

# Visualization Environment for Rich Data Interpretation (VERDI 1.4.1): Developer Instructions

U.S. EPA Contract No. EP-W-09-023, "Operation of the Center for Community Air Quality Modeling and Analysis (CMAS)

Prepared for:	William Benjey and Donna Schwede
	U.S. EPA, ORD/NERL/AMD/APMB
	E243-04
	USEPA Mailroom
	Research Triangle Park, NC 27711

Prepared by: Liz Adams and Darin Del Vecchio Institute for the Environment The University of North Carolina at Chapel Hill 137 E. Franklin St., CB 6116 Chapel Hill, NC 27599-6116

Date: April 30, 2013

# Contents

1	Intr	oduction	1
2	Inst	all Developer Environment	2
	2.1	Download and Install Eclipse	2
	2.2	Download and Install NSIS Installer (Only for Windows)	2
	2.3	Install Subclipse Update 1.2x in Eclipse 3.5	2
	2.4	Install Java Development Kit	2
	2.5	Install Apache Ant	3
3	Star	t Eclipse	4
4	Imp	ort VERDI Source Code	7
	4.1	Select File→Import	7
	4.2	Check Out Projects from SVN	7
	4.3	Create a New Repository Location	8
	4.4	Specify Location of VERDI SourceForge Repository	9
	4.5	Select Folders for Checkout	.10
5	Che	ck Out the netCDF Java Library with Modifications for VERDI	.13
5 6	Che Buil	ck Out the netCDF Java Library with Modifications for VERDI d netCDF Outside Eclipse	.13 .14
5 6 7	Che Buil Set ]	ck Out the netCDF Java Library with Modifications for VERDI d netCDF Outside Eclipse Eclipse Preferences	.13 .14 .15
5 6 7	Che Buil Set 1 7.1	ck Out the netCDF Java Library with Modifications for VERDI d netCDF Outside Eclipse Eclipse Preferences Workspace Preferences	.13 .14 .15
5 6 7	Che Buil Set 1 7.1 7.2	ck Out the netCDF Java Library with Modifications for VERDI d netCDF Outside Eclipse Eclipse Preferences Workspace Preferences Verdi_core Properties	.13 .14 .15 .15
5 6 7	Che Buil Set 1 7.1 7.2	ck Out the netCDF Java Library with Modifications for VERDI d netCDF Outside Eclipse Eclipse Preferences Workspace Preferences Verdi_core Properties 7.2.1 Java Build Path	.13 .14 .15 .15 .16
5 6 7	Che Buil Set 1 7.1 7.2	ck Out the netCDF Java Library with Modifications for VERDI d netCDF Outside Eclipse Eclipse Preferences Workspace Preferences Verdi_core Properties 7.2.1 Java Build Path 7.2.2 Java Compiler	.13 .14 .15 .16 .16 .16
5 6 7	Che Buil Set 1 7.1 7.2	ck Out the netCDF Java Library with Modifications for VERDI d netCDF Outside Eclipse Eclipse Preferences	.13 .14 .15 .15 .16 .16 .16
5 6 7 8	Che Buil 5et 1 7.1 7.2	ck Out the netCDF Java Library with Modifications for VERDI d netCDF Outside Eclipse Eclipse Preferences	<ul> <li>.13</li> <li>.14</li> <li>.15</li> <li>.16</li> <li>.16</li> <li>.16</li> <li>.16</li> <li>.18</li> </ul>
5 6 7 8 9	Che Buil 7.1 7.2 Inst	ck Out the netCDF Java Library with Modifications for VERDI d netCDF Outside Eclipse Eclipse Preferences	<ul> <li>.13</li> <li>.14</li> <li>.15</li> <li>.16</li> <li>.16</li> <li>.16</li> <li>.16</li> <li>.18</li> <li>.21</li> </ul>
5 6 7 8 9 10	Che Buil 7.1 7.2 Inst Mod	ck Out the netCDF Java Library with Modifications for VERDI	<ul> <li>.13</li> <li>.14</li> <li>.15</li> <li>.16</li> <li>.16</li> <li>.16</li> <li>.16</li> <li>.18</li> <li>.21</li> <li>.22</li> </ul>

12	Update Source Code from the Repository	25
	12.1 Open the Synchronization Window	25
	12.2 Synchronize with Repository Using SVN	27
	12.3 Resolve Updates and Conflicts	29
13	Prepare to Build VERDI Distribution	30
	13.1 Microsoft Windows	30
	13.2 Linux	30
	13.3 Mac OS X	30
	13.3.1 Build_dist xml	31
14	Build VERDI	33
	14.1 Build Using Ant	33
	14.1.1 Microsoft Windows Distribution	33
	14.1.1 Linux Distribution	35
	14.1.1 Mac Distribution	35
	14.2 Check Console for Error Messages	36
	14.3 Add Java Compiler to Ant	36

# Figures

Figure 3-1 Select a Workspace	4
Figure 3-2 Eclipse Starting Up	5
Figure 3-3 Eclipse Workbench	6
Figure 4-1 File Import	7
Figure 4-2 Checkout Projects from SVN on Windows/Linux	8
Figure 4-3 Checkout Projects from SVN on Mac	8
Figure 4-4 Create New Repository Location	9
Figure 4-5 Check Out Code	10
Figure 4-6 Accept Digital Certificate	10
Figure 4-7 Select Folders for Checkout	11
Figure 4-8 SVN Checkout Includes a Meter Indicating % Completion	12
Figure 4-9 SVN imported Code into Workspace	12
Figure 7-1 Eclipse Window Preferences	15
Figure 7-2 verdi_core Properties Window	17
Figure 8-1 Select Jars for Export to Dependent Projects	19
Figure 10-1 Run VERDI within Eclipse	22
Figure 11-1 Run Configurations> Java Application	23
Figure 11-2 Run Configurations> Java Application> verdi_script	24
Figure 12-1 Show View → Other	25
Figure 12-2 Expand Team Folder, Highlight Synchronize, Click OK	26
Figure 12-3 Synchronize Window Added to Bottom of Workspace	
Figure 12-4 Click on Synchronize Symbol to Bring Up Pop-Up	27
Figure 12-5 In Synchronize Pop-Up, Select SVN	
Figure 12-6 In Synchronize SVN, Select All	
Figure 12-7 Check for Updates and Conflicts	29
Figure 12-8 Update verdi_core	
Figure 13-1 Review/Edit build.properties File	31
Figure 13-2 dist xml: NSIS Installer Path Section	32
Figure 14-1 Window→Preferences	33
Figure 14-2 Window→Show View→Ant	34
Figure 14-3 Double Click on build.win.dist to Build VERDI Distribution	35

Figure 14-4 Console Error Message	36
Figure 14-5 Open Windows→Preferences	37
Figure 14-6 Expand Ant, Select Runtime, Select Global Entries	38
Figure 14-7 Add tools.jar to Ant Preferences	38

# 1 Introduction

This manual contains instructions on how developers can set up, run, build, and obtain updates from the software repository for the Visualization Environment for Rich Data Interpretation (VERDI) software. Developers are encouraged to develop and contribute code for VERDI. Developers are requested to check the existing bug list and to work on resolving them, and to submit new bugzilla requests for enhancements to <a href="http://bugz.unc.edu">http://bugz.unc.edu</a>. Please submit test datasets, documentation and proposed code to the Community Modeling and Analysis System (CMAS) Center via the bugzilla website for testing prior to committing code to the repository.

Initial development of VERDI was done by the Argonne National Laboratory for the U.S. Environmental Protection Agency (EPA) and its user community. Argonne National Laboratory's work was supported by the EPA through U.S. Department of Energy contract DE-AC02-06CH11357. Further development has been performed by the University of North Carolina at Chapel Hill's (UNC) Institute for the Environment under U.S. EPA Contract Nos. EP-W-05-045 and EP-W-09-023, by Lockheed Corporation under U.S. EPA contract No. 68-W-04-005, and by Argonne National Laboratory. VERDI is licensed under the Gnu Public License (GPL) version 3, and the source code is available through verdi.sourceforge.net. VERDI is supported by the CMAS Center under U.S. EPA Contract No. EP-W-09-023. The CMAS Center is located within the Institute for the Environment at UNC.

# 2 Install Developer Environment

To install this software on Windows, you may need administrator privileges. You should exit all programs before installing software.

# 2.1 Download and Install Eclipse

- a. <u>http://www.eclipse.org/</u>
- b. Download and install Eclipse Classic.
  - i. Windows: install to the directory C:\ (you may need to avoid using the Program Files directory or any other directory name with a space in it)
  - ii. Linux/Mac: install to local directory
  - iii. Note: you may not be able to run Eclipse until you complete the installation of the Java Development Kit (see Section 2.4), as Java is required by Eclipse.

### 2.2 Download and Install NSIS Installer (Only for Windows)

- a. <u>http://nsis.sf.net</u>
- b. Download NSIS installer and install to the directory C:\ (you may need to avoid using the Program Files directory or any other directory name with a space in it)

### 2.3 Install Subclipse Update 1.2x in Eclipse 3.5

- a. Follow a link for zipped downloads: http://subclipse.tigris.org/servlets/ProjectDocumentList?folderID=2240
- b. Download the site-1.2.4.zip, zipped update site for Subclipse 1.2.4. (The website states that this is obsolete, but if you use a more recent version, it will not provide the Import>Other>Checkout Projects from SVN functionality that is needed.)
- c. Extract this zip file and save into the Eclipse directory.
  - i. Windows: you may need to authorize the overwriting of files.
  - ii. Linux/Mac: copy files in Plugin directory to Plugin directory under Eclipse.

### 2.4 Install Java Development Kit

a. http://java.sun.com/javase/downloads/index.jsp#jdk

Install the latest release of Java Development Kit (JKD), JDK 6 that is available from the above link. (Note: JDK 7 appears to work. JDK7 use in VERDI will be tested prior to use in the next release cycle.)

- b. For Linux 64
  - a. download the self extracting *<version>*-linux-x64.bin file
  - b. run the command chmod +x \*linux-x64.bin to give it executable permissions

- c. run the self extracting file ./jkd6\*-linux-x64.bin
- d. Run the alternatives program to tell the system about the existence of your new installation: Alternatives –config java (this will list how many versions are installed. If there is only one then install the jdk as number 2) alternatives --install /usr/bin/java java /opt/jdk1.6.0\_20/bin/java 2 Run the alternatives program again, to chose the new installation alternatives --config java, select number 2
- e. Verify that you are getting the right version of Java *\$ java -version*
- f. Repeat the above steps for javac: Alternatives – install /usr/bin/javac javac /opt/jdk1.6.0\_20/bin/javac 2 Alternatives –config java, select number 2
- c. For Windows 32: Add the Java compiler to your path on the Windows Machine.
  - i. From the <u>desktop</u>, right-click **My Computer** and click **Properties**, or click on **Start-Button**, Click on **Control Panel**, select **System**, select **Advanced System Settings**.
  - ii. In the System Properties window, click on the Advanced tab.
  - iii. In the Advanced section, click the **Environment Variables** <u>button</u>.
  - iv. Finally, in the Environment Variables window highlight the **Path** variable in the Systems Variable section and click the **Edit** button.
  - v. Add the location where the most recent JDK is available on your computer. For example: C:\Program Files\Java\jdk1.6.0\_17\bin
- c. For Window 7, if you are using Powershell, you will need to use the following command to set the path, then exit Powershell and restart it for the path to be set.
  - ii. setx PATH "\$env:path;\the\directory\to\add" -m
  - iii. you should see SUCCESS: Specified value was saved.
  - iv. **restart your session** and the variable will be available. setx can also be used to set arbitrary variables type setx /? at the prompt for documentation.

### 2.5 Install Apache Ant

Apache Ant is a software tool for automating the software build process.

Download the Apache Ant builder from the following website: <u>http://ant.apache.org/bindownload.cgi</u>

Add Ant's bin folder (e.g., "C:\apache-ant-1.8.1\bin") to the "path" environmental variable (semicolon delimited).

# 3 Start Eclipse

**Using Windows:** Go to the C:\Program Files\eclipse directory and double click on eclipse.exe. Specify a location for a workspace folder, for example, C:\VERDI\_1.4.1\_workspace. as is shown in Figure 3-1. Eclipse will create the directory automatically.

**Using a Mac:** Go to the applications directory, to the Eclipse folder, and click on the Eclipse icon. Specify a location for a workspace folder.

**Using a Linux machine:** Go to the directory where Eclipse was installed and run the eclipse executable. Specify a location for a workspace folder.

The start up window for Eclipse includes a progress bar at the bottom that gives you an indication of how far along it is in the process of preparing the workspace.

🔿 Workspace Launcher	
Select a workspace	
Eclipse SDK stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: C:(VERDI_I.4.ILWOrkspace	Browse
Use this as the default and do not ask again	OK Cancel

#### Figure 3-1 Select a Workspace

Figure 3-2 Eclipse Starting Up



To enter the developer workspace, click on the arrow at the right-hand side of the Welcome screen once you hover over the arrow it will say "Workbench – Go to the workbench" (Figure 3-3). The Eclipse workbench contains several windows that allow the user to view source code, edit, and build within a single developer environment (Figure 3-4).

Figure 3-3 Eclipse Workbench



# 4 Import VERDI Source Code

### 4.1 Select File →Import

To import the VERDI source code, use the mouse to select **File** $\rightarrow$ **Import** (Figure 4-1). This will generate a pop-up window titled Import.

### 4.2 Check Out Projects from SVN

In Windows/linux, expand the **Other** Folder in the Import window by clicking on the plus sign next to the folder titled **Other** (Figure 4-2). On a Mac, click on 🗂 and select the SVN wizard (Figure 4-3). Select **Checkout Projects from SVN** by clicking on it, then click next.

🖨 Java - Eclipse SDK						
File Edit Source Refacto	r Navigate Se	earch	Project	Run	Window	Help
New	Alt+Shift+N	۲ <u>ای</u>	# C		p A	• : 9 - 9 - • • • • • •
Open File		-6				
Close	Ctrl+W					
Close All	Ctrl+Shift+W					
📙 Save	Ctrl+S					
📓 Save As						
ng Save All	Ctrl+Shift+S					
Revert						
Move						
Rename	F2					
🔊 Refresh	F5					
Convert Line Delimiters To	)	•				
👜 Print	Ctrl+P					
Switch Workspace		•				
Restart						
r>1 Import						
🛃 Export						
Properties	Alt+Enter					
Exit			🕄 Prot	lems	x a	Javadoc) 🚱 Declaration

#### **Figure 4-1 File Import**



# 4.3 Create a New Repository Location

Use the mouse to highlight the button next to **Create a new repository location** (Figure 4-4), then click **next**.

Checkout from SVN	
Select/Create Location	SVN
Select an existing repository location or create a new location.	=0=
⊙ Create a new repository location	
Ouse existing repository location:	
(?) < Back Next > Einish	Cancel

Figure 4-4 Create New Repository Location

# 4.4 Specify Location of VERDI SourceForge Repository

Copy and paste the URL <u>https://verdi.svn.sf.net/svnroot/verdi</u>, into the location field (Figure 4-5), then click next.

#### Figure 4-5 Check Out Code

E Checkout from SVN	
Select/Create Location	SVN
Location Url: https://verdi.svn.sf.net/svnroot/verdi	
Image: Section of the section of t	Cancel

In the box that appears (Figure 4-6), click on the "Accept Temporarily" button to accept the digital certificate.

#### **Figure 4-6 Accept Digital Certificate**



### 4.5 Select Folders for Checkout

In the window that appears, click on Trunk (or Branch/2012\_06 to download the latest version of the code (VERDI\_1.4.1), then use the mouse and the shift key to highlight the following group

of subfolders: bootstrap, core, saf.core.ui, verdi.data.loaders, and verdi\_dist. Click finish (Figure 4-7). Eclipse will check out the latest version of VERDI from the repository. The SVN checkout routine provides the option to run in the background, and also provides a meter indicating the percentage of completion during the checkout process (Figure 4-8). A message will be displayed in a console window at the bottom of the workspace if there is an error. The console will display an error if you use a directory that already exists and it has permission problems in copying files to the workspace directory. The workspace and the directory where the VERDI software has been installed should not share the same location. Figure 4-9 shows that the code has been successfully imported into the workspace. If you have a red X by one of the folders, on the other hand, this indicates a problem. The Problems Tab at the lower right of Eclipse lists errors and what may be missing. If the problem or error is related to a XTiffImage.java, this will be fixed in Section 9, "Modifying TiffConvert.java."



#### **Figure 4-7 Select Folders for Checkout**

00	X	Java – Eclipse SDK				
<u>F</u> ile <u>E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> av	vigate Se <u>a</u> rch <u>P</u> roject <u>R</u> un	<u>W</u> indow <u>H</u> elp				
📑 • 🗄 G 🕹 🚸 • 🔍 •	🂁 · 🔌 🖶 🞯 · 🥭 🤞	? ▪ 앞 · 한 · *=		2		
				🙀 Quick Ac	cess 🖹 🖹	🎝 Java
😫 Package Explore 🕱 📟 🗖					🗄 Outline 🛙	
Ē 😫 ▽					An outline is not available.	
▶ 😂 2012_06	000	X SVN Checkout				
	Checking out verdi_dis	t				
	Always run in background					
	_ Aways ign in Suckyround					
		Cancel <u>D</u> e	etails >> Ru	n in <u>B</u> ackground	4	
	🖹 Problems 🕱 @ Javadoo	: 🗟 Declaration			~	- 0
	0 items					
	Description		Resource	Path	Location	Туре
		III				>
			SV	N Checkout: (21	.%)	

Figure 4-8 SVN Checkout Includes a Meter Indicating % Completion

Figure 4-9 SVN imported Code into Workspace

Java - Eclipse SDK	A REAL PROPERTY AND A REAL PROPERTY.					
<u>File Edit Navigate Search Project Rur</u>	<u>W</u> indow <u>H</u> elp					
i 📬 • 💷 🖻 i 🎄 • 🗿 • Q, • i 🖟	8 🗃 🞯 + 1 🍅 🛷 + 1 🖗 + 🖏 + 🤝 🤅				📑 🐉 Java	
🗎 Daskage Em 🛞 🧯 Hierarchy 🖓					Outline 8	
					outline is not availa	ble
verdi beetrtrap (trunk/beetrtrap)						
verdi core [trunk/core]						
🔐 verdi_data_loaders [trunk/verdi.data.lo						
verdi_dist [trunk/verdi_dist]						
🚰 verdi_saf_ui [trunk/saf.core.ui]						
	Problems X @ Javadoc & Declaration					~
	0 items					
	Description	Resource	Path	Location	Туре	
		i n a n			# : a l *	mm 66 ^
1		Building wor	rkspace: (73%)	-	۵ ا 🗠 📔 🔍	📧 😻 🔶

# 5 Check Out the netCDF Java Library with Modifications for VERDI

Checking out and downloading the netCDF Java code is needed only if you may need to debug an issue with reading in files that may result from a problem in the netCDF Java code.

- 1. Check out the netcdf project from <u>http://verdi.svn.sf.net/svnroot/verdi/netcdf</u>.
  - a. When in Eclipse, use File> Import> Other> SVN
  - b. Select Checkout Projects from SVN
  - c. Select use a new repository location
  - d. Put URL in the text box, click next
  - e. Select the netcdf folder
  - f. Check Out As a project in the workspace

# 6 Build netCDF Outside Eclipse

(You will need this section of the manual only if you want to make changes to netCDF source code.)

VERDI uses a version of netCDF that contains changes to the WRF Convention that allow it to read additional WRF variables, and also removes the M3IOVGGridConvention.java convention, as it resulted in an issue with reading polar stereographic grids incorrectly. These changes to the netCDF project have not been incorporated yet into the netCDF Java source code from Unidata, but this code is used by default in the VERDI builds. If you would like to make additional changes to the netCDF code, you will need to build a netCDF Java jar file outside of Eclipse, and then copy the netcdf.jar to the location where VERDI looks for it.

- 1. From a command prompt, go to the folder netcdf/nrSrc-4.2/cdm
- 2. Type in ant and hit enter key to build it.
- 3. The final jar file should be in the folder /netcdf/nrSrc-4.2/cdm/target
- 4. Copy netcdfAll-4.2.35.jar to /verdi\_core/lib/

# 7 Set Eclipse Preferences

### 7.1 Workspace Preferences

Eclipse can be set up to build the projects automatically after a developer makes local changes to the Java source code.

To automatically build after source code changes are made, enable this preference using: Eclipse> Window > Preferences > General > Click on Workspace > Select the Build Automatically option to the right of the Preferences Window (Figure 7-1).



**Figure 7-1 Eclipse Window Preferences** 

There is also a setting to automatically recognize files that are added to the workspace. To automatically synchronize the workspace with the underlying file system, enable this preference: Eclipse > Window> Preferences > General > Workspace > Select the Bofresh on Access or Bofresh Automatically or Build Automatically option

Select the Refresh on Access or Refresh Automatically, or Build Automatically option.

### 7.2 Verdi\_core Properties

In the Package Explorer view, right click on verdi\_core and select the Properties option at the bottom of the pop-up menu. A pop-up window titled Properties for verdi\_core will appear (Figure 7-2).

#### 7.2.1 Java Build Path

Select Java Build Path, and the window on the right will update showing four tabs, containing information on the **Source** folders, the required **Projects**, the **Libraries** (Java ARchives (JARS) and class folders on the build path), and the **Order and Export** (entries that are selected for export to dependent projects). After we add the JAI, JAI ImageIO, and Java3D to the Java Development Environment in Chapter 8, we will review the jars under the Libraries tab to verify that they are installed correctly.

#### 7.2.2 Java Compiler

Select Java Compiler, and the window on the right will update to show the version of the JDK that is currently being used by VERDI, and that a check mark is by the option to enable project specific settings for the Java Compiler.

#### 7.2.3 Project References

The netCDF project (in this screen shot it is the folder is named netcdf) is used by verdi\_core. To allow Eclipse to find the netCDF Java source code while debugging the verdi\_core project, place a check by the netcdf folder to indicate that it is a needed reference for verdi\_core.

Properties for verdi_core	
type filter text	Project References $(\neg \neg \neg \neg \neg$
<ul> <li>Resource</li> <li>Builders</li> <li>Java Build Path</li> <li>Java Code Style</li> <li>Java Compiler</li> <li>Java Editor</li> </ul>	Projects may refer to other projects in the workspace. Use this page to specify what other projects are referenced by the project. Project references for 'verdi_core':
Java Euror Javadoc Location 	Figure di bootstrap [branches/2012_06/bootstrap]     Figure di bootstrap [branches/2012_06/verdi.data.loaders]     Figure di data_loaders [branches/2012_06/verdi_dist]     Figure di saf_ui [branches/2012_06/saf.core.ui]
?	OK Cancel

#### Figure 7-2 verdi\_core Properties Window

# 8 Install JAI, JAI-ImagelO, and Java 3D to the JDK

- a. For Windows 32-bit or 64-bit JDK:
  - i. Download JAI from

http://download.java.net/media/jai/builds/release/1\_1\_3/ (choose the jai\*1586.exe file ) and install it into JDK directory. Comprehensive installation instructions are available from http://download.java.net/media/jai/builds/release/1\_1\_3/INSTALL.html# Windows.

After downloading the executable to a folder, double click on it, and an InstallShield Wizard will start. Specify the Destination Folder as the location where you installed the JDK, for example: C:\Program Files\Java\jdk1.7.0\_17\jre.

ii. Download JAI-ImageIO from <u>http://download.java.net/media/jai-imageio/builds/release/1.1/</u> and install it into the jre directory under your JDK.

Download jai\_imageio-1\_1-lib-{yourplatform}.exe by clicking on it and saying save as. Run the executable by double clicking on it to start the InstallShield Wizard. Specify the Destination Folder as the location where the JDK was installed, for example C:\Program Files\Java\jdk1.7.0\_17\jre.

- iii. Download and install Java 3D: http://java3d.java.net/binary-builds.html
  - Save to a folder. Right click on the zip file to start the Extraction Wizard. Select the folder to extract files, for example C:\Program Files\.
  - Change to the directory C:\Program Files\j3d-1\_5\_2-windows-i586 and right click on j3d-jre.zip to extract to the JDK directory, for example C:\Program Files\java\jdk1.7.0)17\jre.
  - Instructions are available in the following readme: <u>http://download.java.net/media/jai-</u> <u>imageio/builds/release/1.1/INSTALL-jai\_imageio.html</u>.
- b. For Linux
  - i. Add the compiler to a local directory.
  - ii. Install JAI, JAI-ImageIO, and Java 3D:
    - Download 32- or 64-bit JAI that matches your JDK from <u>http://download.java.net/media/jai/builds/release/1\_1\_3/</u> and install it into the JDK directory. More comprehensive install instructions are available from <u>http://download.java.net/media/jai/builds/release/1\_1\_3/INSTALL.</u> <u>html#Linux</u>.
    - Download 32- or 64-bit JAI-ImageIO that matches your JDK from <a href="http://download.java.net/media/jai-imageio/builds/release/1.1/">http://download.java.net/media/jai-imageio/builds/release/1.1/</a> and install it into the JDK directory.

Note: The 64-bit version of Jai-<u>ImageIO</u> is corrupt. Fix it after download and use the fixed version:

```
sed s/+215/-n+215/ jai_imageio-1_1-lib-linux-
amd64-jdk.bin > jai_imageio-1_1-lib-linux-
amd64-jdk-fixed.bin
sed s/+215/-n+215/ jai_imageio-1_1-lib-linux-
i586-jdk.bin > jai_imageio-1_1-lib-linux-i586-
jdk-fixed.bin
```

- Download 32- or 64-bit Java 3D that matches your JDK and install it: <u>http://java3d.java.net/binary-builds.html</u>.
- Extract the j3d-jre.zip according to the instructions in the README-unzip.html in the downloaded version.
- In the package explorer in Eclipse, select Window> Preferences, select Installed JRE's, highlight the jdk1.6.0\_45 and click edit. Check to see that the jai and j3d jar files are all listed. If they are not there, add them as external jars..
- Go to verdi\_core, right click and select properties. Click on the Order and Export tab confirm that the j3dcore.jar, j3dutils.jar, and vecmath.jar are there. Note: all of the jar files in the order and export tab should have check marks next to them, except for the projects and jar files that are shown as unchecked in Figure 8-1.

#### Figure 8-1 Select Jars for Export to Dependent Projects



- c. For Mac
  - i. Java Advanced Imaging (JAI) Library is already installed on Mac OS X. Use Software Update to update it to the latest version.
    - Check version using command: javac -version
    - Check path using command: which javac
  - ii. The Image I/O tools jar file may be obtained from a linux system, where you have run the installer to obtain a *jai\_imageio.jar* file
  - iii. See the following link: <u>http://gvsigce.sourceforge.net/wiki/index.php/Java\_Advanced\_Imaging\_(J\_AI)\_and\_64\_bit</u>.
  - iv. Install Java3D to support the contour plot on Mac for VERDI: <u>http://www.downloadjava3d.com/mac.php</u>

# 9 Modify TiffConvert.java within verdi\_core

VERDI allows the user to save an image in the tiff format using the following command within the VERDI Plots menu options: File> Export> File Format: Tiff Image

Currently, the code within VERDI, TiffConvert.java, is platform dependent. There is a Win64 version of TIFConvert.java called TIFConvert.java.Win64 that needs to be copied to TIFConvert.java file when you are building VERDI for the Windows 64 platform. When you are building on all other platforms, i.e., Mac, Linux, or Windows 32, the TIFConvert.java.NonWin64 version needs to be copied to the TIFConvert.java file. This allows the following command from the VERDI plots menu to work on all platforms: File> Export image> File Format: Tiff Image

From the command prompt or using Windows Explorer, go to the Eclipse workspace folder, i.e., C:\VERDI\_1.4.1\_workspace\ verdi\_core\src\anl\Verdi\plot\io, and delete the file TIFConvert.java.

For 64 bit JDK under windows, rename the TIFConvert.java.Win64 to TIFConvert.java.

For the 32 bit Windows version of JDK, rename <code>TIFConvert.java.NonWin64</code> to <code>TIFConvert.java.</code>

For the 32 bit Linux version of JDK, rename TIFConvert.java.linux32 to TIFConvert.java.

For the 64 bit JKD for Linux, rename TIFConvert.java.linux64 to TIFConvert.java

Now, in Eclipse, right click on verdi\_core, and click refresh. Then click on Project> Clean, and that will force eclipse to rebuild the project.

# **10 Run VERDI within Eclipse**

You can run VERDI using the verdi.launch script in verdi\_bootstrap. Go to the verdi\_bootstrap folder, and right click on verdi\_launch. Select Run As→verdi (Figure 10-1)

🖨 Java - Eclipse SDK								
File Edit Navigate Search Project Run Wi	ndow Help							
i 🐟 🚛 💿 i 🐟 🗸 New	•	:	- *5 /5 - ~ -				<b>F</b> \$ <b>#</b>	d must
Show In Open Open Open With	Alt+Shift+W ≯ F3							
Copy	Ctrl+C							
	Ctrl+V							
🕀 📑 jpf-boot, jar 10 4, 💢 Delete	Delete							
💷 🔄 log4j-1.2.13.jar 1 🛛 Build Path	•							
🗷 🛋 JRE System Librar 🛛 Move								
🗷 🔤 saf.core.runtime.; Rename	F2							
🗄 🔄 jai_core.jar 10 4/ 🕀 📷 jai_imageio.jar 10 🚵 Import								
🕀 📑 j3dcore.jar 10 4/ 🚮 Export								
i3dutils.jar 10 4/1	F5							
🕀 🗁 data 🔹 Run As	Þ	1 verdi						
🗈 🗁 help Debug As	•	J I Voldi						
🗉 🗁 lib Team	•	Run Configur	ations					
Compare With	•							
Boot.properties 2. Replace With	•							
bootstrap.iml 10								
unproperties 10 Properties	Alt+Enter	J						
werdi, core [trupk/core]								
wind_core [dangeore]								
e 🙀 verdi dist [trunk/verdi dist]								
🗉 🛋 JRE System Library [jre6]								
🗉 🗁 data	🛣 Problems 🐹 🖉 @	Javadoc 🔯 De	eclaration					
🗈 🔄 distfiles	0 errors, 1,170 warnings,	0 others (Filter m	natched 100 of 1170 iter	ns)				
🕮 🥁 distlib	Description A		Resource	Path	Locat	Туре		
- august 205 10/16/09 10:34 AM	🗄 👶 Warnings (100 o	f 1170 items)						
build.properties 46 1/16/09 11:09 AM								
build.properties. IEMPLATE 18 7/25/08								
< >								
🕴 📑 verdi.launch - verdi_bootstrap				1			ļ	@   ♀ ⊠ ♥ ♦
Model Data, Importing Obser	vation Data, Projects	Selections of	or Domains. Lave	s. and time-			40.0%	
Start 🔂 8 Wi 🔹 🥹 Eclips	🚚 4 SS 👻 🥞	Inbox	🔄 VERDI 👘 🦉 unt	tle 🦉 Ven	diD 🥃 2	ecl 👻	100%	🔨 🚾 🎦 8:23 PM

#### Figure 10-1 Run VERDI within Eclipse

# **11 Test VERDI Using Scripts within Eclipse**

Scripts are available for testing VERDI within Eclipse for several plot types. Prior to using these scripts, extract the data.zip file under the verdi\_dist/distfiles directory using the commands cd verdi\_dist/distfiles unzip data.zip.

From the Eclipse Main Menu, select either Run>Run Configurations or Run> Debug Configurations (if you want to run the script within the debugger).

Select Java Application to view the scripts (Figure 11-1).

000	Run Configurations	
Create, manage, and run config	urations	
Eclipse Application ■ Eclipse Application ■ Java Applet ■ Java Application JuJUnit JUJUnit Plug-in Test → OSGi Framework	<ul> <li>Configure launch settings from this dialog:</li> <li>Press the 'New' button to create a configuration of the selected type.</li> <li>Press the 'Duplicate' button to copy the selected configuration.</li> <li>Press the 'Delete' button to remove the selected configuration.</li> <li>Press the 'Filter' button to configure filtering options.</li> <li>Edit or view an existing configuration by selecting it.</li> <li>Configure launch perspective settings from the 'Perspectives' preference page.</li> </ul>	
Filter matched 13 of 15 items		
(?)		Close Run

### Figure 11-1 Run Configurations> Java Application

The script names include verdi\_script, verdi\_script\_batch, verdi\_script\_camx, verdi\_script\_two\_bars, and verdi\_script\_vertical\_crosssection. Select verdi\_script, and then click on the arguments tab to view the command line arguments that are passed to VERDI in the script (Figure 11-2). When you run the script, VERDI automatically loads the data and creates plots using the script commands specified in the arguments tab. Setting up and running scripts

shortens the time it takes to debug plot issues, as it is easier to quickly reproduce plots. (Note: The path names are specified relative to the distfiles/data directory in the arguments tab. This allows developers to run the test scripts on different platforms [Windows, Linux, or Mac] without having to edit the path name to load the data correctly.)

$\Theta \odot \odot$		Run Configurations	
Create, manage, and run configura Run a Java application	tions		
Yee       Yee         type filter text <ul> <li>Eclipse Application</li> <li>Java Applet</li> <li>Java Application</li> <li>JomainPanel</li> <li>NetcoffDataset</li> <li>NetcoffFile</li> <li>verdi_script_batch</li> <li>verdi_script_camx</li> <li>verdi_script_vertical_crossse</li> <li>Juplinit</li> <li>JUpint</li> <li>JUpint</li> <li>OSGi Framework</li> </ul>	Name: verdi_script             Main (** Arguments)         Arguments         Arguments         Program arguments:         -f "/verdi_dist/distfiles/data/model/CCTM46_P16.baseO2a.36k.O3MAX" -s "O3[1]" -configFile/verdi_dist/         distfiles/data/configs/o3_10bin.cfg -g tile		
	Working direct	tory:	
	O Other:	Workspace File System Variables	
Filter matched 14 of 16 items		Apply Revert	
?		Close	

Figure 11-2 Run Configurations> Java Application> verdi\_script

# **12 Update Source Code from the Repository**

If you have previously obtained the code from the repository and want to obtain any bug fixes or new features that developers have committed to the repository, the synchronization window allows you to do this.

# 12.1 Open the Synchronization Window

Select Window  $\rightarrow$  Show View  $\rightarrow$  Other to open the **Show View** pop-up window (Figure 12-1). Expand the Team Folder (Figure 12-2) by clicking on the plus symbol, then highlight the word Synchronize and click OK. The synchronize window will be created at the bottom of the workspace (Figure 12-3).

🖨 Java - Eclipse SDK				
File Edit Navigate Search Project Run	Window Help			
📬 • 🖫 🗁   🏇 • 🕥 • 💁 •	<b>New Window</b> New Editor	2 - Q - 🏷 🤇	⇔ - ⇔ -	
🛱 Package Explor 🛛 🍃 Hierarchy 🗖	Open Perspective	-		
E 🕏	Show View 🔹 🕨	∦ Ant		
🗐 😭 verdi_bootstrap [trunk/bootstrap]	Customize Perspective	📃 Console	Alt+Shift+Q, C	
🗁 verdi_core	Save Perspective As	🚯 Declaration	Alt+Shift+Q, D	
🗈 📷 verdi_data_loaders [trunk/verdi.data.l	Reset Perspective	👰 Error Log	Alt+Shift+Q, L	
verdi_dist [trunk/verdi_dist]	Close Perspective	🍃 Hierarchy	Alt+Shift+Q, T	
ter ata	Close All Perspectives	@ Javadoc	Alt+Shift+Q, J	
	Navigation 🕨 🕨	😋 Navigator		
🗄 🗁 🗁 data	Preferences	📴 Outline	Alt+Shift+Q, O	
Config.properties 96 4/2/09 11	,10	🛱 Package Explorer	Alt+Shift+Q, P	
	AN	R Problems	Alt+Shift+Q, X	
ire1.6.0.zip 94 4/2/09 5:06 PM	l qu	0 Progress	-	
Ed ICENSES.ZIP 18 7/25/08 11:24 - B PELEASE-NOTES byt 239, 12/1	41Y1	Project Explorer		
run.bat 165 5/29/09 6:24 PM	aur	Search	Alt+Shift+O, S	
shape2bin.zip 141 5/20/09 5:5	6 P	Tasks	0	
	9.10			
	qur	- Templaces		
	4 q	Other	Alt+Shift+Q, Q	
👘 verdi.sh 224_11/23/09 1:12 PM	1 q/			
Oistlib     Devild dict yml 225, 11/20/00, 10/28	DM Problems 🗐 Synchro	nize 🕅		
build_properties 46_1(16(09.11:09				
build.properties.TEMPLATE 18 7/2	5/08			

Figure 12-1 Show View → Other



Figure 12-2 Expand Team Folder, Highlight Synchronize, Click OK

Figure 12-3 Synchronize Window Added to Bottom of Workspace



### 12.2 Synchronize with Repository Using SVN

The Synchronize window contains a synchronize symbol in the upper right corner (Figure 12-4). Click on the synchronize symbol to open the Synchronize pop-up window. Click on SVN to select the subversion software package manager, then click NEXT (Figure 12-5). A Synchronize SVN pop-up window will appear listing the packages that are available as resources for synchronization (Figure 12-6). Click the Select All button to select all the packages, then click FINISH. A pop-up window labeled Confirm Open Perspective will ask if you would like to change from the Java Perspective to the Team Synchronization Perspective. If you opt to change perspectives, there is a right arrow button in the upper right-hand corner of the workspace to switch back to the Java Perspective once you are finished reviewing the code in the Team Synchronization Perspective.



Figure 12-4 Click on Synchronize Symbol to Bring Up Pop-Up

🖨 Synchronize	
Synchronize Synchronize files managed with SVN	
Available synchronization types:	
(?) < <u>Back</u> <u>Next</u> > Einish	Cancel

#### Figure 12-5 In Synchronize Pop-Up, Select SVN

#### Figure 12-6 In Synchronize SVN, Select All

		X Unknown			
Synchronize SVN Select the resource	l es or working sets to be syn	nchronized.			
Available resources	s to Synchronize:				
👂 🗹 🚔 verdi_boo	otstrap [branches/2012_06/t	bootstrap]			
👂 🗹 🚔 verdi_con	e [branches/2012_06/core]				
👂 🗹 🚔 verdi_data	a_loaders [branches/2012_	06/verdi.data.load	ers]		
👂 🗹 😭 verdi_dist	[branches/2012_06/verdi_	dist]			
👂 🗹 🚔 verdi_saf	_ui [branches/2012_06/saf.@	core.ui]			
Select <u>A</u> ll	eselect All				
Scope					
Workspace	Selected Resources				
○ <u>W</u> orking Set: (					Choose
?		< <u>B</u> ack	<u>N</u> ext >	Cancel	Einish
					11.

### 12.3 Resolve Updates and Conflicts

After synchronization, there will be a blue incoming arrow at the bottom of Eclipse window that reports the number of files on the repository that are different than your local workspace, a green outgoing arrow, and red double conflict arrow (Figure 12-7). If you click on the blue arrow, and then right click on the Verdi\_core, a pop-up menu will allow you to select Update, which will bring the updates to the code down to your local workspace (Figure 12-8).

e ci	/S Repository Exp	oloring - verdi_dist/build_dist.xml - Eclipse SDK	
File	<u>E</u> dit <u>N</u> avigate Se <u>a</u> i	rch <u>P</u> roject C <u>V</u> S <u>R</u> un <u>Wi</u> ndow <u>H</u> elp	
1 63	• 🛛 🗠 🛛 🗛	•   <b>ペ</b> •   😭   ふ J   2 - 知 · 🍄 ゆ • ウ ·	😰 🛃 CVS Reposito 🧯 🎽
	1 P 🛛 🗖 🗖	🔁 Project Explorer ( 🏶 Ant 🖆 Synchronize 🕴	🗗 - 군 🕹 😯 🖻 🔮 👘 😭 👄 🏹 🗖 🗋
8		SVN (Workspace)	
	E Cal verdi boots	The second secon	
	🕀 🙀 verdi_core [		
	🕀 🚮 verdi_data_		
	🖨 🚰 verdi_dist [t		
	I IRE Sys		
	🗷 🔄 distlib		
	📑 build_di:		
	👘 build_di:		
	build.pro		
	🗄 🖂 verdi saf u		
	< >		
		6 files modified on repository	
	ЖIР XX С С	1 file modified locally	
: □¢		(a 6) (b 1) (b 0) Writable (c)	al i 🖉 🖉 😵 🔶

Figure 12-7 Check for Updates and Conflicts

Figure 12-8 Update verdi\_core



# **13 Prepare to Build VERDI Distribution**

Once VERDI has been checked out of the repository, the folders will be displayed in the Project Explorer Window on the Workbench. The next step is to edit the build.properties file.

### 13.1 Microsoft Windows

If you are building for the Windows platform, open and edit the build.properties.win32 or build.properties.win64 file that matches your JDK by double clicking on the file to open it in the text editor (Figure 14-1). Edit the build properties file to specify the JDK used to compile and the directory where Eclipse will build the VERDI distribution, and to specify the directory on your computer that contains the NSIS installer software. Example JDK location: C:\\Program Files\\Java\\jdk1.6.0\_12

# 13.2 Linux

Edit the build properties file (build.properties.linux32 or build.properties.linux64 that matches your JDK) to make it specify the JDK used to compile and the local directory where Eclipse will build the distribution. Right click on the build properties file and select Save. Example JDK location:

/home/lizadams/jdk1.6.0\_45

# 13.3 Mac OS X

Edit the build properties file (build.properties.mac) to specify the JDK used to compile and the local directory where Eclipse will build the distribution. Right click on the build properties file and select Save. Example JDK location:

/System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/



Figure 13-1 Review/Edit build.properties File

#### 13.3.1 Build\_dist xml

There are five build\_dist xml files available for Ant building, each for a specific version of JDK: build\_dist\_linux32.xml, build\_dist\_linux64.xml, build\_dist\_win32.xml, build\_dist\_win64.xml, and build\_dist\_mac.xml. The build\_dist\_platform.xml files provide the instructions for how to build the respective Mac OS X, Linux, and Windows distributions of VERDI. The build\_dist\_platform.xml file obtains the local directory settings from the corresponding build properties file. It is also possible to make changes to specify these directories in the build\_dist xml file, but the build properties file has been created to clearly identify what settings are dependent on the local directory configurations, and should reduce errors that might be incurred by a user editing the build\_dist xml file. Figure 14-2 shows the section of the build\_dist xml file that provides the paths for the installer used to build the Windows distribution, if you really want to edit it in the build file.



Figure 13-2 dist xml: NSIS Installer Path Section

# 14 Build VERDI

# 14.1 Build Using Ant

Before building the releases, a few settings have to be done through the Eclipse menu options Window  $\rightarrow$  Preferences.... Once the Preferences dialog is open (Figure 14-1), select General then Workspace. In the Workspace panel, select the check boxes for Build automatically and Refresh automatically. Click the OK button to close the preferences dialog.



#### Figure 14-1 Window→Preferences

### 14.1.1 Microsoft Windows Distribution

The Windows distribution can be built using the scripts (build\_dist\_win32.xml or build\_dist\_win64.xml) within verdi\_dist on a Windows machine. Select the Eclipse menu options Window $\rightarrow$ Show View $\rightarrow$ Ant to create a subwindow for Ant (Figure 15-2). Drag the

corresponding build\_dist\_{machineOS}.xml into the Ant window. Click on the plus button next to verdi to open and display the contents.

- 1. Double click on build-version to update the build version number
- 2. Double click on compile-version to update the compile version number
- 3. Double click on build.win.dist to build the VERDI distribution for a Windows machine (Figure 15-3).



Figure 14-2 Window→Show View→Ant



Figure 14-3 Double Click on build.win.dist to Build VERDI Distribution

### 14.1.1 Linux Distribution

The Linux distribution can be built by using Ant to run build.linux.dist on a Linux machine. The verdi\_dist folder contains the build\_dist\_linx32.xml or build\_dist\_linx64.xml script. Select the Eclipse menu options Window  $\rightarrow$  Show View  $\rightarrow$  Ant to create a subwindow for Ant (Figure 14-2). Drag the corresponding build\_dist xml into the Ant window. Click on the plus button next to verdi to open and display the contents.

- 1. Double click on build-version to update the build version number
- 2. Double click on compile-version to update the compile version number
- 3. Double click on build.linux.dist to build the VERDI distribution for a Linux machine.

#### 14.1.1 Mac Distribution

The Mac distribution can be built by using Ant to run build.mac.dist on a Mac OS X machine. The verdi\_dist folder contains the build\_dist\_mac.xml script. Select the Eclipse menu options Window $\rightarrow$ Show View $\rightarrow$ Ant to create a subwindow for Ant (Figure 14-2). Drag the

corresponding build\_dist xml into the Ant window. Click on the plus button next to verdi to open and display the contents.

- 1. Double click on build-version to update the build version number
- 2. Double click on compile-version to update the compile version number
- 3. Double click on build.mac.dist to build the VERDI distribution for a Mac OS X machine.

### 14.2 Check Console for Error Messages

Error messages will appear in the console window underneath the Ant window.



Figure 14-4 Console Error Message

If you obtain this error, add the Java compiler to your path on the Windows or Linux machine (please review Sections 2.4 and 7.2.2 to resolve and fix the problem)

# 14.3 Add Java Compiler to Ant

(You will need this section of the manual only if you obtain the error shown in Figure 14-4.)

To allow the Ant compiler to find the Java compiler, you will also need to change the Ant Preferences to add tools.jar as an external jar as follows:

- 1. In the Eclipse main menu, select Window  $\rightarrow$  Preferences (Figure 14-5)
- 2. In the Preference Window, select Ant $\rightarrow$ Runtime (Figure 14-6)
- 3. Click on the Classpath tab, then select Global Entries
- 4. Click on Add External JARs
- 5. Locate the tools.tar under the lib folder on the JDK local installation directory, then click OK, and click OK again. (Figure 14-7)



Figure 14-5 Open Windows→Preferences

Preferences		
type filter text	Runtime	⇔• ⇒ • ▼
Pype Hier text	Runtime       Settings used when running Ant buildfiles:       Image: Setting the setting of the setting the set	Add JARs Add External JARs Add Eolder Add Yariabie Ark Home Rgmove Up Dogon
3	Rest	ve Defaults Apply OK Cancel

Figure 14-6 Expand Ant, Select Runtime, Select Global Entries

Figure 14-7 Add tools.jar to Ant Preferences

