>

Eclipse
<http://www.eclipse.org/>

Globus
<http://www.globus.org/>

Globus Tutorial
<http://gdp.globus.org/gt4-tutorial/>

Java
<http://java.sun.com/>