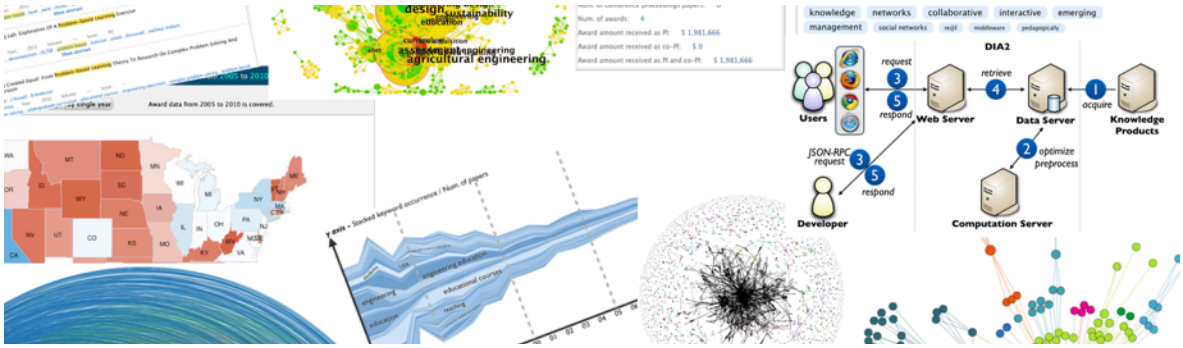


Documentations for DIA2
Purdue University
Virginia Tech
Arizona State University
Stanford University

Getting Started with DIA2 Development

Prepared by: Xin Chen, Hanjun Xian, Umer Arshad, Krishna Madhavan

Delivered: February 3, 2014

Hardware Infrastructure

Hardware Infrastructure

At Purdue: 9 Mac Mini Servers

At NSF: 4 Mac Mini Servers

Table 1. Server usage and deployment at Purdue and NSF

Function	Development	Production New	Production	NSF
Web Server	1. Apache, HTML, JavaScript, PHP	4	7	NSF1
Data Server	2. MySQL	5	8	NSF2
Computation Server	9. Java, SOLR, PHP, Python, Git	6	9	NSF3

Figure 1. DIA2 Servers

At Purdue, nine servers are divided into three groups, each of which contains a web server, a data server, and a computation server. Machine 4 and 5 are new servers with solid state drives. They have different mountain lion OS, while other servers have Snow Leopard OS. However, machine 3 is dead. We may order 2 new machines and one of them will be used to host the GIT version control repository. Some servers have not restarted over two years, so we need to setup a cron job to restart the machines on a regular base, and keep a log of when the machine restart.

Inside NSF, machine 1,2,and 3 serve as web server, data server, and computation server respectively. Some of the servers inside NSF often die, but when restart, they function well. The reason is unclear now. When they die,

they responses to ping but cannot use SSH to connect. We may need to setup a Cron job for the machines to restart on a regular basis. Right now, some services do not automatically restart when the machine restart such as SOLR. So we need to put them in the start-up items folder. Inside NSF, machine 4 is a spare machine.

Software Infrastructure

Software Infrastructure

Table 2. Software Versions on Each Machine

Software	1	2	3 (Dead)	4	5	6	7	8	9
OS Version	Mac OS X 10.6.8	10.6.4		10.8.4	10.8.4	10.6.8	10.6.8	10.6.8	10.6.8
Build Version	10K540	10F2025		12E55	12E55	10K549	10K540	10K540	10K549
Apache	2.2.17	N/A		2.2.22	N/A	2.2.22	2.2.17	N/A	N/A
PHP (Apache Web)	5.3.4	N/A		5.3.15	N/A	5.3.15	5.3.4	N/A	N/A
PHP (cli)	5.3.4	5.3.2		5.3.15	5.3.15	5.3.15	5.3.4	5.4.3	5.3.15
MySQL	N/A	5.0.88		N/A	5.6.12	N/A	N/A	5.0.92	N/A
Java(JVM)	1.6.0 HotSpot (TM)	1.6.0 HotSpot (TM)		N/A	N/A	1.6.0 HotSpot(TM)	1.6.0 HotSpot (TM)	1.6.0 HotSpot (TM)	1.6.0 HotSpot (TM)
Javac(JDK)	1.6.0	1.6.0		N/A	N/A	1.6.0	1.6.0	1.6.0	1.6.0
Python (Default)	2.7.4	2.6.1		2.7.2	2.7.2	2.6.1	2.6.1	2.6.1	2.6.1

Commands to check software version:

OS and build version: $ sw_vers

System profile and time since boot: $ system_profiler SPSoftwareDataType

PHP web and Apache: create a php file in the default path, call the function phpinfo(), then open the php file in the browser.

PHP (cli)-command line: $ php -version

MySQL: $ mysql

Java: $ java -version

Javac: $ javac -version

Python: $ python

Developer Accounts

Currently, only machine 1 has different developer accounts, because this is the major development server. Other machines only have one user: ci4ene. There are three user groups: uidesign(UI design),uidev (UI development), and datadev(data development). On machine1, there are two files: Users/ci4ene/createUser.sh and Users/ci4ene/createGroup.sh. These two files manage the developer accounts. Use the command "$ id accountName" to check the groups an account belong to.

Person Name	Account Name	Groups on 01
admin	ci4ene	staff
Hanjun Xian	hxian	uidesign/uidev/datadev
Xin Chen	xchen	uidev
Muhammad Umer Arshad	ssingh	datadev
Yuetling Wong	ywong	uidev
Zhihua (Emma) Dong	zdong	uidesign
Rachel Whitson	rwhitson	uidesign
Qing Liu	qliu	uidev
Anali Sakhala	asakhala	uidev

Table 3. Developer Accounts on Each Machine

Setup A New Server

You can set up a new server following these steps as long as you have an IP address available to use.

Setup A New Web Server

1. Plug-in power to the machine. Plug the monitor to the machine. Plug the keyboard USB to the machine rather than the monitor.
2. Turn on the machine and follow the instruction. Use whatever AppleID to register.
3. Set up server. Config the IPv4, choose “manually” from the drop-down list, manually type in the IP address.
4. In the server pane, turn unnecessary stuff such as calendar and messages off. For the web server, only turn “websites” on.
5. Update whatever available softwares from software update.
6. Change the security preferences to accept software installers from anywhere, otherwise, we won’t be able to install softwares such as git, and graphviz.
7. In system preferences->sharing, enable “remote login”. This is to enable SSH. This is for both the web server and data server.
8. For the web server, write a file named phpinfo.php and put it in the default web path. The default web path is different on the server with OS X 10.8. The default web path is /Library/Server/Web/Data/Sites/Default. On the server with OS X 10.6 and 10.7, the default web path is /Library/WebServer/Documents. Call the function phpinfo() in that file. Need to use “sudo” to write the file. Open the web browser to see the php version. After the testing, change the file name to phpinfo.ppp or remove it for security reason.
9. For the web server, change the PHP memory limit to be larger. In php.ini (/private/etc/php.ini), change memory_limit = 2G. The default setting is 128M. By default, DIA2 machine use PHP memory_limit as 2G.
10. For the web server, write a file named db.php, similar as DIA2 dbconn.php to test the connection from this server to any data server. Also, after the test, change the file name to db.ppp or remove it for security reason.
11. For the web server, download and install git and graphviz. Run “git” in terminal to test git. Run “neato” in terminal to test graphviz. If the response is “command not found”, it means graphviz is not successfully installed, need to install again. If there is no response, it means graphviz is installed.
12. Restart the machine (Remote restart: $ sudo shutdown -r now) and then use another computer to test the SSH connection. Put a post-it on the machine and in the box stating the host name and IP address.
13. Ask the machine room personnel to rack the machine in the machine room, and test the SSH connection again.

Setup A New Data Server

1. Plug-in power to the machine. Plug the monitor to the machine. Plug the keyboard USB to the machine rather than the monitor.
2. Turn on the machine and follow the instruction. Use whatever AppleID to register.

3. Set up server. Config the IPv4, choose “manually” from the drop-down list, manually type in the IP address stated above.
4. In the server pane, turn unnecessary stuff such as calendar and messages off. For the data server, turn everything off.
5. Update whatever available softwares from software update.
6. Change the security preferences to accept software installers from anywhere, otherwise, we won’t be able to install softwares such as git, and graphviz.
7. In system preferences->sharing, enable “remote login”. This is to enable SSH. This is for both the web server and data server.
8. For the data server, MySQL is missing on OS X 10.8, download from MySQL site http://dev.mysql.com/downloads/mysql/ . Use the Mac OS X ver. 10.7 (x86, 64-bit), DMG Archive. This works fine for 10.8.

   Following the MySQL part in the following link to install MySQL.

   http://coolestguyplanettech.com/downtown/install-and-configure-apache-mysql-php-and-phpmyadmin-osx-108-mountain-lion

   A back up of the .dmg file and the instruction is on machine05 /Users/ci4ene/Downloads.

   Install all 3 in the installation package. MySQLPrefPane is for us to see whether MySQL is turned on in the preference pane.

   - mysql5.6.xxx.pkg
   - MySQLstartupitem.pkg
   - MySQLPrefPane

   After installation, in order to use mysql commands without typing the full path to the commands. We need to add the mysql directory to the shell path. “.bash_profile” file in the home directory, using vi or nano, write the following line in the .bash_profile: export PATH=”/usr/local/mysql/bin/: $PATH”

   Here are the commands to remotely install .dmg files through command lines:

```
$ sudo hdiutil attach <image>.dmg
$ sudo installer -pkg /Volumes/<image>/<image>.pkg -target /
$ sudo hdiutil detach /Volumes/<image>
```

9. For the data server, change the max_allowed_packet of MySQL to be larger in my.ini or my.cnf as follows. By default, DIA2 servers use max_allowed_packed as 1024M.

```
[mysqld]
max_allowed_packed=1024M
```

   The path of my.cnf file is /private/etc/my.cnf.

   For MySQL 5.6.X, my.cnf file is in /usr/local/mysql, use the following command to copy it to /etc/my.cnf.

```
$ sudo cp /usr/local/mysql/my.cnf  /etc/my.cnf
```

   After making the changes, restart MySQL

```
$ sudo /usr/local/mysql/support-files/mysql.server restart
```

10. Enter MySQL monitor, use the following syntax to create a new database.

```
create database DBNAME;

create user USERNAME identified by 'PASSWORD';

grant all privileges on DBNAME.* to USERNAME@'%' identified by 'PASSWORD';

grant all privileges on DBNAME.* to USERNAME@'localhost' with grant option;
```

11.Restart the machine (Remote restart: $ sudo shutdown -r now) and then use another computer to test the SSH connection.

12.Put a post-it on the machine and in the box stating the host name and IP address.

13.Ask the machine room personnel to rack the machine in the machine room, and test the SSH connection again.

Setup A New Computation Server

A computation server needs to have Apache Solr installed. Solr relies on Java. Many other computations on the computation server also rely on Java. The servers with OS 10.6 and 10.7 comes with Java, but 10.8 does not have Java installed, so if we buy new servers in the future, we need to install Java.

Install and Run Solr:

1.Download solr from one mirror site provided in this link http://www.apache.org/dyn/closer.cgi/lucene/solr/4.4.0 to the directory /Users/ci4ene/solr-4.4.0.tgz. The following are instructions to download the file via command lines.

```
$ cd /Users/ci4ene
$ curl -O http://www.eng.lsu.edu/mirrors/apache/lucene/solr/4.4.0/solr-4.4.0.tgz
```

2. Untar the .tgz file, so we will have a folder named solr-4.4.0.

```
$ tar xzf Downloads/solr-4.4.0.tgz
```

3. The instructions 4, 5, and 6 are in the README.txt file under the /solr-4.4.0/apache-solr-4.4.0/example directory.

```
$ cd /solr-4.4.0/apache-solr-4.4.0/example
```

4. To run solr execute the command.

```
$ nohup java -jar start.jar &
```

5. Open the web browser and type the url http://ci4eneX.ecn.purdue.edu:8983/solr/ to confirm solr running.

Index Files

1. Use the following commands to index .xml file. The DIA2 data is usually in files named nsf.xml or nsfpub.xml.

```
$ cd /solr-4.4.0/example/exampledocs
$ ./post.sh nsfpub.xml
```
(this one adds to the old index)

or

```
$ cd /solr-4.4.0/example/exampledocs
$ java -jar post.jar nsfpub.xml
```
(this one replaces the old index)

2. Browse the indexed files, go to http://ci4eneX.ecn.purdue.edu:8983/solr/browse

Update Solr Indexes:

1. The following is the typical format of a .xml file for Solr index.

```
<add>
  <doc>
    <field name =”id” ></field>
    <field name = “”></field>
  </doc>
  <doc>
    <field name =”id” ></field>
    <field name = “”></field>
  </doc>
</add>
```

2. To add new data to the index, here are the commands:

```
$ cd /solr-4.4.0/example/exampledocs
$ curl http://ci4eneX.ecn.purdue.edu:8983/solr/update?commit=true -H 'Content-type: text/xml' --data-binary filename.xml
```

This command is actually what is implemented in example/exampledocs/post.sh mentioned in instruction 6 under “Install and Run Solr”. So the following command can also be used to add new data into the index.

```
$ cd /solr-4.4.0/example/exampledocs
$ ./post.sh filename.xml
```

“id” is the unique key for each document, if one added document has the same id with a previously indexed document, Solr will not index it again, even if the other fields of the document have been changed.

3. To delete previously indexed documents, here are the commands:

```
$ curl http://ci4eneX.ecn.purdue.edu:8983/solr/update?commit=true -H 'Content-type: text/xml' --data-binary <delete><id>someID</id></delete>
```

Or

```
$ curl http://ci4eneX.ecn.purdue.edu:8983/solr/update?commit=true -H 'Content-type: text/xml' --data-binary <delete><query>someQuery</query></delete>
```

Editing Schema.xml:

1. The default schema of xml files for Solr index is in solr-4.4.0/example/solr/collection1/conf/schema.xml. This files defines the types of all xml fields, as well as the uniqueKey field (the default uniqueKey is id). If the xml file to be indexed has different field names, the field names need to be added into the schema.xml. Just add the new field names into the schema.xml file with in the <fields></fields> tag, and above the <dynamicField/>. The following is an example of adding a new field abstract.

```
<field name=”abstract” type=”text_general” indexed=”true” stored=”true”>
```

When the schema.xml is modified, Solr needs to be restarted for the new schema to be effective.

2. When the schema is modified, the Solr GUI will not show the new fields on http://ci4eneX.ecn.purdue.edu:8983/solr/browse, the product_doc.vm file under solr-4.4.0/example/solr/collection1/conf/velocity needs to be modified for the GUI to show the new fields. If the Solr GUI is not working, the following method can be used to query Solr indexed data.

http://ci4eneX.ecn.purdue.edu:8983/solr/select?q=fieldName:query

Run Multiple Indexes:

There are two major ways to run multiple indexes with Solr.

1. Run multiple Solr instances with separate ports. Port information is in example/exampledocs/port.sh and example/etc/jetty.xml. This can be advantageous if you want to manage resources for each Solr instance completely independently.

2. Use the Solr multicore features. It is possible to segment Solr into multiple cores, each with its own configuration and indices. Multiple cores are administered through a common administration interface. Multiple cores are using the same one port and one Java process. The core admin page for Solr is http://ci4eneX.ecn.purdue.edu:8983/solr/#/~cores/collection1. "collection1" is the default core if there is only one core running. New cores can be added by clicking "Add Core" on this core admin page, and specify the core name, instance directory, data directory, solrconfig and schema files paths. The default instanceDir is solr-4.4.0/example/solr/collection1. The default dataDir is solr-4.4.0/example/solr/collection1/data. These data are used by the first core collection1, so if a new core is created, it is better to specify a new data folder such as data2. The default solrconfig.xml and schema.xml files are in solr-4.4.0/example/solr/collection1/conf. Different solrconfig and schema files can be created for each core.

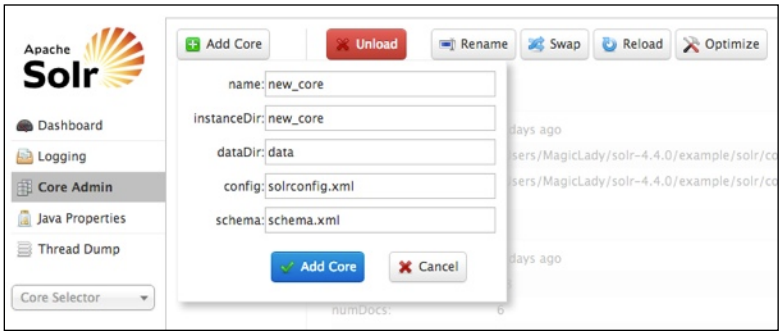


Figure 2. Multiple Core Admin Page

3. If there are multiple cores running, the only difference on browsing and updating the indexes is to add the core name in the URL. For example, if there is only one core, the browse URL is http://ci4eneX.ecn.purdue.edu:8983/solr/browse, and update URL is http://ci4eneX.ecn.purdue.edu:8983/solr/update. If there are two cores named core0 and core1. Then the URLs for core0 are http://ci4eneX.ecn.purdue.edu:8983/solr/core0/browse and http://ci4eneX.ecn.purdue.edu:8983/solr/core0/update. Then the URLs for core1 are http://ci4eneX.ecn.purdue.edu:8983/solr/core1/browse and http://ci4eneX.ecn.purdue.edu:8983/solr/core1/update.

Solr Server and Client:

Right now, machine 01/02 is using the Solr on machine 09 through port 8983, and machine 07/08 is using the Solr on machine 09 through port 9009 too. Machine 04/05 is using the Solr on machine 06 through port 8983.

There are three solr folders on 09: solr3.6.2, solr3.6.2_II, and solr4.1. We are currently using solr3.6.2 with port 8983. But they other solr servers are also running, so there are 3 start.jar process running. Need to kill them all, and restart the one in use.
Check current Solr server:

```
ps -e | grep start.jar
```

PHP Solr client is stalled on 01, 04, and 07. On these three webservers, under any DIA2 repository, the Solr client is in /DIA2/site/JSONRPC/SolrPhpClient. The file that calls this client is /DIA2/site/JSONRPC/getDocumentIDsByConstraints.php in the function getDocumentIDsByFullText. Under /DIA2/site/JSONRPC/, the file generateFullXMLSOLR.php is to convert database data into .xml for Solr index. Inside NSF, there is some codes in this file to process special characters.

Solr Tutorials and Wiki-page:

1. Apache Solr 4.4 Reference Guide: http://apache.tradebit.com/pub/lucene/solr/ref-guide/apache-solr-ref-guide-4.4.pdf

2. Solr tutorial: http://lucene.apache.org/solr/4_4_0/tutorial.html

3. Solr Wiki: http://wiki.apache.org/solr/FrontPage

How to Use Git

Git Version Control

Git is currently hosted on machine 09. Not all DIA2 codes are under version control. Only the web and UI codes on machine 1,4,and 7 are using version control. Machine 1 is the development server, so it is bi-directional. It can both push and pull codes from the repository, but machine 4 and 7 should only pull codes. The future plan is to have all codes especially the ones on the computation servers under version control.

Basic Operations

Here are some basic operations when working with git on dev machine1:

1. Use SSH to connect to ci4ene01.

2. Create your own copy of DIA2:

```
$ cd /Library/WebServer/Documents/
$ mkdir xxx_DIA2
```

XXX is the name you assigned to your folder

```
$ cd xxx_DIA2
$ git clone git@ci4ene09.ecn.purdue.edu:DIA2.git      (password is ***)
```

The above steps will create a folder of your own, within which you download the DIA2 web copy.

3. Git supports branching which means that you can work on different versions of your collection of files. A branch separate these different versions and allows the user to switch between these version to work on them. Some typical branches are masters, dev, alpha, and beta. DIA2 use the dev branch for development. Change the default branch to dev, using the following commands:

```
$ cd DIA2
$ git checkout dev
$ git config push.default simple
```

4. Edit DIA2 codes
   ..... (Edit files like you usually do)
   If you add new files/folders, run this in xxx_DIA2/DIA2/

```
$ git add .
```

5. Debugging your codes
   In your browser, see the UI by http://ci4ene01.ecn.purdue.edu/xxx_DIA2/DIA2/pages/

6. Update the branch
   So far, you run and test your codes in your own folder. When you are ready to roll out changes to the git server:

```
$ git commit -am 'Some message here indicating what changes have been made since the last version'
$ git push    (password is ***)
```

Comprehensive GIT Tutorial

Here is a comprehensive Git tutorial: http://www.vogella.com/articles/Git/article.html

New DIA2 Developers

Two things to do

1. Are you a data developer, a UI developer, or a UI designer? Request an account on the DIA2 server from the DIA2 team according to your role.

2. Getting familiar with basic operations of Git from the “How to Use Git” chapter in this document.

Data Acquisition

Data Source

1. Engineering Village http://www.engineeringvillage.com

The ev data we downloaded before are in the format of .ev. When downloading new data from ev, we can choose to use RIS format.

2. RIS (what are the datasources of those data in the RIS folder?)

3. NSF (http://nsf.gov/awardsearch/download.jsp)

NSF data are organized by fiscal year (Oct1 - Sep 30).

Data source might change from time to time, because some databases may decide to add new publications or stop index certain publications.

Data Acquisition Code

1. Before running the data acquisition code, create a new database with the DIA2 database schema. The DIA2 database schema is at: /Users/ci4ene/Documents/DBbackup/createDIA2DB.sql. Copy this file to a data server as needed. Enter MySQL monitor,

```
> create database [DBname];
```

exit MySQL monitor, run the following command:

```
$ mysql -u [username] -p[password] [DBname] < createDIA2DB.sql
```

2. On machine 09, under /Users/ci4ene/workspace_02252013/DIA2/importData. There is a backup of the entire workspace folder on machine 06 at /Users/ci4ene/workspace_09202013/

Under the importData folder, there is a file named dbconn.php. Make sure to add the database created above into this file.

Under folder ev, there are folder repository, files EVParser.php, import.php and run.sh.

Under folder RIS, there are folder repository, files RISParser.php, import.php and file run.sh.

Under folder nsf-pub, there are folder repository, file NewNSFPubParser.class.php, newImport.php and file newRun.sh.

Data are in the respective repository folders. Specify which database to connect in import.php or newImport.php (in the case of NSF data).

To insert the data in repository into the database, use

```
$ sh run.sh
```

in the case of the NSF data, use

```
$ sh newRun.sh
```

3. Documentation of detailed functions in each class in php files is on ci4ene09, under /Users/ci4ene/documentations/dataAcquisition. Use browser to open the masterTOC.html in order to view the documentation.

Validation

1. The commands above insert all data in the repository folder into the database. To make sure data are inserted correctly, a better practice is to insert a small portion into the database first and validate whether they are correct before bulk insertion.

2. Use the following command to insert one data file into the database:

```
$ php import.php repository/filename
```

in the case of the NSF data, use

```
$ php newImport.php repository/filename
```

use the following command to count how many entries are in the data file

```
$ grep “entrybreaker” [filename] | wc -l
```

“entrybreaker” varies for different file formats. For example, for RIS, it is “ER -”, for ev, it is “<RECORD”.

After obtained this number, compare with the entries inserted into the database to see whether they are the same.

Network Program

Output Graph Size

SVGFileExport feSVG = new SVGFileExport(network);

feSVG.export(outFile + ".svg", 2000, 2000);

“2000, 2000” is the final output file size. Title takes the upper 10%, and legend takes the bottom 5%. These are defined in SVGFileExport.java, the function exportSVG. If need to make the graph only square, needs to reverse calculate and adjust.

Canvas Size

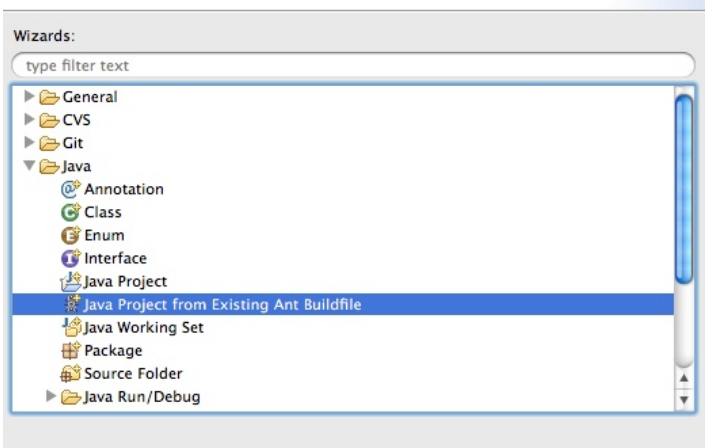
The canvas size is set to 100000 X 100000 in the network class. The output graph size is different from canvas size. When there are too many iterations for Fruchterman-Reingold, the nodes are being pushed to the edge of the canvas, so the graph looks square rather than round eventually. Gephi does not have this problem, because it’s canvas is unlimited.

Debug Using Eclipse (Java) on Local Machine

Eclipse supports Ant Buildfile.

File-->New-->Other-->Java Project from Existing Ant Buildfile

Open the starting XML file. The entry point of the entire program is usually in the default package, or the nanoHUB package.



Animation Generator

In the nanoHUBcode, there is a file nhGenAnimation.sh.

This shell scripts connect multiple graphs into a .gif animation.

Athena DB and Topic Map

Athena DB Translator

On ci4ene06, /Users/ci4ene/bak/DIA2-06032013/athenaDBTranslator/AthenaParser.class.php

Topic Mapper with Thesaurus with Solr

/Library/WebServer/Documents/Xin_DIA2/DIA2/site/JSONRPC/generateConceptDocMap.php

Inside NSF, this may have been moved to 103. This relies on Solr client.

Internal NSF Cron Jobs

Both 101 and 103 have cron jobs. Use crontab -l to get the list of current cron jobs. Here you can get the location of where the scripts of data acquisition, name disambiguation, and document topic map are.