The Geodise Toolboxes

Installing the Geodise XML, Compute and Database Toolboxes

Geodise

Release: GeodiseLab v.1.3.2

Version: INSTALL.doc 1.3.7

Title: The Geodise Toolboxes – Installation Instructions

Authors: Dr Graeme Pound
Dr Jasmin Wason
Dr Marc Molinari, m.molinari@soton.ac.uk
Dr Hakki Eres, hakki.eres@soton.ac.uk
Dr Zhuoan Jiao, z.jiao@soton.ac.uk
Dr Andrew Price, a.r.price@soton.ac.uk

PI: Prof Simon Cox, s.j.cox@soton.ac.uk

Web: http://www.geodise.org/

Contents

Prerequisites

The prerequisite required by the Geodise Toolboxes is any of the supported versions of Matlab:

- Matlab 6.5 (release 13)
- Matlab 7.0 (release 14)
- Matlab 7.0.1 (release 14 Service Pack 1)
- Matlab 7.0.4 (release 14 Service Pack 2)
- Matlab 7.1.0 (release 14 Service Pack 3)
- Matlab 7.2.0 (release 2006a)
- Matlab 7.3.0 (release 2006b)

Installing the GeodiseLab toolboxes

To install the GeodiseLab toolboxes for Matlab it is necessary to add the several Java libraries to the Matlab classpath, and to make the GeodiseLab functions available to the Matlab path.

Further configuration will be required to use the Geodise Compute and Database Toolboxes.

Extract the toolboxes

Unzip 'GeodiseLab-1.3.2.zip' to a location <GEODISELABBASE>.

The GeodiseLab distribution contains toolboxes compiled for Matlab 6.5 (release 13), Matlab 7 (release 14) and Matlab 7.0.4 (release 14 service pack 2). These releases reflect the version of the Java Virtual Machine upon which Matlab operates (Java 1.3.1, 1.4.2 and 1.5.0 respectively). Install the GeodiseLab toolboxes from the directory corresponding to the correct version of Matlab, either 'GeodiseLab-matlabR13/', 'GeodiseLab-matlabR14/' or 'GeodiseLab-matlabR14SP2'. This directory will be referred to as <GEODISELAB> throughout this document.

Install the GeodiseLab toolboxes

Do either of the following:

(a) Run the installation script

Navigate to the directory <GEODISELAB> and run the installation script. On Windows:

```
>> install.bat
```

On Linux:

```
>> ./install.sh
```

The user will be queried for the location of the directory containing the Matlab executable, and the installation directory to which the GeodiseLab Toolboxes will be copied. An example of the installation of the GeodiseLab Toolboxes for Matlab 7 on Windows XP is shown below:

```
>> install.bat
```

```
Buildfile: build.xml


init:


query_matlabhome:
    [input] Please enter the path to the directory
containing the Matlab executable (for example
'/usr/local/matlab6p5/bin/glnx86/' or
'c:\matlab6p5\bin\win32\'):
```

```
d:\matlab7\bin\win32\
```

```
check_matlabhome:


query_installpath:
    [input] Please enter the path of the installation
directory [default C:\Documents and
Settings\gep\GeodiseLab\]:
```

```
c:\GeodiseLab\
```

```
test_installpath:


check_installpath:


delete_installpath:


make_installpath:
    [mkdir] Created dir: C:\GeodiseLab
     [copy] Copying 255 files to C:\GeodiseLab


local_install:
     [echo] local installation
     [copy] Copying 1 file to C:\GeodiseLab
    [touch] Creating C:\GeodiseLab\startup.m
    [touch] Creating C:\GeodiseLab\GeodiseLab.properties
    [touch] Creating C:\GeodiseLab\uninstall.bat


shared_install:


make_shortcut:
[win32.shortcut] Creating shortcut C:\GeodiseLab\Matlab -
Geodise Toolboxes.lnk
[win32.shortcut] Creating shortcut C:\Documents and
Settings\gep\Desktop\Matlab - Geodise Toolboxes.lnk


pre-install:


BUILD SUCCESSFUL
Total time: 1 minute 13 seconds
Press any key to continue . . .
```

To use the GeodiseLab Toolboxes it is necessary to start Matlab in the installation directory. Under Windows shortcuts to start Matlab with the GeodiseLab Toolboxes are provided in the installation directory, and upon the Windows desktop. From the Linux command-line it is necessary to navigate to the installation directory before starting Matlab.

The installers also support the installation of the GeodiseLab Toolboxes into a shared Matlab installation. In this case the classpath and search path of the shared Matlab installation are edited and the changes will affect all users. For a shared installation the following commands should be used. On Windows:

```
>> install.bat -Dshared.installation=true
```
On Linux:
```
>> ./install.sh -Dshared.installation=true
```

(b) Manually install the toolboxes

Set the Matlab classpath

The Java libraries used by the GeodiseLab Toolboxes must be made available via the Matlab classpath. This is set in the file 'classpath.txt' in the directory '$MATLABROOT/toolbox/local/'. However on a shared system changes made to this file will affect all users, it may be preferable to edit a copy of the original 'classpath.txt' kept in the directory from which Matlab is started. For more information see:

http://www.mathworks.com/access/helpdesk_r13/help/techdoc/matlab_external/ch05mat4.html

It is necessary to restart Matlab for changes to the classpath to take effect.

```
<GEODISELAB>/ComputeToolbox/lib/cog-jglobus.jar
<GEODISELAB>/ComputeToolbox/lib/cog-lib.jar
<GEODISELAB>/ComputeToolbox/lib/cryptix-asn1.jar
<GEODISELAB>/ComputeToolbox/lib/cryptix.jar
<GEODISELAB>/ComputeToolbox/lib/cryptix32.jar
<GEODISELAB>/ComputeToolbox/lib/jce-jdk13-120.jar
<GEODISELAB>/ComputeToolbox/lib/jgss.jar
<GEODISELAB>/ComputeToolbox/lib/junit.jar
<GEODISELAB>/ComputeToolbox/lib/log4j-1.2.8.jar
<GEODISELAB>/ComputeToolbox/lib/puretls.jar
<GEODISELAB>/DatabaseToolbox/lib/axis.jar
<GEODISELAB>/DatabaseToolbox/lib/commons-discovery.jar
<GEODISELAB>/DatabaseToolbox/lib/commons-logging.jar
<GEODISELAB>/DatabaseToolbox/lib/dom4j.jar
<GEODISELAB>/DatabaseToolbox/lib/jakarta-regexp-1.3.jar
<GEODISELAB>/DatabaseToolbox/lib/jaxrpc.jar
<GEODISELAB>/DatabaseToolbox/lib/jcert.jar
```

```
<GEODISELAB>/DatabaseToolbox/lib/jnet.jar
<GEODISELAB>/DatabaseToolbox/lib/jsr173_1.0_api.jar
<GEODISELAB>/DatabaseToolbox/lib/jsr173_1.0_ri.jar
<GEODISELAB>/DatabaseToolbox/lib/jsse.jar
<GEODISELAB>/DatabaseToolbox/lib/jug.jar
<GEODISELAB>/DatabaseToolbox/lib/saaj.jar
<GEODISELAB>/DatabaseToolbox/lib/sunjce_provider.jar
<GEODISELAB>/DatabaseToolbox/lib/wsdl4j.jar
<GEODISELAB>/DatabaseToolbox/lib/xalan.jar
<GEODISELAB>/DatabaseToolbox/lib/xercesImpl.jar
<GEODISELAB>/DatabaseToolbox/lib/xml-apis.jar
<GEODISELAB>/DatabaseToolbox/lib/xmlParserAPIs.jar
<GEODISELAB>/DatabaseToolbox/lib/xmlsec.jar
```

*** IMPORTANT ***

The ComputeToolbox provides a SSL library as part of the Java CoG. This library (cryptix32.jar) needs to take precedence over the library provided by Matlab (icessl.jar) in R14 releases. For R14+ installations please move the following entry to the bottom of the classpath.txt file:

```
$matlabroot/java/jarext/ice/icessl.jar
```

Set the Matlab search path

Add the directories containing the Compute, Database and XML Toolboxes to the Matlab search path. There are a number of methods to add these .m files to the path:

1) If using the Matlab desktop, navigate to the 'Set Path' dialog ('File' > 'Set Path'). Click the 'Add Folder' button and browse to the directory containing the Compute, Database and XML Toolboxes, select 'OK' to confirm. You may wish to click the 'Save' button to preserve the configuration between sessions. Click 'Close' to dismiss the dialog.

2) If you are using Matlab via the Unix terminal you can instead use the 'addpath' and 'savepath' functions at the Matlab command line.

```
>> addpath <GEODISELAB>/XMLToolbox
>> addpath <GEODISELAB>/ComputeToolbox
>> addpath <GEODISELAB>/DatabaseToolbox
```

3) System administrators configuring a multi-user Matlab installation may find it preferable to edit '$MATLABROOT/toolbox/local/pathdef.m' to make changes to the Matlab search path available to all users.

Uninstalling the GeodiseLab toolboxes

The uninstall script may be invoked to uninstall the GeodiseLab toolboxes only if the installation script has previously be used to install the toolbox. Uninstall scripts are available in the both the installation and <GEODISELAB> directories. On Windows:

```
>> uninstall.bat
```

On Linux:

```
>> ./uninstall.sh
```

XML Toolbox

This version of the XML Toolbox for Matlab provides easy XML functionality to the Matlab environment. It has been tested with Matlab versions 6.5 (R13) and 7.0 (R14) (http://www.mathworks.com/). It can be used standalone and does not require Java.

Test Installation

To confirm that the Matlab search path has been successfully configured execute

```
>> str = which('xml_help')
```

at the Matlab prompt. The variable `str` should contain the path of the XML Toolbox.

From the Matlab command line run:

```
>> demo_xml
```

This script will demonstrate the functionality of the toolbox commands. For further information on the XML Toolbox use `xml_help`, or to obtain help on individual functions use Matlab's default `help` command, e.g. `help xml_format`.

As an alternative test and example, try the `xml_tests` script in the 'tests/' subdirectory. Alternatively, try to execute the following at the Matlab command prompt:

```
>> xml_format( 12345 )      or      >> xml_format( 'hello' )
```

Compute Toolbox

This version of the Geodise Compute Toolbox for Matlab provides Globus GT2 client functionality to the Matlab environment. It has been tested with Matlab version 6.5, 7, 7.0.4 and R2006a (http://www.mathworks.com/). The Geodise Compute Toolbox uses the Java CoG v1.2 (http://www.globus.org/cog/)[1].

These installation instructions will describe how to configure the Java CoG, and how to install the Geodise Compute Toolbox in the Matlab environment.

Configuration

During configuration it is necessary to create a 'cog.properties' file containing the default settings for the Java CoG, in a '.globus/' directory in the user's home directory. The user should have a valid X509 certificate from a Certificate Authority (CA) acceptable to the managers of the Globus resources that the user wishes to access. To access Globus resources the user should submit the subject line of their certificate to the resource managers to allow them to map it to a user account.

a) Create a '.globus/' directory in the user's home directory. Note that in Windows this is best done at the command line with the command 'mkdir .globus'.

b) Copy the CA certificates for the CAs that you wish to trust into the '.globus/' directory. These must include the CA certificates for the CA which signed your user certificate, as well as the CAs for any Globus resources that you wish to access. The new and legacy CA certificates for the UK eScience CA (adcbc9ef.0 and 01621954.0 respectively) are included with the Geodise Compute Toolbox distribution.

c) Copy the example 'cog.properties' file distributed with the Geodise Compute Toolbox into the '.globus' directory, and edit the values of the properties 'usercert', 'userkey', 'proxy' and 'cacert' to the correct values. The example 'cog.properties' file contains example configurations for Windows and Unix platforms.

---

[1] This product includes software developed by and/or derived from the Globus project (http://www.globus.org/).

The properties 'usercert' and 'userkey' refer to locations of the PEM encoded user certificate and corresponding private key. The file 'cacert' contains the certificate of the CA which signed the user's X.509 certificate (in PEM format). Where 'proxy' will be the location of the user's proxy certificate once it has been generated by `gd_createproxy`. The properties 'proxy.strength' and 'proxy.lifetime' contain default settings for the cryptographic strength and lifetime of the proxy certificate. Note that the file separator on a Windows PC must be defined with double backslashes, "\\".

Test Installation

To confirm that the Matlab search path has been successfully configured run:

```
>> str = which('gd_jobsubmit')
```

The variable str should contain the path of the Geodise Compute Toolbox.

From the Matlab command line run:

```
>> demo_compute
```

This script will prompt the user to enter the name of a Globus Compute resource which the user is authorised to use. GridFTP and GRAM services on this resource will then be accessed.

Troubleshooting

There are four types of error which you may receive;

a) The toolbox is incorrectly installed. If you receive an error such as;

```
??? Undefined variable 'org' or class 'org.geodise.computational.'
```

the Geodise computational toolbox classes, or any of the packages which it requires is not visible to Matlab. Check that all the .jar files listed above have been added to the Matlab classpath.

b) The Java CoG is incorrectly configured. A correct cog.properties file is required by the Java CoG to locate the user's certificate, private key and CA certificates.

c) A failure to communicate with the Globus server. Unfortunately there are

numerous ways for Globus to fall over, typically the Java CoG will return a more or less cryptic Java error. I am trying to identify and handle these errors within the Matlab code, if you have such an error please send it to me and the problem may be handled better in the next release. This webpage may help diagnose any mysterious errors;

http://www.grid-support.ac.uk/archive/starterkits/gt2errors.htm

d) A bug in the code. The test script `demo_compute.m` is designed to be flexible, but it may not be appropriate for configuration of your server, let me know if you have a problem.

Other problems that may occur

e) Matlab 7.0.1 prints Log4J logging information (Linux and Windows).
The output of some functions are unnecessarily verbose when using Matlab 7 [SP1]. To correct this comment out (#) the following line in the classpath.txt file:

```
$matlabroot/java/jarext/axis.jar
```

A copy of axis.jar which does not cause this behaviour is supplied with the Database Toolbox.

Reinstallation

If you are updating a previous installation of the Geodise Compute Toolbox you will need to remove previous changes made to the Java classpath. If the Geodise Compute Toolbox is on the Matlab search path, you may need to alter the search path to reflect the new location of the functions.

Database Toolbox

The Geodise Database Toolbox for Matlab provides client side functions for archiving, querying, retrieving, grouping, and sharing data in an archive. Metadata and Matlab structures are stored in an Oracle database accessed though Web services and files are stored on a Globus enabled server.

Requirements

The following toolboxes must be installed and added to the Matlab path before installing the Geodise Database Toolbox:

      a) The XML Toolbox for Matlab

      b) The Geodise Compute Toolbox for Matlab

Add certificate subject to database.

The default database for this distribution is hosted by the UK National Grid Service (NGS) on the CCLRC-RAL Data Cluster.

      a) Read the terms of use for the UK National Grid Service:

      http://www.ngs.ac.uk/NGS-tacu.shtml

      b) Join the NGS-ANNOUNCE mailing list to keep informed of any scheduled downtime or other issues related to the NGS RAL Oracle database.
If you have registered with the NGS you should already receive these emails. Otherwise you can join the list at:

      http://www.jiscmail.ac.uk/cgi-bin/webadmin?SUBED1=ngs-announce&A=1

      c) Contact the database administrator (z.jiao@soton.ac.uk for the NGS RAL Geodise database) to register your certificate subject so that you are authorised to store and access data in the repository. You can find this in your usercert.pem file, or by viewing the result string returned by calling `gd_certinfo` from the Compute Toolbox. Here is an example:

            /C=UK/O=eScience/OU=Southampton/L=SeSC/CN=joe bloggs

Get an account on the file store host.

You need an account on the host you will be storing files on. This can be any Globus enabled server, e.g. `grid-data.rl.ac.uk` or `pacifica.iridis.soton.ac.uk`. The administrator will need to add your certificate subject and username to the gridmap file on that machine.

Configuration

Linux C-Shell (csh or tcsh) Matlab 6.5 users should insert the following in the non-interactive section of ~/.cshrc

    setenv MATLAB_SHELL /bin/bash
    or
    setenv MATLAB_SHELL /bin/sh

Setup

Start Matlab from the directory containing classpath.txt (in Windows create a new shortcut to Matlab and set the "Start in" directory to the directory containing your classpath.txt).

Run the `gd_dbsetup` command which will create a new .geodise configuration directory in your home directory (<home_dir>) if one does not already exist.

e.g. C:\Documents and Settings\your_username\.geodise on Windows or $HOME/.geodise on Linux.

It will then copy the contents of <GEODISELAB>/DatabaseToolbox/.geodise into your <home_dir>/.geodise directory.

File Store

`gd_dbsetup` will prompt you for the following configuration properties which determine the file store host:

    hostname - a Globus enabled server.
    hostdir - a directory on that server where files can be stored.

`gd_dbsetup` saves these properties in <home_dir>/.geodise/ClientConfig.xml

    e.g.
    <fileserver>
        <hostname>grid-data.rl.ac.uk</hostname>
        <hostdir>/home/<ngs-username>/geodise-data</hostdir>
    </fileserver>

Test Installation

To confirm that the Matlab search path has been successfully configured run:

```
>> str = which('gd_archive')
```

The variable `str` should contain the path of the Geodise Database Toolbox.

Run the `demo_db` function. The first thing the function does is call `gd_createproxy` to create a proxy certificate, as this is needed to use the Geodise Database Toolbox functions.

This is the end of the installation instructions. The following sections provide information on configuration and troubleshooting.

Advanced Configuration

A remote database is used to store Matlab variables and metadata, and all database interaction is done through web services. The location of these services can also be configured in ClientConfig.xml under the <webservices> tag, although it is unlikely you will want to change the default settings for these.

```
<webservices>

    <metadataWS>https://portal.e-science.soton.ac.uk/
GeodiseDB_1_1_1/services/MetadataService</metadataWS>
 ...
</webservices>
```

Troubleshooting

(a) Error running `gd_dbsetup` in Linux using Matlab 6.5

If you see the following error running `gd_dbsetup`:

```
ccsshh::  Nuos ienngt rdyu mfbo rt etremrimnianla ls ettytpien g"s'.
M A T L A B   C o m m a n d   W i n d o w ' "
```

Which translates to:

```
csh: No entry for terminal type "'MATLAB Command Window'"
csh: using dumb terminal settings
```

then your login is C-Shell (csh or tcsh) and you didn't successfully setup

MATLAB_SHELL in .cshrc. Execute `gd_dbsetup` again and it will complete normally the second time.

(b) Why does my query cause an 'OutOfMemoryError'?
This error can happen when there are a large number (thousands) of results returned from a query. If the result message from the database service is too large then the Java client code (which sits behind the Matlab functions) runs out of memory when trying to parse it.

Here are two simple ways to solve this problem:

    i) Reduce the size of your query results (see the next section).

    ii) Decrease the value of <query_results><chunk_bytesize> in the <home_dir>/.geodise/ClientConfig.xml file. The database toolbox parses large query results a 'chunk' at a time, and the default chunk size is 2000000 (about 2MB). Decreasing this number means your machine deals with the query results in smaller chunks and is less likely to run out of memory.

(c) How can I reduce the size of my query results?
Reducing the query result size can speed up the query execution time. There are two general approaches which can be used together:

    i) Be more selective by making the query search conditions more specific. For example,

```
>> gd_query('standard.userID = me & standard.archiveDate
>= 2004-10-19 & params < 5')
```

    is more selective than

```
>> gd_query('standard.userID = me')
```

    and will reduce the size of the query results.

    ii) Reduce the number of fields returned in each result structure. The 3rd argument to gd_query can be set to a comma separated list of fields you want returned. The default, '*', will return all the fields available. By only requesting a subset of these you reduce the size of each result structure. The following example only returns the unique identifier(standard.ID) and params

field for each matching result.

```
>> q = 'standard.archiveDate >= 2004-10-19 &
standard.archiveDate <= 2004-10-21'
>> gd_query (q, 'file', 'standard.ID, params')
```

Combining approaches i and ii together will reduce the query result size significantly.

(d) 'Unknown error in XPath'
This is caused by an old version of Apache's xalan.jar that ships with Matlab 6.5. A newer version (2.5.2) is shipped with the Geodise Database Toolbox. Comment out (#) Matlab's reference to xalan.jar in classpath.txt to make sure the newer version (<GEODISELAB>/DatabaseToolbox/lib/xalan.jar) is being used.

(e) Security Exception when using Java 1.4.2 (versions 1.4.2_05 and above)
Java 1.4.2 (versions 1.4.2_05 and above) contains an old version of Xalan which causes an exception in the Apache XML Security software used by the Database Toolbox. To determine the version of Java used by Matlab type:

```
>> version -java
```

Copy xalan.jar from <GEODISELAB>/DatabaseToolbox/lib/ to <MATLAB_HOME>/sys/java/jre/<OS>/jre*/lib/endorsed/ to correct this. Create the endorsed directory in <MATLAB_HOME>/sys/java/jre/<OS>/jre*/lib/ if it does not exist.

 * Use the jre1.4.2 directory if it exists (e.g. for Matlab 7.0.1 on Linux), otherwise use the jre directory.

(f) Matlab 7.0.1 prints Log4J logging information (Linux and Windows).
The output of some functions are unnecessarily verbose when using Matlab 7 [SP1]. To correct this comment out (#) the following line in the classpath.txt file:

```
$matlabroot/java/jarext/axis.jar
```

A copy of axis.jar which does not cause this behaviour is supplied with the Database Toolbox.
(g) Query GUI displays a console error message in Matlab 7.0

A call to gd_query on Matlab 7.0 may produce the following error message in the Matlab console:

    ??? One or more output arguments not assigned during call to '(null)'.

The Query GUI will still function correctly and this error message, which is only occurs in the pcode version of gd_query, can be ignored.