

OpenGDA 8.26 walk through

Introduction

In this walk through, we will be installing GDA from scratch in a 32 bit Linux virtual machine and on a 32 bit Windows XP machine using files downloaded from the web.

If you wish to follow the walk through using the Linux virtual machine, you will need to [download](#) & install VMware player (for Windows this is currently `vmware-player-5.0.2-1031769.exe` ; 76 MB) and either download the OpenGDA CentOS 6.4 appliance ([CentOS-6.4-i386.zip](#) 807MB from <http://www.opengda.org/downloads/opengda/826walkthrough>) or create your own VM and install CentOS 6.4 yourself.

This virtual machine was created as a VMware Workstation 5.x VM, so should work with VMware Workstation 4 or later, VMware Server 1 or later and VMware Player 2 or later. When extracted from the archive, the virtual machine will be approximately 2.3GB in size, but by the end of this walk through the Virtual machine will have expanded to over 6GB. It should never expand to more than 20GB however.

The downloadable virtual machine has two users, `root` and `guest` both with `guest1` as the password.

If you wish to follow the walk through on a Windows machine, please bear in mind that we do not do this often so the process may not be quite as smooth as with a Linux installation.

- **Note:** Some of the steps in this walk through (such as installing VMware player on windows or the jdk on Linux), require `Administrator` or `root` privileges on Windows or Linux. If you do not have the required privileges on your machine, then unfortunately you may not be able to complete this walk through.

Virtual machine setup

The Linux virtual machine is an *Easy Install* of the 32bit CentOS 6.4 DVD, plus VMware tools. While most Diamond workstations are now 64 bit, I use the 32 bit version for this walk through because 64 bit virtual machines won't work on all 32bit host machines.

The only other modifications I made to the CentOS 6.4 installation were to configure it for UK timezone and keyboard, and to perform a `yum update` , to make sure that it was up to date:

```
su
yum update
exit
```

I then removed any unnecessary temporary files, zeroed free space and compacted the virtual disk, to minimise the download size.

As is usual with Linux, you should log in as `guest` and only `su` to `root` when needed.

Compiling and running GDA

OpenGDA Documentation

The GDA documentation page is a good place to start.

- Launch <http://www.opengda.org/documentation/> in your browser.
- Alternatively, google for `opengda` , click on the Documentation link, then click on the [Master documentation page](#) link.

The specific documents we will be using are:

- The **Reference Card**: http://www.opengda.org/documentation/manuals/Reference_Card/trunk/reference.html
- The **Infrastructure Guide**: http://www.opengda.org/documentation/manuals/Infrastructure_Guide/trunk/contents.html
- The **GDA QuickStart Guide**: http://www.opengda.org/documentation/manuals/GDA_QuickStart_Guide/trunk/contents.html

While the Quickstart Guide is very out of date, we can still use it to guide our steps using the newer Infrastructure Guide.

Install JDK

- See the Infrastructure Guide [section 1.1](#)

The first step is to download the current Java 6 development kit from <http://www.oracle.com/technetwork/java/javase/downloads/index.html> and install it.

For the 32 bit CentOS VM we currently need [jdk-6u43-linux-i586-rpm.bin](#) which using Firefox defaults to downloading to `~/Downloads` .

Following the [linked instructions](#), provided by Oracle, for an rpm capable Linux like CentOS, Fedora or Red Hat Enterprise Linux we need to:

```
cd ~/Downloads/  
chmod a+x ./jdk-6u43-linux-i586-rpm.bin  
su  
./jdk-6u43-linux-i586-rpm.bin  
exit
```

If you are using a 64bit Linux you will need to download the appropriate x64 rpm (currently [jdk-6u43-linux-x64-rpm.bin](#)), while if you are using a non rpm based Linux, you will need to follow the relevant instructions provided by Oracle.

On a Windows machine, we need to download and run the appropriate installer: [jdk-6u43-windows-i586.exe](#) for 32 bit or [jdk-6u43-windows-x64.exe](#) for 64 bit.

Install Eclipse

- From the Infrastructure Guide [section 1.2 \(for Eclipse Juno\)](#) or [section 1.3 \(for Eclipse Indigo\)](#)

For this walk through I will be using Eclipse Indigo, but Juno should be similar if you would prefer to use a more recent version of Eclipse.

Following the instructions at [section 1.3.1](#) for "Using a previously-configured Eclipse" I need to download one of the Eclipse Indigo 32 bit archives. From outside Diamond, we can access these files via <http://www.opengda.org/downloads/opengda/826walkthrough>.

On a 32bit Linux VM we need to download [eclipse372_32bit.tar.gz](#) 251 MB from <http://www.opengda.org/downloads/opengda/826walkthrough> and extract it to `/var/tmp` , for example:

```
cd /var/tmp  
tar -zxf ~/Downloads/eclipse372_32bit.tar.gz
```

Since typing `/var/tmp/eclipse372_32bit_20120830/eclipse` would be a hassle and CentOS defaults to having `~/bin` on the user path, for convenience we can create a bin directory in home and add a link:

```
mkdir ~/bin
ln -s /var/tmp/eclipse372_32bit_20120830/eclipse ~/bin/eclipse
```

On a 64bit Linux, we would benefit from downloading [eclipse372_64bit.tar.gz](http://www.opengda.org/downloads/opengda/826walkthrough) 249 MB from <http://www.opengda.org/downloads/opengda/826walkthrough> instead.

On a 32 bit Windows box, we need to download [eclipse372_32bit.zip](http://www.opengda.org/downloads/opengda/826walkthrough) 280 MB from <http://www.opengda.org/downloads/opengda/826walkthrough> and extract it to C:\OpenGDA using Extract All in windows explorer, and giving it C:\OpenGDA as the destination.

Again, for ease of access, we can create a shortcut on the desktop to the Eclipse executable: C:\OpenGDA\eclipse372_32bit\eclipse.exe

On 64bit Windows, we would want to download [eclipse372_64bit.zip](http://www.opengda.org/downloads/opengda/826walkthrough) 284 MB from <http://www.opengda.org/downloads/opengda/826walkthrough> instead.

As an alternative to using a previously configured Eclipse Indigo, you can follow the instructions in [section 1.2.2](#) or [section 1.3.2](#) of the Infrastructure Guide to build your own Indigo or Juno Eclipse installation. If you do so, you may also want to install the Spring Source Tool Suite, see [section 1.4.1](#) of the Infrastructure Guide.

Copy over the template workspace

- From the Infrastructure Guide [section 3.1.2](#)

The easiest way to get started is to use a template workspace from <http://www.opengda.org/buckminster/templates/>. This configures various eclipse settings for you, removing the need to do so yourself.

On Linux we can download the latest template (currently [template workspace v2.4.zip](#) 47 KB from <http://www.opengda.org/buckminster/templates/>) and extract it to `/var/tmp/gda_8.26/workspace`, for example:

```
mkdir -p /var/tmp/gda_8.26/workspace
cd /var/tmp/gda_8.26/workspace
unzip ~/Downloads/template_workspace_v2.4.zip
```

On Windows, we can download the same template (again [template workspace v2.4.zip](#) 47 KB from <http://www.opengda.org/buckminster/templates/>) and extract it to C:\OpenGDA\gda_8.26\workspace using Extract All in Windows Explorer, and typing C:\OpenGDA\gda_8.26\workspace as the destination.

As an alternative, you may want to configure your workspace manually. To do this, see the Infrastructure Guide [section 3.1.3](#)

Start Eclipse with the new workspace

To start Eclipse with the workspace we have just set up, on Linux we run:

```
eclipse -data /var/tmp/gda_8.26/workspace &
```

On Windows, we can just run `eclipse.exe` (from the desktop shortcut) and browse to `c:\OpenGDA\gda_8.26\workspace` (you can also do this on Linux too of course).

Eclipse fails to start

If starting Eclipse fails, it may be because it is trying to use a default java rather than the jdk we installed earlier. This is easily fixed by editing the eclipse initialisation file.

On Linux, edit the `eclipse.ini` file:

```
gedit /var/tmp/eclipse372_32bit_20120830/eclipse.ini
```

Since the jdk installed java to `/usr/java/default`, we need to add a `-vm` line and a `/usr/java/default/bin/java` line immediately before the `-vmargs` line:

```
...
-vm
/usr/java/default/bin/java
-vmargs
...
```

Similarly, on Windows, you can use `notepad` to edit the `eclipse.ini` file in `C:\OpenGDA\eclipse372_32bit`

Since the sdk installed java to `c:\Program Files\Java\jdk1.6.0_43`, we need to add a `-vm` line and a `c:\Program Files\Java\jdk1.6.0_43\bin\java` line immediately before the `-vmargs` line:

```
...
-vm
c:\Program Files\Java\jdk1.6.0_43\bin\java
-vmargs
...
```

Alternatively you could set the `JAVA_HOME` environment variable such that `$JAVA_HOME/bin/java` exists.

Update proxy settings

From v.2.4 the template workspace configures Eclipse to make HTTP and HTTPS connections via the Rutherford lab webcache and bypass the proxy for all `diamond.ac.uk` domains. Outside of Diamond these settings will cause problems, so they will need to be removed or adjusted for your local proxy. To do this:

- Open the **Preferences** dialog (on the **Window** menu)
 - Navigate to **General, Network Connections** and then either

- Select **Direct** as your **Active Provider** if you don't need any proxy settings for your internet connection
- or, if you do need proxy settings, leave **Direct** selected and as your **Active Provider**, remove the `*.diamond.ac.uk` proxy bypass and edit the HTTP & HTTPS schemas to provide the correct proxy settings

Materialize OpenGDA

- From the [Reference Card](#), under **Checking Out, Using the IDE, Materialising in Eclipse**

Since we have already set up the workspace and started eclipse, we can skip to:

- **File > Open a Component Query...**and enter the full CQuery URL
 - For GDA 8.26 the CQuery is <http://www.opengda.org/buckminster/base/gda-v8.26.cquery> (normally a copy and paste from the [reference card webpage](#) is the easiest way to do this)

This should open the CQuery, if not then you may need to check your http and https proxy settings, see above.

Before we can *materialize* the workspace, there are a few parameters we need to change to select the correct configuration and access the public github repositories rather than the internal diamond git repositories.

- On the **Main** tab:
 - In **Component Name**, replace `$(component)` with `example-config`
 - In the future, this should work with `example`, but for the moment we need to specify we mean the config.
- On the **Properties** tab:
 - For the key `download.location` replace `diamond` with `public`
 - For the key `skip_uk.ac.gda..test` replace `false` with `true`

Now we can actually start the materialization:

- Click on **Resolve and Materialize**

Note, there may be a few non fatal errors, but these can be safely ignored.

An important note about the workspace location

Once the materialize has completed, the Eclipse workspace contains **absolute paths**, so you **must not rename**, **move or copy** the **workspace**, it's corresponding `workspace_git` directory (created during the materialization process) or any directory above it. If you do then those workspaces will not work and will corrupt any workspace you materialize with the same path.

If you need to your workspace to be elsewhere, extract the template workspace at the new location, perform a fresh materialize from within that workspace and then clean up your old workspace (commit or stash any changes and then pull those changes or unstash them into the new workspace).

- **Warning: Do not rename, move or copy a GDA eclipse workspace after materialization.**

Build GDA

In the **Navigator** pane, find `uk.ac.gda.example.site`, open `buckminster.properties`, comment out the all platforms entries (`target ... =*`), un-comment the appropriate target for your installation and save.

On 32 bit Linux `buckminster.properties` will look like this:

```
#target.os=*
#target.ws=*
#target.arch=*

target.os=linux
target.ws=gtk
target.arch=x86
...
```

While on 32 bit Windows box, it will look like this:

```
#target.os=*
#target.ws=*
#target.arch=*
...
target.os=win32
target.ws=win32
target.arch=x86
```

Then we can attempt to build the product using buckminster:

- Then, right click on `uk.ac.gda.example.site` and select **Buckminster > Invoke Action...**
- Select the following **Properties File** from the **Workspace**: `uk.ac.gda.example.site/buckminster.properties`
- Select **create.product** and **Ok**

This may error, in which case, you may need to restart your Eclipse & try again:

- **File > Restart**
- Right click on `uk.ac.gda.example.site` and select **Buckminster > Invoke Action...**
- Select **create.product** and **Ok**

Often this won't work unless you do a clean build of the project first:

- **Project > Clean...**
 - Select **Clean all projects, Build the entire workspace** and **Ok**

If you get a "Jython not configured" dialog:

- Select **Manual config**
 - **New...**
 - Browse the following directory and select `jython.jar` from
 - `/var/tmp/gda_8.26/workspace_git/gda-libs.git/uk.ac.gda.libs/jython2.5.1` on Linux
 - `C:\OpenGDA\gda_8.26\workspace_git\gda-libs.git\uk.ac.gda.libs\jython2.5.1` on Windows
 - Set **Interpreter Name:** to `Jython2.5.1`

If you get a "Python not configured" dialog, just click **Don't ask again** as Python is not required to build the product.

Sometimes, a clean build isn't enough and you have to tell eclipse to build again, one or more times, until all of the dependencies are built. You can do this by running:

- **Project > Build All**

Once there are no errors in the *Problems* view, you know that you no longer need to keep rebuilding, and the buckminster product creation should now complete:

- Right click `uk.ac.gda.example.site` and select **Buckminster > Invoke Action...**
- Select `create.product` and **Ok**

Install Python (Windows only)

Most Linux distributions contain a version of python suitable for the GDA start up scripts to use, but windows machines rarely do.

You can install Python 2.7.3 on a windows machine by running the appropriate msi installer, [python-2.7.3.msi](#) for 32 bit or [python-2.7.3.amd64.msi](#) for 64 bit, from <http://www.python.org/getit/>.

This walk through assumes that you install this to the default `C:\Python27` directory.

Launch the GDA Servers

GDA consists of a series of *servers* and a *client*. On Diamond beamlines the *servers* are run on a beamline control machine, while a *client* is run on one or more Beamline workstations. For development, both the *servers* and *client* can run on a single machine.

To launch, or relaunch, the servers:

- On Linux select **Run > External Tools > example gda servers dummy restart**
- On Windows select **Run > External Tools > example gda servers dummy restart (win)**

If these options don't appear in the External Tools list, then select **Run > External Tools > External Tools Configurations...** expand the **Program** collection in the browsing pane on the left, select the *launcher* mentioned above and then click on the **Run** button. After this has been done one, this launcher will appear in the External Tools list,

JAVA_HOME problems

If starting the servers fails with an error about `JAVA_HOME` being incorrect or not being set, then we may need to adjust the gda servers script that the launcher relies on.

On Linux, edit the `example-config/bin/gda_servers_from_ide` script and add the line the following line to the top of the file with the location of the jdk we installed earlier:

```
export JAVA_HOME="/usr/java/default"
```

On Windows, we need to edit the `example-config/bin/gda_servers_from_ide.bat` script to modify the `JAVA_HOME` line, such that the java executable can be found at `JAVA_HOME\bin\java.exe`

For example:

```
set JAVA_HOME=C:\Program Files\Java\jdk1.6.0_43
```

Alternatively, if `JAVA_HOME` is set in the environment, we may just need to comment out the `set JAVA_HOME` li

ne or remove it completely.

Python problems

If launching the servers fails with an error about being unable to find python, then we need to point the start up scripts at the correct version of python.

On Linux, edit the `example-config/bin/gda_servers_from_ide` script and replace `python2.6` with `python` or whatever python2.x is called on your machine (since this may not work with python 3.x).

On Windows, edit the `example-config/bin/gda_servers_from_ide.bat` script to change `c:\python27\python.exe` to point to wherever your `python.exe` is located.

Windows firewall alert

Since the GDA servers open up various TCP/IP ports for communication, the Windows firewall may pop up an alert asking in you want to block java. If this happens select **Unblock**.

Launch the GDA Client

When the console window shows the `Server initialisation complete` message, then we can start the client:

- Go to `uk.ac.gda.example.feature`
- Right click on `uk.ac.gda.example.product.launch` and click on **Run As > uk.ac.gda.example.product**

If all goes well, the GDA client should now be running and communicating with the GDA servers.

Validation error

If launching the product fails with a **Validation** error, as it will when using a launcher configured for 64 bit on a 32 bit machine, then we need to

- **Cancel** the **Validation** dialog (selecting Ok here would continue trying to launch the client but eventually fail)
- Right click on `uk.ac.gda.example.product.launch` and click on **Run As > Run Configurations...**
 - Go to the **Plug-ins** tab.
 - Deselect the **Include optional dependencies** option.
 - Click on the **Deselect All** button.
 - Click on the **Add Required Plug-ins** button.
 - Finally click on **Apply** and then **Run**.

Windows firewall alert

As the GDA client also opens up various TCP/IP ports for communication, the Windows firewall may pop up an alert asking in you want to block java. Again, if this happens select **Unblock**.

Windows peculiarities

We don't use GDA with Windows at Diamond, so rarely try deploying to Windows. While most things *should just work*TM there are things which creep in which we might not notice.

Several such issues present themselves with our standard example configuration. Since we have not had much time to use the Windows installation, there may be other things which fail to work We would be happy to hear about them, or receive patches or pull requests to fix any that you find. See the collaboration contact information

at the end of this walk through.

Server launchers do not always work as expected

Running **example gda servers dummy restart (win)** works fine to start the servers the first time, but fails to shut down the old servers if they are already running, so when they attempt to restart they fail.

Similarly, running **example gda servers dummy stop (win)** seems to work if you run it before running **example gda servers dummy restart (win)** for a second time, but not if you try to run it after. In this case you have to manually kill the GDA server processes by using the **End Process** button on the four `java.exe` processes listed in the Windows Task Manager.

Jython Script project names

- This problem manifests as **Project Explorer** pane being blank.

The Jython Script project names all contain colons, which appear to cause a problem with the **Project Explorer** pane in the Windows GDA client (probably because it is not a valid character in a Windows file path).

To work around this, edit `example-config/xml/server/jythonServerFacade.xml` to replace all occurrences of " `script:` " with " `Script -` ".

For example, change:

```
...
    <constructor-arg index="1" value="Scripts: User" />
...
    <constructor-arg index="1" value="Scripts: Config" />
...
    <constructor-arg index="1" value="Scripts: Core" />
...
```

to :

```
...
    <constructor-arg index="1" value="Scripts - User" />
...
    <constructor-arg index="1" value="Scripts - Config" />
...
    <constructor-arg index="1" value="Scripts - Core" />
...
```

After a restart of the client and servers, the **Scripts User** entry should now be shown in **Project Explorer**.

Nexus libraries

- This manifests as a NeXus error when trying to run a `scan` command.

Our NeXus libraries do not appear to be working on windows at the moment.

To work around this, switch to using the old ASCII `.dat` file. Edit `example-config/properties/java.properties` and replace `NexusDataWriter` with `SrsDataFile` :

```
gda.data.scan.datawriter.dataFormat = SrsDataFile
```

You can also do this temporarily from the Jython prompt by executing the following commands:

```
from gda.configuration.properties import LocalProperties
LocalProperties.set("gda.data.scan.datawriter.dataFormat", "SrsDataFile")
```

Scan Data Plot 1 view

- This manifests as the default **Scan Data Plot 1** view showing nothing but grid lines.

The work around seems to be to show the newer **Scan Plot** view:

- **Windows > Show View > Other...**
 - Select **Data Acquisition - General > Scan Plot** and **Ok**

The next **scan** that you do will go to both the new **Scan Plot 1** view and the old **Scan Data Plot 1** view, but **Scan Data Plot 1** will be brought to the front and data from old scans re-plotted.

Using OpenGDA

Follow the QuickStart Guide

You should now be able to play with the "Quick Tour" of GDA starting at [section 1.5](#) of the GDA QuickStart Guide.

As this guide is rather out of date, some things will look different to the screenshots, while some changes to default settings will have changed the behaviour.

QuickStart section 1.5.1

In this section you will need to open the **XY Scan Plot** view before you will see the graph of the **scan** (also see the *Windows peculiarities* section above):

- Option 1: Select the **XY Scan Plot** button near the top of the GDA client window.
- Option 2: **Windows > Show View > Other...**
 - Select **Data Acquisition - General > XYScanPlot** and **Ok**

QuickStart section 1.5.4

Since this section assumes that the default data writer is Srs, if you are using Linux this section will not work unless you pre-emptively switch to using the Srs data writer:

```
>>> LocalProperties.set("gda.data.scan.datawriter.dataFormat", "SrsDataFile")
```

You may also find that when you add a plot view, it has a name other than **Plot 1**. If that is the case then commands used should be adjusted to specify the actual plot name rather than **Plot 1**. For example:

```
>>> RCPPlotter.plot('Plot 3', data[0], data[1])
```

QuickStart section 1.5.5

On Windows, you will not be able to complete this section due to the issue with the Nexus libraries (see the *Windows peculiarities* section above).

QuickStart section 1.5.6

On Linux, you may not be able to complete this section because the Data Project may not appear in the Project Explorer.

If this happens then you can load the file manually:

- **File > Open File...**
 - Navigate to the `.nxs` file using the filesystem browser

The name of the data file is shown when a scan starts. For example:

```
>>> scan sgw .2 2. .2 scannableGaussian0 -1. 1. .02
Writing data to file (NeXus):
/var/tmp/gda_8.26/workspace_git/gda-core.git/example-config/users/data/2013/3.nxs
width      x          y
0.2000     -1.0000    -0.010782
...
2.0000      1.0000     0.24555
Scan complete.
```

Unfortunately, the data may still not show on the graphs.

QuickStart section 1.5.7 & 1.5.8

These sections should be skipped, as we do not include the *Object Viewer* or *Control System Studio* in the example configuration.

Collaboration

We are keen to encourage people to try OpenGDA, find, report and fix bugs, develop new features and contribute all of these things back.

You can follow the various projects on github:

- Go to <https://github.com/openGDA> and <https://github.com/DiamondLightSource>

We are happy to help wherever we can, and recommend that people join the OpenGDA Developers mailing list, or the low traffic Announcement list:

- For the OpenGDA Developers mailing list, go to <https://lists.opengda.org/mailman/listinfo/developers>
- For the OpenGDA Announcements mailing list, go to <https://lists.opengda.org/mailman/listinfo/announce>

The Developers list was created as a place for external collaborators to communicate both with the Diamond GDA team and other collaborators around the world.

We are also happy to take direct questions to the Diamond GDA team:

- E-mail <mailto:dag-group@diamond.ac.uk>

Thank you for taking the time to try OpenGDA.
