

Space Details

Key:	CLT
Name:	Software Infrastructure
Description:	
Creator (Creation Date):	kgomes (Oct 18, 2006)
Last Modifier (Mod. Date):	kgomes (Apr 25, 2016)

Available Pages

- Collaboration Tools Home Page
 - CVS at MBARI
 - Adding code to CVS Repository with WinCVS GUI
 - CVS Client Software Downloads
 - Setting up a SVN Graphical User Interface (GUI)
 - Setting up CVS Graphical User Interface (GUI)
 - Setup to access CVS Repository with Command line instead of GUI (Linux users)
 - Welcome to CVS!
 - Deployment Information
 - Installation Notes
 - Java Signing at MBARI
 - MBARI Acronym Page
 - SVN at MBARI
 - Adding your code to MBARI's SVN repository
 - Secure SVN Development
 - SVN Client Software Downloads
 - Setting up SmartSVN
 - UML
 - Artisan Studio
 - IBM Rational Software Architect
 - MagicDraw
 - VisualParadigm
- SearchNavigation
- TreeNavigation
- Software Infrastructure Home 
 - MBARI Central Authentication Service

Collaboration Tools Home Page

This page last changed on Dec 17, 2010 by [kgomes](#).



Introduction

This space contains information about the collaboration and design tools that we use at MBARI. The links below will take you to the MBARI installation of each tool. For a more detailed description of the tools themselves, visit their respective websites.

Contact Kevin Gomes kgomes@mbari.org, Brian Schlining brian@mbari.org or Kent Headley headley@mbari.org with questions or permission requests for certain tools.



External Availability

Please note that Confluence, JIRA, and Alfresco are all available outside the MBARI firewall and can have external collaborator accounts that are not MBARI accounts. This was a desirable feature of these tools and allows us to externally collaborate without risking security breaches at MBARI.

Best Practices

When creating 'spaces' (collaborative areas) in the collaborative development environments, we ask users to start the project name with the MBARI project number, if there is one. Follow the templates of the other projects. (Note that an MBARI project number is *not* required in order to add a space to these environments.)

We encourage users to store most content, and **all** documentation related to MBARI engineering projects, in the Alfresco environment. Its content management features are extensive, and its content will be much more readily accessible within MBARI than the content in other spaces. Confluence is particularly appropriate to make multiple discussions and pages (text or HTML) accessible and editable by MBARI staff, especially for informal content.

Authentication and Single Sign On With Crowd

[Crowd](#) is an authentication tool that is developed by [Atlassian](#) and is provided to MBARI under a community/non-profit license. It provides an internal user database so we can create external user accounts that can then be linked to other applications such as JIRA and Confluence. It also connects to our internal MBARI user database using LDAP so that MBARI users can have Single Sign on capability across multiple applications.

Links for Crowd at MBARI:

- [Crowd Web Application](#)
 - [Documentation of MBARI Crowd Installation](#)
-

Content Management with [Alfresco](#)

Alfresco is a Content Management System (CMS) intended to replace the Project Library on Tornado as the repository for MBARI project documentation. New projects are typically created on Alfresco rather than the Project Library share, and old projects are migrated as their owners see fit.

Accounting Creates Projects in Alfresco

One of the main benefits of Alfresco is that it can interact with our accounting system in an automated fashion. Because of this, we have the accounting system currently creating project spaces and directory structures in Alfresco automatically. This allows the accounting software to then export all accounting activity (PDFs of AMEX, PO's, etc.) to the Alfresco spaces automatically.

Your first visit you may see the Guest page. You must log in to Alfresco (MBARI network login/password without using "shore\" in front) to see the project library materials. The **Projects** folder is the space that contains the content for projects at MBARI.

To create a short cut to the Projects folder (or any folder), enter that folder, then select "More Actions/View Details", then "Create Shortcut" (on the right side of the page). This will create a shortcut that will be available in the "Shelf". To view the shortcuts in your "Shelf", click on the down arrow next to "Navigator" on the left hand sidebar menu, then select "Shelf" and you will see your created shortcuts.

Here are some useful links related to MBARI's installation of Alfresco:

- [Alfresco's Homepage](#)
- [Alfresco's Developer Home Page](#)
- [Alfresco's Enterprise Customer Portal](#)
- [Deployment Information for MBARI's Alfresco Installation](#)
- [Installation Notes](#)
- [alfresco.mbari.org Installation Notes](#)
- [alfresco-test.shore.mbari.org Installation Notes](#)

WIKI Collaboration Pages ([Confluence](#))

Confluence provides an environment to collaboratively work on activities and projects. Log in with your network account.

Navigation starts from the Dashboard, the page you see on entry. Use the blue navigation bar at the top to get back to Dashboard. Some material is only visible if you are logged in, or if you have specific permission to view it.

Creating a 'team' in Confluence lets you organize a set of spaces to appear for that team.

Contact Kevin Gomes to create a new space or team in Confluence.

1. [Confluence Installation Notes](#)

Issue (bug) tracking ([JIRA](#))

JIRA is a sophisticated bug tracking tool with advanced search features. It is also well suited to tracking tasks and feature requests for a project.

JIRA has a small number of projects tracking bugs and issues on it now, but is being considered as a replacement for phpbt, the engineering bug tracking database.

Log in with your network account. You may find the Browse Projects tab (in the blue task bar at the top) a more convenient way to see the projects in JIRA.

Contact Kevin Gomes to create a new JIRA project.

Maven Dependency Repository [Maven](#)

MBARI hosts an externally accessible Maven repository at <https://oceana.mbari.org/maven2>, an internal repository at <http://seaspray.shore.mbari.org:8082/archiva/repository/internal/> and a repository manager at <http://seaspray.shore.mbari.org:8082/archiva>. [Maven](#) is a software project management and comprehension tool. Similar to [Ant](#), Maven offers powerful dependency management features. The core of dependency management is the *Maven repository* where builds of Java code can be shared with others and easily included in projects. By using repository aware build tools, such as [Maven](#) or Ant (via [Ivy](#)), projects can be built without requiring the developer to check in all the dependencies in local version control. Instead, the dependencies are fetched from the repository by the build tools when they are needed.

See [Maven At MBARI](#) for more details.

Modeling with UML (JBuilder 2008 R2)

You can download JBuilder 2008 R2 from here:

[Mac OS X](#)

[Linux](#)

[Windows](#)

Once downloaded and installed, you need to copy the following [.slip](#) file to the:

JBUILDER_HOME/license

directory.

Concurrent Versioning System

CVS (Concurrent Versioning System) is an open source software tool used by software developers to manage code changes within their source code tree. This source code versioning system is the predecessor of Subversion (see below) and applies stricter security access when editing MBARI project files. Similar to Subversion, CVS enables programmers to store current and all previous records of code changes (along with those who made the changes) over the existence of the code. Access to projects on the CVS repository at MBARI, however, also requires a pre-set individual passwords for all users in order to make changes to the code. Without access permission, users may only view and download code - not upload it into the repository to make changes. Access is given to users who are part of that project's group on CVS. To gain a password or access to edit a project requires notification to the server administrator: Karen Salamy (salamy@mbari.org).

In addition to acquiring a password from the server administrator, you must also install a CVS software client. There are multiple CVS clients for most major operating systems including WinCVS (Windows), MacCVS (Macs), and gcvs (Linux). The CVS repository may also be accessed using the command line with the following syntax [CVS command Line Reference](#). The most common CVS client at MBARI is WinCVS (version 1.2) which is a stable Windows version release. Additional information on these clients can be found at <http://cvsgui.sourceforge.net/>; Detailed documentation on using CVS can be found at <http://ximbiot.com/cvs/manual/>.

The MBARI CVS repository may also simply be "viewed" from a web browser for any users who wish to look at its existing source code tree structure at <http://moonjelly.shore.mbari.org/cgi-bin/cvsweb.cgi/>.

See [CVS at MBARI](#) for more details.

Subversion

SVN (Subversion) is a tool used by many software developers to manage changes within their source code tree. SVN provides the means to store not only the current version of a piece of source code, but a record of all changes (and who made those changes) that have occurred to that source code. Use of SVN is particularly common on projects with multiple developers, since SVN ensures changes made by one developer are not accidentally removed when another developer posts their changes to the source tree.

In order to access a Subversion repository, you must install a special piece of software called a Subversion client. Subversion clients are available for most any operating system. Information about the official subversion client can be found at <http://subversion.tigris.org/>. Excellent documentation on using Subversion can be found at <http://svnbook.red-bean.com/>

The MBARI SVN repository may also simply be "viewed" from a web browser for any users who wish to look at its existing source code tree structure at <http://kahuna.shore.mbari.org/viewvc/?root=svn>.

See [SVN at MBARI](#) for more details.

Java Security Certificate

If you intend to distribute java applications via Java Web Start, you will need to *sign* your jar files. Details about how to do this can be found at [Java Signing at MBARI](#)

CVS at MBARI

This page last changed on Nov 10, 2010 by [salamy](#).

Concurrent Versioning System

CVS (Concurrent Versioning System) is an open source software tool used by software developers to manage code changes within their source code tree. This source code versioning system is the predecessor of Subversion and applies stricter security access when editing MBARI project files. Similar to Subversion, CVS enables programmers to store current and all previous records of code changes (along with those who made the changes) over the existence of the code. Access to projects on the CVS repository at MBARI, however, also requires a pre-set individual passwords for all users in order to make changes to the code. Without access permission, users may only view and download code - not upload it into the repository to make changes. Access is given to users who are part of that project's group on CVS. To gain a password or access to edit a project requires notification to the server administrator: Karen Salamy (salamy@mbari.org).

NOTE: In the rare event the administrator is on vacation or unavailable, please submit your CVS request to the IS Remedy page at: <https://ishelp.shore.mbari.org/helpdesk/WebObjects/Helpdesk> and it will be assigned by IS appropriately based on urgency level.

In addition to acquiring a password from the server administrator, you must also install a CVS software client. There are multiple CVS clients for most major operating systems including WinCVS (Windows), MacCVS (Macs), and gcvs (Linux). The CVS repository may also be accessed using the command line with the following syntax CVS command Line Reference. The most common CVS client at MBARI is WinCVS (version 1.2) which is a stable Windows version release. Additional information on these clients can be found at <http://cvsgui.sourceforge.net/>; Detailed documentation on using CVS can be found at <http://ximbiot.com/cvs/manual/>;

The MBARI CVS repository may also simply be "viewed" from a web browser for any users who wish to look at its existing source code tree structure at <http://moonjelly.shore.mbari.org/cgi-bin/cvsweb.cgi/>.

[Welcome to CVS!](#) - (New Users **MUST Read and Provide Password for CVS Access!**)

[CVS Graphical User Interface \(GUI\) Client Software Downloads](#) - Windows, Mac, Linux Systems

[Setting up the WinCVS GUI](#)

[Adding code to CVS Repository with WinCVS GUI](#)

[Command Line Access to CVS Repository \(i.e., no GUI\)](#) - For Linux System Users

[View the CVS Repository](#)

Adding code to CVS Repository with WinCVS GUI

This page last changed on Jun 25, 2008 by [salamy](#).

Standard method of operation for working with the CVS GUI clients

Creating a project: Due to the need for proper organization of files and file permissions, ALL projects (*not including subdirectories*) in the CVS repository on moonjelly are to be set-up by the repository administrator (*Karen Salamy: salamy@mbari.org*). Set-up involves creating project groups to restrict unauthorized users from editing or destroying files. New repository projects (modules) are generally set-up at the request of project managers, although any person who wishes to use the repository can make a request for a new project. To have a project (module) created, please e-mail the desired project name and list of users who will need access to the CVS administrator (salamy@mbari.org). Examples of previous projects are *tiburón*, *isi*, *nmc*.

Checking out a module: To check out a module (project or folder) in the repository, click **Create** on main menu then click **Checkout Module....** In the checkout module section, type in the module name you will be working with (no need to indicate /home/cvs). If this is a new project with no previous files in it or if you wish to check out all files in a project, check out the project (e.g., new project name "toolsled" - type in the word *toolsled*). If you desire a particular subdirectory already created in a project enter the following (*projectname/subdirectoryname*), OR, to checkout an individual file from that project, enter (*projectname/subdirectoryname/filename*). The latter will checkout the one file and all listed subdirectories. Be sure to set your home directory before checking files out as the default is to place all files on your computer's hard drive. If this works, you should see the GUI display folders and files checked out. Folders will be in orange, while files will be shown in green. Note: If you plan to work on files in your home directory that you already checked out previously, it is a good idea to first update the copy with the most recent version in the CVS repository. This is important as the file may have been edited by another group user since you last worked on it. To Update a project, folder or file in your home directory, simply click on the directory item and then on the menu bar click **Modify - Update Selection**. You may also click on the user interface button (white rectangle with a downward pointing green arrow). The WinCVS command line will show file and folder updates.

Adding files to a project. In CVS, users edit and create new files in their home directory (a.k.a. a working directory or sandbox). *To set up your home directory see: [Setting up a CVS GUI](#)*
To add a new file to a directory, follow these steps:

Step 1: You must first have a working copy of the directory.

Step 2: Create the new file(s) inside your checked-out working copy of the directory.

Step 3: In WinCVS, use the explore window (usually in left upper corner of screen) to browse to your home directory and view the new file(s) you have added.

Click the new file(s) with your mouse and add the file(s) by either:

1) Clicking on menu bar **Modify - Add Selection. If the file contains binary data click **Modify - Add Selection Binary****

OR

2) Clicking on the user interface button (white rectangle with a red "plus" sign on it). If the file is a binary, click the binary add button which is white with a red "plus" sign and 01 on it.

NOTE: (Holding the shift key down allows the addition of more than one file at a time).

Step 4: Commit the selection. This step actually "Adds" (checks-in) the file(s) to the repository. This can either be done with the file menu or with the user interface button.

1) Clicking on menu bar **Modify - Commit Selection**

OR

2) Clicking on the user interface button (white rectangle with a red arrow pointing left on it).

NOTE: (Holding the shift key down allows the addition of more than one file at a time).

You can also add new directories to a project. However, you must first add the directory in a separate step before adding any files to it. Unlike other CVS commands, the **add** options are not recursive.

Removing files from a project. To remove a file, but remain able to retrieve old revisions follow these steps:

Step 1: Make sure that you have not made any uncommitted modifications to the file. This can be done by using the **diff** command. You can also use the **status** or **update** commands. If you remove the file without committing your changes, you will not be able to retrieve the file as it was immediately before you deleted it.

Step 2: Remove the file from your working copy of the directory. This can be done by first highlighting the file and then clicking the main menu item **Modify - Remove Selection**.

Step 3: Indicate to CVS that you want to delete the file. This can be done in WinCVS by first highlighting the file and then clicking the main menu item **Modify - Remove Selection**. CVS should then highlight the file in red.

Step 4: Remove the file from the repository. Click the commit GUI button or click on menu bar **Modify - Commit Selection**.

When you commit the removal of the file, CVS records the fact that the file no longer exists. It is possible for a file to exist on only some branches and not on others, or to re-add another file with the same name later. CVS will correctly create or not create the file, based on the revision number or date options specified during **checkout** or **update**.

NOTE: When you remove a file, it is removed only on the branch that you are working on. You can later merge the removals to another branch if you want.

You can also remove directories from a project. The concept of removing directories is similar to removing files - you want the directory to not exist in your current working directories, but you also want to be able to retrieve old releases in which the directory existed. The way that you remove a directory is to remove all files in it. You do not remove the directory itself; there is no way to do that. Instead you specify the "prune empty directories" option to and update or checkout. This will cause CVS to remove empty directories from working directories. Probably the best way to do this is to always specify "prune empty directories". If you want an empty directory then put a dummy file (e.g., '.keepme') in it to prevent the "prune empty directories" option from removing it.

It is best to become familiar with the WinCVS GUI and its checkout options before working in the repository. WinCVS has a "Help" menu that is useful. For more in-depth information, go to the CVS main web page <http://www.cvshome.org/> and download the CVS manual.

CVS Client Software Downloads

This page last changed on May 17, 2012 by [salamy](#).

There are two recommended CVS Clients to access the MBARI CVS Respository including: **WinCVS** and **SmartCVS**. Both clients work for all major operating systems including Windows, Macintosh, and Linux systems. The key bits of information to use when setting up either client are the following:

- Your access method needs to be set to **pserver** (SmartCVS) or **passwd file on server** (WinCVS)
- User (Your MBARI user/e-mail account, e.g., *salamy*)
- Repository Server (MBARI CVS Repository is hosted on moonjelly.shore.mbari.org)
- CVSROOT ([/home/cvs](#))

Thus, the full CVSROOT path for user *salamy* would be: `salamy@moonjelly.shore.mbari.org:/home/cvs`

CVS Client Installation Files

WinCVS

- [WinCVS-1.2.0.zip](#) - **Windows**
- [MacCVS-3.3b1.dmg](#) - **Macintosh**
- [gCVS-1.0-1.i386.rpm](#) - **Linux (Red Hat)**

SmartCVS

- [SmartCVS-Windows-7.1.6.zip](#) - **Windows** (Requires JRE 1.4.1 or higher)
- [SmartCVS-MacOSX-7.1.6.tar.gz](#) - **Macintosh** (Requires OS X v. 10.4 or higher)
- [SmartCVS-Linux-7.1.6.tar.gz](#) - **Linux**

CVS Client Set-Up Instructions

WinCVS

- [Setting up WinCVS-1.2.0.doc](#) - **Windows** (**NOTE:** Setup same for MacCVS and gCVS as well)

SmartCVS

- [SmartCVS Getting-Started Tutorial](#) - **PDF Document**
-

CVS Administrative Information

- [CVS Admin Notes.doc](#) - **NOTE: Password Access Required**

Setting up a SVN Graphical User Interface (GUI)

This page last changed on Apr 24, 2008 by [salamy](#).

Setting up CVS Graphical User Interface (GUI)

This page last changed on Jun 25, 2008 by [salamy](#).

Setting up WinCVS (v. 1.2) on your Windows 2000/XP (x86) computer to access the MBARI *moonjelly* CVS source code repository.

- 1). To install WinCVS 1.2 on your computer please click the following link: **WinCVS-1.2.0 Download** and download the **WinCVS-1.2.0.zip** folder to your Desktop. Extract all files to your **C:\Program_Files** folder for installation.
- 2). Next, open the WinCVS120 folder and double click on the **Setup.exe** icon. WinCVS should install on your computers' harddrive as **C:\Program_Files\GNU\WinCVS_1.2**.
- 3). Once installed, open WinCVS (**Startmenu - Programs - GNU - WinCVS_1.2 - WinCVS**). The following edits will need to be performed for the GUI to "view" the moonjelly CVS repository.
 - Under **Admin** on the main menu - click **Preferences**. Under the *General* tab enter your ID name and repository information where it says **"ENTER THE CVSROOT"**. An example of this would be (*salamy@moonjelly:/home/cvs*). Users would type this in exactly, replacing only their ID names for "salamy". This sets the path to the moonjelly repository (/home/cvs) and logs in the user. **NOTE:** Your ID name is your e-mail name (the part before the @mbari.org).
 - Still under the *General* tab, set the **"AUTHENTICATION:"** setting to - **"passwd" file on the cvs server**.
 - Finally, click the *WinCVS* tab and set your **"home"** directory (aka. working directory or sandbox). This directory is where you will download repository files to make edits or add files. It is suggested to place this directory on the MBARI network (e.g., Tornado, Tempest) where it will be backed up nightly. Also, indicate the editor that you would like to view repository files with (e.g., CodeWarrior, emacs, etc.).
- 4). **NOTE:** To actually use this GUI on the moonjelly repository you will first need an exrypted CVS respository password (see also **"Password Access"** at the following link: <https://oceana.mbari.org/confluence/display/CLT/Welcome+to+CVS%21>)
 - To do this, telnet to moonjelly with your MBARI UNIX account (NIS password) and at the initial prompt type the command **encryptCVSpw**. **This runs a script requesting you to create and verify a password for the CVS repository. It outputs an encrypted version of that password. Please copy and e-mail the ENCRYPTED password (*exactly as listed*) to the CVS administrator (*salamy@mbari.org). You will receive an e-mail verification stating that the password account has been updated. You should then be able to access the repository.**

A good way to maneuver through the repository with your GUI is to have your web browser open to: <http://moonjelly.shore.mbari.org/cgi-bin/cvsweb.cgi/> the same time in order to "view" the module's directory structure. This will greatly simplify your use of the system if you know exactly where the files are to be stored or retrieved.
- 5). Next, click **Admin** on main menu - click **Login**. Type in your requested repository password (normal text - not the scripts' encrypted text!). If you were able to log in properly you should see a line that says *****CVS exited normally with code 0*****. Odd as it seems, this is correct.

Setup to access CVS Repository with Command line instead of GUI (Linux users)

This page last changed on Jun 26, 2008 by [salamy](#).

Command Line Access Setup

The MBARI CVS Repository may be accessed on **Linux systems with Command Line control should you decide to not use a GUI.**

Setup of .bashrc file

To set up the CVSRROOT on your Linux system, first log into your user **/home** account and add the following line to your account's **.bashrc** file.

```
export CVSRROOT=:pserver:your_username@moonjelly.shore.mbari.org:/home/cvs
```

This will allow you to access the CVS Repository without having to set up the CVSRROOT each time (i.e., your CVSRROOT will be renewed each time you log in to your computer).

NOTE!! Remember to re-source your .bashrc file the first time you set this up so that the edit takes effect.

```
[username@computerName]$ source ~/.bashrc
```

Logging in to CVS

After logging into your account, you will need to log in to the CVS Repository. You should only have to do this once in the window you are using during each session.

```
[username@computerName]$ cvs login
```

You should get the following response from the computer:

```
Logging in to :pserver:your_username@moonjelly.shore.mbari.org:2401/home/cvs
CVS password:
```

Enter your CVS password at the prompt. If you have entered the correct password you will be returned to a **[username@computerName]\$** prompt. You are now logged in and can access project modules on the system. If you receive a login error from the system, re-check your password as forgotten passwords is the number one problem experienced by CVS users. If the problem persists, please contact Karen Salamy (the CVS Repository Administrator): salamy@mbari.org.

CVS Command Line Usage

CVS has an **enormous** number of commands and options available. Usage of Command Lines for CVS requires familiarization with the command calls. A common link is provided below to help get started:

[CVS Command Line Reference Guide](http://badgertronics.com/writings/cvs/command-line.html)

Additional Usage Guidance

(from <http://badgertronics.com/writings/cvs/command-line.html>)

The general format of CVS commands is:

% cvs general-options command-name command-specific-options filename

Here's how to get help:

cvs --help-commands	Show all of the commands
cvs --help-options	Shows the "global" options, which apply to all commands. The most useful of these is -n, which doesn't execute anything that will change files stored on the disk, but shows what work it would have done.
cvs -H command	Show the command-specific options.

Task Reference

Action	How to do it.
Sync your tree with the repository. See what files are modified or haven't been checked in	cvs update(sample)
See complete revision history for a file	cvs log filename(sample)
See revision comment for one revision	cvs log -r1.5 filename
See the differences between a file and its last checked in version	cvs diff filename
See the differences between two checked-in revisions of a file	cvs diff -r1.4 -r1.7 filename(sample)
See who modified each line of a file, with what revision	cvs annotate filename(sample) This is handy when you see a line of really bogus code and you want to know who was responsible.
Add a new file to the repository	Create the file cvs add filename cvs commit -m "Initial Revision" filename
Add a new binary file to the repository	Create the file cvs add -kb filename cvs commit -m "Initial Revision" filename A "binary file" won't have CVS keyword expansion performed on it.
Add a new directory to the project	cd to the directory's parent cvs add directoryname
Checking a file's changes into the repository.	cvs commit -m "check-in comment" filename If your check-in comment is more than one line, leave off the -m option and use the editor that is brought up.
Revert a file to its last checked-in revision	Remove the file cvs update filename
Add a whole directory tree as a new project	cd to the project directory cvs import -m "project name" directory "ArsDigita" "initial-development" Note that this does not change directory into a CVS-controlled

	tree. You'll need to remove the tree and check it out.
Check out a whole project	<code>cvs checkout project-name</code>
Label a tree for concurrent development	<code>cd</code> to the parent directory of your tree <code>cvs tag -R -b labelname</code> This will use the currently-checked-out versions of the files as they exist in your directory tree. The <code>-b</code> argument means that this tag is a "branch" tag, meaning that someone can check out a new tree based on this tag and do development. If you omit the <code>-b</code>, the label will be a "sticky" tag, meaning that someone won't be able to check in new changes under that label. See the notes on Labeling and Branching
Check out a tree based on a label	<code>cvs checkout -r labelname projectname</code>If <code>labelname</code> was created as a "branch" tag, then you can check in changes without affecting anyone using the development tip. Otherwise files are considered "sticky" and changes cannot be checked in.
Change a file to be treated as a binary file after it's been added	<code>cvs admin -kb filename</code> A "binary file" won't have CVS keyword expansion performed on it.

Welcome to CVS!

This page last changed on Aug 17, 2017 by [salamy](#).

Welcome to CVS!

CVS is the MBARI Software Engineering source code repository which resides on the Linux server '[moonjelly](#)'. Someone has requested that you be added to a module (*project*) in the CVS repository. As a member of these modules, you will have access privileges to any current project whose **ownership** or **group** you belong to. If you need to be added to another module in the future, please let me know as I am the CVS system administrator for moonjelly (salamy@mbari.org).

Permissions

Permissions for these modules are set-up so that all files in the module are made writable only for members of the specified group. If someone is not part of a particular group, they will not have access to write to any files under that directory. These permissions are set recursively throughout the project for security purposes.

Password Access

Access to the repository requires another (*separate*) password. **NOTE: To allow you access to the repository, I will need you to supply a cvs repository password.**

To do this, access the MBARI network (*usually with the command prompt, Cygwin Window, or a Linux Terminal*) by logging in to moonjelly via **ssh** with your standard MBARI password (example: `ssh salamy@moonjelly.shore.mbari.org`). This will run the script: **encryptCVSpw**

Follow the onscreen instructions to create an encrypted version of your password. You may put any password in that you wish - even the same one as your standard MBARI network password. That is up to you. The system will then email you a copy of your encrypted password. **Please forward that e-mail to me so that I may add/update your password to the repository password list. I will then send you an e-mail when you are able to access the repository.**

NOTE: *moonjelly* is on the NIS server at MBARI so you should be able to access the system through telnet via your main MBARI password. If you cannot access *moonjelly* for any reason, please let me know.

CVS GUIs

GUIs are the main way users interact with the repository to checkout, edit, and checkin files. I highly recommend the "Free" [SmartCVS Client](#) version to access moonjelly. However, there are many other free clients including WinCVS (windows-based), gcvs (Linux-based), MacCVS (Mac-based) should you prefer them. You may also use the command line if you do not wish to use a GUI. I have written up a basic "getting started" document for repository users. At the moment, however, it outlines installing and working with WinCVS only since Windows is the main OS used at MBARI. I am not sure if you plan to use a GUI or command line. If you wish to try out WinCVS, please review the link [Setting up WinCVS 1.2](#). You may also reference the guide should you wish to use the Linux version of the GUI, gcvs, as the menu structure is very similar for both GUIs.

The gcvs rpm can be downloaded for most RedHat systems from here: <http://rpm.pbone.net/index.php3/stat/4/idpl/1125012/com/gcvs-1.0-1.i386.rpm.html>

For the most in-depth cvs information, please see the CVS manual at: <http://www.cvshome.org/docs/manual/>.

You can also simply "view" the current repository tree structure at: <http://moonjelly.shore.mbari.org/cgi-bin/cvsweb.cgi/>.

This will allow you to simply read any of the project modules. Let me know how it works and if you have any problems. Please remember to send me your encrypted password as soon as possible.

Regards, - Karen A. Sality (*MBARI CVS administrator*) - sality@mbari.org

Deployment Information

This page last changed on Nov 08, 2007 by [kgomes](#).

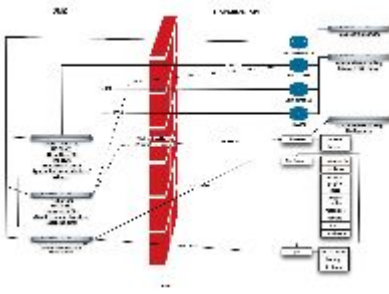
Here are some notes about the deployment of the Alfresco server at MBARI:

On November 7, 2007 Todd, Kent and Kevin had a teleconference with Joe Van De Graaff and Jared from Alfresco. We had some questions about things and wanted to get a good feel for how Alfresco should be setup for the production environment. Joe first talked about the current situation with Alfresco licensing and where Alfresco sits with their non-profit status. Basically, Alfresco has changed from 50% discount to a 25% discount and they are not interested in name branding or references any more (more popular now). There are basically two version of Alfresco: Community and Enterprise. Community is the version we are using now and Enterprise is the only supported version. So the bottom line is that if we want any kind of support, we have to install Enterprise and to install Enterprise, we need a support contract. There is no way to install an unsupported Enterprise version. We then turned the discussion to Jared to talk about the technical aspects. I am not supremely confident that Jared has done any high visibility installations, so we may have to take a step of faith on some of his answers:

1. **Do you have any general recommendations on server?** Yes, see http://wiki.alfresco.com/wiki/Repository_Hardware. In general it looks like their configuration pages have improved see http://wiki.alfresco.com/wiki/Server_Configuration.
2. **What database is best (knowing that we prefer SQL Server)?** He said SQL Server is one of the four main supported servers (MySQL, SQL Server, PostgreSQL, and Oracle).
3. **Should the database be colocated with Alfresco install?** Should not matter unless you are going over REALLY long distances. Even if you install in same machine as Alfresco, there is no way to do direct communication, it will still go over TCP/IP stack.
4. **Why are we having problems with CIFS on Mac?** Hard to tell, but probably related to authentication. We should use passthru (we are). We asked about being on a LINUX box operating in a Windows Domain. He thought that might be related and running it on a Windows box might help.
5. **Does Alfresco support Kerberos authentication?** Yes. I can verify that they have documentation and examples of Kerberos setup, but I have not tried it. Todd was curious about this as this is where MBARI is probably going for single-sign-on authentication.
6. **How do companies backup/restore Alfresco?** His feeling was that the big companies are using Alfresco's export mechanism. This seems implausible to me as there is **huge** amount of overhead associated with exporting. He said that you could export only the rules and backup the content through CIFS. Again, this seems silly to me. The view through CIFS is not the file system level view of the files. Alfresco changes the files to manage them and then makes them look like native files when you view them through CIFS. By backing up through CIFS, you would have to restore through the Alfresco server and I am not sure that is even really possible. The export option seems better than that. It sound to me like the best way to backup Alfresco is to have the file storage on a NetAppliance and coordinate database backups with the NetAppliance backups. This would mean that if the server crashed, you could restore the filesystem and the DB to the correct state and point the new server to them. This should work, but timing is key. We could possibly have some Alfresco downtime when the backups occur, or just do the backups during a time when we feel like there would be little chance of changes occurring during backup. Not really an optimal situation.
7. **How does failover work?** There are two options: cold and hot. Hot is bascially clustering (you point both servers to the same DB and file location and the clustering keeps the local caches in synch), but then you have to pay for the CPU license for the hot backup machine. The other option (cold) is to have an identically configured server sitting dark, bug configured exactly the same way as the production server. This would allow you to make a DNS change and then fire up the cold server if you needed to do maintenance on the production server. The cold option means you don't need a separate license.

So, I **think** the bottom line is that we will want to have a dedicated Alfresco server, 1 CPU (quad core would be nice and maybe buy a machine with two sockets in case we want to upgrade later). Install Windows, point to a SQL Server instance that I.S. is maintaining, map a Tornado drive on the server and use that for the locale storage. Have an identical server configured but sitting dark. It will take some work, but setup a backup regime that will backup the NetAppliance share and the DB at the same time (this will need to be figured out).

Here is a diagram of what this might look like:



If this is where we want to go, the cheapest way might be the following:

1. Purchase a less potent server to use as the cold failover (maybe \$4-5K)
2. Purchase Gold SLA contract with Alfresco at 25% discount (\$11,250)
3. Configure cold server as alfresco.mbari.org, export content from oceana.shore.mbari.org and import to cold failover.
4. See if engineering (or I.S.?) has a server that is not being utilized that we could put in the DMZ and migrate the Confluence and JIRA applications to that server (call this server oceana.shore.mbari.org.
5. Reconfigure oceana.shore.mbari.org in the same way the cold failover was configured. Point to the existing DB and file stores, bring down the cold failover, make a DNS switch to point to alfresco.mbari.org. No export/import should be necessary.

So assuming we could find a server to host Confluence and JIRA, the bottom line cost would be around \$15-16K with annual support contract of around \$12K. At some later time, we would probably want to purchase a test server and license (total of about \$8,000, with an additional \$2,800 annual contract for the test installation).

Installation Notes

This page last changed on Nov 03, 2007 by [kgomes](#).

I had already setup Alfresco 1.3 before writing up these installation instructions. 1.3 had some issues and so I upgraded to 2.0 and included those installation instructions here. I first setup a test installation on the machine prey to make sure it could do all we wanted it to. I originally wrote these instructions for the install on prey and then updated them as I installed on oceana (production server).

In order to install Alfresco, I needed Java and a database installed. On the original installation, I used a database on Solstice (MSSQL) as this was our institutional database server. One of Alfresco's big draw back was its performance and I suspect it might have had something to do with a remote database. For that reason, I installed Alfresco so it was a completely local installation using MySQL as the database server. Prey is a Itanium II server running RedHat Enterprise 4. Oceana is an AMD Opteron box running RHE4.



Test Machine Installation Difference

I also setup a test machine to help Alan Lash have something to develop against and for me to test somethings like using MS SQL Server instead of my SQL. I will write notes where the instructions differ. I setup the test machine on ssdsdevpc.shore.mbari.org which is an HP desktop machine running Fedora Core 7.

1. I downloaded and installed BEA's JRockit JVM to use for this installation. (J2SE 5)
2. I downloaded Alfresco 2.0 (tomcat version) from [alfresco.org](#) (the tar.gz file and moved it to /opt)



Test Machine Installation Difference

On the test machine, I installed Alfresco 2.1 tomcat version

3. MySQL was already installed in RHE4 so I was good there (MySQL version 5.0.27). I simply turned on the mysql service.



Test Machine Installation Difference

I.S. was hoping to get Alfresco to run against the institutes MS SQL Server for support reasons so on the test machine, I used a database named 'Alfresco' on our database server [solstice.shore.mbari.org](#). Neil created the database for me and the user DB_Collaboration was used to access it.

4. At first there is no root password on mysql, so I changed it using the 'mysql' utility like so (Note the password was removed here.):

```
[root@prey ~]# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5 to server version: 5.0.27-standard
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

```
Database changed
mysql> UPDATE user SET password=PASSWORD('*****') WHERE User='root' and Host='localhost';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql>
```

5. Now I enabled root access from my desktop so I could use the MySQL Admin GUI like so:

```
mysql> GRANT USAGE ON *.* TO 'root'@'skagit.shore.mbari.org' IDENTIFIED BY '*****';
Query OK, 0 rows affected (0.00 sec)
```

mysql>

6. BTW, PHP 5.1.4 was already installed (this is optional for Alfresco)



Test Machine Installation Difference

Fedora core 7 had 5.2.2 installed

7. I cd'd to the /opt directory and gunzipped the alfreco tar.gz file
8. I created a directory for alfresco as the tar.gz file does not have a base directory. I created a directory called alfresco-2.0



Test Machine Installation Difference

I named the directory alfresco-2.1

9. I then moved the tar into the alfresco-2.0 directory



Test Machine Installation Difference

alfresco-2.1 directory

10. I cd'd into that directory and untarred it.
11. In order to setup the DB on the mysql server, I cd'd to the /opt/alfresco-2.0/extras/databases/mysql directory and started up 'mysql -u root -p' and ran

```
mysql> source db_setup.sql
```

```
Query OK, 1 row affected (0.01 sec)
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
Query OK, 0 rows affected (0.00 sec)
```

mysql>

12. The database is now created and the appropriate permissions are given to a user name 'alfresco' with password 'alfresco'. That should be fine for now.
13. I did the optional OpenOffice install by visiting openoffice.org and downloading version 2.2 for linux.
 - a. This may not always be the case, but I had a previous install of OpenOffice which I wanted to remove first
 - i. I first ran 'rpm -qa | grep openoffice > rmlist' which gave me a list of all rpms with openoffice in them.
 - ii. I looked over the list to make sure I really wanted to remove everything in that list (I did) and then ran 'rpm -e `cat rmlist`'
 - b. I kept the download in my home directory as this will be an RPM based install
 - c. I unzipped the download
 - d. I untarred the download (this created an OOF680_m14_native_packed-1_en-US.9134 directory)
 - e. I cd'd into the new directory and then into the RPMS directory
 - f. I already had java installed, so I moved the jre-6-linux-i596.rpm out of this directory
 - g. I then installed the OpenOffice core by running 'rpm -Uvh --prefix /opt/openoffice-2.2/ *.rpm'
 - h. It said I was missing two dependencies (libgnomevfs and libgconf), but they look like they are located in /usr/lib64. Since I think I have them, I ran the previous command with --nodeps appended to skip dependency checking
 - i. I then cd'd into the desktop-integration directory and executed
 - i. rpm -Uvh openoffice.org-redhat-menus-2.2-9119.noarch.rpm
 - j. Now run /opt/openoffice-2.2/program/soffice to get the GUI interface which will allow you to register soffice (run this as the user that will be running alfresco/soffice)
 - k. Exit OpenOffice
 - l. Run OpenOffice in headless mode to make sure it runs. Before you run OpenOffice in headless mode you may have to actually install Xvfb so you can run from the command line (I leave that to you to figure out).
 - i. /usr/bin/Xvfb -fp /usr/share/X11/fonts/misc :1 -screen 0 800x600x16 &

- ii. `/opt/openoffice-2.2/program/soffice "-accept=socket,host=localhost,port=8100;urp;StarOffice.ServiceManager" -nologo -headless -display :1`
- m. Cntl-C out of soffice (assuming it ran)
- n. Now cd to `/opt/alfresco-2.0` and rename the `zstart_oo.sh` to `start_oo.sh` and edit so that it looks like the following:

```
#!/bin/sh
# -----
# Start script for the OpenOffice transform service
# -----
export DISPLAY=:1
echo "Starting Xvfb..."
`/usr/bin/Xvfb -fp /usr/share/X11/fonts/misc :1 -screen 0 800x600x16` &

echo "Starting OpenOffice service..."

# Comment or uncomment the appropriate location using #
# Assumes OpenOffice is installed in /opt
/opt/openoffice-2.2/program/soffice "-accept=socket,host=localhost,port=8100;urp;StarOffice.ServiceManager" -nologo -headless -nofirststartwizard -display :1 &
```



Test Machine Installation Difference

This was a bit different in FC7. In order to get Xvfb installed I used "Add/Remove Software" to install it. OpenOffice was already installed so I just had to find the soffice command location. The command to get Xvfb running was also a bit different. I had to create a ascii file called `/etc/xvfb-auth.cfg` that only had one line in it that read 'localhost'. Then the command to start Xvfb was 'Xvfb :1 -auth /etc/xvfb-auth.cfg'. Then I could start soffice the same way. I changed the `start_oo.sh` file to reflect those commands.

14. In order to make sure Alfresco was using the MySQL installation, I removed the following files from the `/opt/alfresco-2.0/tomcat/shared/classes/alfresco/extension` directory:

```
custom-db-and-data-context.xml
custom-db-connection.properties
custom-hibernate-dialect.properties
```



Test Machine Installation Difference

Did not do this

15. Now before running, I want to make sure the configuration points to the right location where the data will be stored and the database connections.
 - a. I created a directory `/data/alfresco-2.0` to house the data



Test Machine Installation Difference

`/data/alfresco-2.1`

- b. I cd'd to `/opt/alfresco-2.0/tomcat/shared/classes/alfresco/extensions`
 - c. I copied the `custom-repository-context.xml.sample` to `custom-repository-context.xml`



Test Machine Installation Difference

This file already existed and was no different from the sample, so I skipped this step

- d. I copied the `custom-repository.properties.sample` to `custom-repository.properties`
 - e. I edited the `custom-repository.properties` file:



Test Machine Installation Difference

This file already existed and so copied the original to custom-repository.properties.orig and edited the existing one.

- i. Set the dir.root=/data/alfresco-2.0/data
- ii. Uncommented the db.* properties and left as is (that is the way I set them up)



Test Machine Installation Difference

I changed them to work with the MS SQL Server login so they looked like:

```
db.username=DB_Collaboration
db.password=*****
db.pool.initial=10
db.pool.max=100
```

- iii. I commented out the two HSQL connection lines
- iv. Uncommented the two MySQL connection lines and changed the driver to 'com.mysql.jdbc.Driver'



Test Machine Installation Difference

Left the MySQL lines commented out, but uncommented the SQLServer lines and changed them to

```
db.driver=net.sourceforge.jtds.jdbc.Driver
db.url=jdbc:jtds:sqlserver://solstice.shore.mbari.org/
Alfresco;domain=SHORE;user=DB_Collaboration;password=*****
```

- v. I also added the following lines at the end of the file to connect up to the external mail server

```
#
# The external email server properties
#
mail.host=mail.shore.mbari.org
mail.port=25
mail.username=
mail.password=
```

- vi. Saved the file
- f. I copied custom-hibernate-dialect.properties.sample to custom-hibernate-dialect.properties



Test Machine Installation Difference

Already existed, so backed up to .orig and edited to comment out the HSQL and uncomment the SQLServer one.

- g. I edited the custom-hibernate-dialect.properties file and uncommented the MySQL Dialect line



Test Machine Installation Difference

The downloaded alfresco already has the MySQL driver jar in the install, but since I am using SQLServer, I downloaded the jtds driver from the internet and installed the jtds-X.X.X.jar file in the /opt/alfresco-2.1/tomcat/common/lib to make it available to alfresco.

- 16. I then cd'd back to the /opt/alfresco-2.0 directory and edited the alfresco.sh script and change the APPSERVER to point to the correct directory of /opt/alfresco-2.0. I also added the following line after the line that stated 'killall soffice.bin':
killall Xvfb. That ensured the Xvfb process was killed when alfresco was shutdown.



Test Machine Installation Difference

Set it to /opt/alfresco-2.1/tomcat for the test machine



In the tomcat version of alfresco, the actual alfresco deployment is in a .war file in the tomcat/webapps directory. If the server has been started, the .war gets expanded into a directory and you could make changes to the files there, but if the war file was ever "touched" again, it would redeploy those file and overwrite your changes. That is why it is best to make the changes in the .war file so they are permanent



Test Machine Installation Difference

For the test machine, I wasn't too worried about details, so I skipped the next 15 or so steps (picks up again at configuring for MBARI LDAP below.

17. Copy the /opt/alfresco-2.0/tomcat/webapps/alfresco.war file to your home directory
18. cd to your home directory and make another copy so you will have the original (something like alfresco.war.orig)
19. Now I created a new directory called 'scratch' and moved the war there so that it was a clean directory for expansion. I moved the alfresco.war file to that directory, cd'd to it and then ran 'jar -xvf alfresco.war'. This expands the contents of the war file (like tar).
20. I also wanted to change the "Company Home" that seems to be everywhere.
 - a. Edited ./WEB-INF/classes/alfresco/messages/webclient.properties and changed company_home from 'Company Home' to 'MBARI'
 - b. Edited ./WEB-INF/classes/alfresco/messages/webclient_en_US.properties and changed company_home from 'Company Home' to 'MBARI'
 - c. I edited ./WEB-INF/classes/alfresco/messages/bootstrap-spaces.properties and changed the first two lines to read

```
spaces.company_home.name=MBARI
spaces.company_home.description=The root space for MBARI
```
 - d. Edited ./WEB-INF/classes/alfresco/messages/bootstrap-spaces_en_US.properties and made the same changes as the previous step



This took a bit to find all these, and to be honest, I am not sure which one actually made the change on the GUI, but I suspect it was the _en_US ones.

21. When users are imported via LDAP, user spaces get created automatically. In order to put them where I wanted them to go (instead of right in the Company Home), I edited the ./WEB-INF/classes/alfresco/authentication-services-context.xml file. In the file there is a bean with an name of personalHomeFolderProvider. There is a property name 'path' and I added

```
/${spaces.user_homes.childname}
```

to the end to try to get them to create the user spaces there instead of in the Company Home.

22. I wanted to change the logo that is used. I attached the replacement logo [here](#). Copy this to ./images/logo and overwrite the one that is there.
23. In order to setup SSL for the application (will do that later), edit ./WEB-INF/web.xml and add the following

```
<!-- Added this security constraint to allow for SSL protection -->
<security-constraint>
<web-resource-collection>
<web-resource-name>Login Pages</web-resource-name>
<url-pattern>/*</url-pattern>
</web-resource-collection>
<user-data-constraint>
<transport-guarantee>CONFIDENTIAL</transport-guarantee>
</user-data-constraint>
</security-constraint>
```

24. In order to configure CIFS server, I edited the file ./WEB-INF/classes/alfresco/file-servers.xml
 - a. I added 'domain="SHORE"' to the <host ...> tag.
 - b. I made sure the domain broadcast submask was 255.255.255.255
 - c. I uncommented the WINS section and changed it to

```
<WINS>
```



```
<primary>134.89.10.32</primary>
<secondary>134.89.12.87</secondary>
</WINS>
```

- d. I deleted the enterprise authenticator and replaced it with this:

```
<config evaluator="string-compare" condition="Filesystem Security">
<authenticator type="passthru">
<Server>blizzard.shore.mbari.org</Server>
<Domain>shore</Domain>
</authenticator>
</config>
```

25. OK, this is a total HACK and needed to be done due to the wonderful world of open source. In Internet Explorer with SSL turned on, you would get a dialog pop-up that stated the secure server was trying to serve non-secure content. This was due to the Alfresco Community logo being served from <http://www.alfresco.com>. Only IE did this and you can work around it on the client end by doing the following:
- Startup IE.
 - Go to Tools->Internet Options ...
 - Click on Security tab
 - Click on "Trusted Sites" zone
 - Click on Sites button
 - Make sure check box for "Require Server verification (https 🤪 for all sites in this zone)" is unchecked 7. Type in "http://www.alfresco.com" into the "Add this website to the zone:" text field and click Add.
 - Click on close
 - Make sure "Trusted Sites" is still selected.
 - Click on "Custom Level..." under Security level for this zone.
 - Scroll down till you find the option that says "Display Mixed Content" and select "Enable"
 - Click on OK
 - Click on the "Local Intranet" zone.
 - Click on "Custom Level..." under Security level for this zone.
 - Scroll down till you find the option that says "Display Mixed Content" and select "Enable"
 - Click on OK
 - Click on OK to close Internet Options dialog.
26. Now there are a couple of reasons why this is bad, but the main one is that if you are trying to reach this from outside the firewall, you will have to change those settings for the Internet Zone (I think) which is bad. Plus every user will have to do this. To get around this, I had to:
- Take [this](#) logo image and save it to ./images/logo directory.
 - This next step can either be very easy or very complicated. You can try downloading [this](#) jar file and copying it to ./WEB-INF/lib. If when you start up Alfresco you get some exceptions that look like they are related to a class called PageTag, you will have to do it the hard way:
 - Now you have to actually check out the subversion source code from Alfresco. I am assuming you know how to do that. By the way, you will need ant and java installed to do this. The SVN URL for the repository is `svn://www.alfresco.org/alfresco`.
 - Once it is checked out find the PageTag.java file (should be in `HEAD/root/projects/web-client/source/java/org/alfresco/web/ui/repo/tag` directory) and edit:
 - Change the line


```
private final static String ALF_LOGO = "http://www.alfresco.com/images/alfresco_community_horiz20.gif";
```

 to:


```
private final static String ALF_LOGO = "/alfresco/images/logo/alfresco_community_horiz20.gif";
```
 - Save the file.



If you run this next step on the machine where the application is installed and have a TOMCAT_HOME environment variable defined, it will try to deploy everything to your Tomcat installation. The best thing to do is build this on a separate machine.

- iv. Open a command prompt in the HEAD/root directory and type "ant build-tomcat". After a long wait, the build will actually fail if you don't have a TOMCAT_HOME defined in your environment.
- v. Once the build has failed you will need to create a new jar file called a0.jar with only the PageTag.class file in it.
- vi. Copy the a0.jar file to the ./WEB-INF/lib directory.
- 27. That takes care of all the changes to the WAR file. We can now put it back into .war form.
- 28. You should still have a alfresco.war file in the directory where you ran the jar -xvf command. Delete or move that file out as you will be recreating it.
- 29. While you are in the directory you expanded the war file in, run
 - a. jar cvfm alfresco.war META-INF/MANIFEST.MF .
- 30. You should now have an alfresco.war file in the directory. Move that back to the alfresco install and overwrite the existing one.
 - a. mv alfresco.war /opt/alfresco-2.0/tomcat/webapps
- 31. Most users like the details view, so I set the default to that by editing /opt/alfresco.2.0/tomcat/shared/classes/alfresco/extensions/web-client-config-custom.xml and added:

```
<!-- MBARI Custom: Added this to change the default view to details -->
<config evaluator="string-compare" condition="Views">
<!-- the views available in the client -->
<views>
<!-- default values for the views available in the client -->
<view-defaults>
<browse>
<!-- allowable values: list|details|icons -->
<view>details</view>
<page-size>
<list>20</list>
<details>20</details>
<icons>20</icons>
</page-size>
</browse>
</view-defaults>
</views>
</config>
```

- 32. Now in order to use MBARI accounts, I needed to configure LDAP.



In the older version of Alfresco, I actually used NTLM to manage logins, but currently 2.0's NTLM for ActiveDirectory is broken. The supposedly have fixed it in the Enterprise version, but it has not made it to the Community edition yet. That is OK, as I will just use LDAP to authenticate.

- a. I cd'd to the /opt/alfresco-2.0/tomcat/shared/classes/alfresco/extensions directory
- b. I then copied the ldap-authentication-context.xml.sample to ldap-authentication-context.xml and edited to match our ActiveDirectory setup. I have attached the file to this WIKI pages instead of detailing out all the changes. The file is [here](#)
- c. We wanted to be able to make sure we could have collaborator accounts that were not MBARI LDAP accounts so that we could have Alfresco users that had not permissions at all on the MBARI network. In order to do this, I used the Alfresco native user service and chained the authentication together with the MBARI LDAP. In order to do this, I copied the chaining-authentication-context.xml.sample to chaining-authentication-context.xml. I then edited the file to make the appropriate changes (I have also attached the changed config file [here](#) for downloading):
 - i. The first change was to the first bean declaration's id from **authenticationServiceImpl** to **authenticationService**.
 - ii. In the same bean declaration, I changed the first authentication services reference from **authenticationServiceImplJAAS** to **authenticationServiceImplLDAP**
 - iii. I then changed the JAAS bean declaration to point to the ldap declaration by changing the bean id from **authenticationServiceImplJAAS** to **authenticationServiceImplLDAP**
 - iv. There are two properties in this bean and I had to change them to point to the LDAP configuration (these changes actually reference the services defined in the ldap-authentication-context.xml configuration file).
 - i. Change the authenticationDao property's value from **authenticationDaoJAAS** to **authenticationDaoLDAP**

- ii. Change the authenticationComponent property from **authenticationComponentImplJAAS** to **authenticationComponentImplLDAP**
- d. In order to grant certain accounts admin privs, I created a new custom-authority-services-context.xml in the same directory as the ldap-authentication-context.xml file and added

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" 'http://www.springframework.org/dtd/spring-beans.dtd'>
<beans>
<bean id="authorityService"
class="org.alfresco.repo.security.authority.AuthorityServiceImpl">
<property name="authenticationComponent">
<ref bean="authenticationComponent" />
</property>
<property name="personService">
<ref bean="personService" />
</property>
<property name="nodeService">
<ref bean="nodeService" />
</property>
<property name="authorityDAO">
<ref bean="authorityDAO" />
</property>
<property name="permissionServiceSPI">
<ref bean="permissionServiceImpl" />
</property>
<property name="adminUsers">
<set>
<value>kgomes</value>
<value>admin</value>
<value>administrator</value>
</set>
</property>
</bean>
</beans>
```

- e. In the same directory, I edited the file-servers-custom.xml file and added a new config section

```
<!-- Allow guest access to file systems -->
<config evaluator="string-compare" condition="Filesystem Security">

<authenticator type="passthru">
<Server>blizzard.shore.mbari.org</Server>
<Domain>shore</Domain>
</authenticator>

</config>
```

33. Now, it is time to setup the server so we can login over SSL.

Test Machine Installation Difference

Did not setup SSL for the test machine

- a. In a shell run the keytool and answer the questions as shown (note the first and last name must be the fully qualified host name where alfresco will be run)

```
kgomes$ sudo keytool -genkey -alias tomcat -keyalg RSA
Enter keystore password: alfresco
What is your first and last name?
[Unknown]: skana.shore.mbari.org
What is the name of your organizational unit?
[Unknown]: MBARI
What is the name of your organization?
[Unknown]: Monterey Bay Aquarium Research Institute
What is the name of your City or Locality?
[Unknown]: Moss Landing
What is the name of your State or Province?
```

```
[Unknown]: CA
What is the two-letter country code for this unit?
[Unknown]: US
Is CN=mbari.org, OU=MBARI, O=Monterey Bay Aquarium Research Institute, L=Moss Landing,
ST=CA, C=US correct?
[no]: yes
```

Enter key password for <tomcat>
(RETURN if same as keystore password):

This will generate a .keystore file in your home directory.

- b. Now edit the /opt/alfresco.2.0/tomcat/conf/server.xml. Locate the <Connector> element with port of 8443 and uncomment that section. Also add the keystorePass="alfresco" phrase. It should look like

```
<!-- Define a SSL HTTP/1.1 Connector on port 8443 -->
<Connector port="8443" maxHttpHeaderSize="8192"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" disableUploadTimeout="true"
acceptCount="100" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS" keystorePass="alfresco"/>
```

34. I then started alfresco with './alfresco.sh start'

 This takes a couple minutes as it has to first build the DB and all the indexes and such.

 Wait 5 minutes

After you start up Alfresco with LDAP turned on, it takes 5 minutes before it will import users and groups from Active Directory, so go get a cup of coffee.

35. Then I can go to the web interface at <http://oceana.shore.mbari.org:8080/alfresco> and login as your MBARI username.
36. TODO: Make sure startup/shutdown script in /etc/init.d is working correctly
37. I.S. then setup backup regime to do incremental backups nightly and full backups weekly
38. TODO: Verify I know where the MySQL files are and that we can restore them correctly. Also turn on binary logs and use a test machine to learn (and document here) how to restore from backup.

 IMPORTANT

For some strange reason, the Alfresco stack would seem to crash after 20-30 days of operation with some very cryptic errors. I found some entries in the forums about changing the ulimit value in linux. So I issued a 'ulimit -n 4096' and then restart Alfresco. I haven't seen that problem since.

Java Signing at MBARI

This page last changed on Dec 04, 2007 by [brian](#).

About Jar Signing

If you intend to distribute an application using Java Web Start you will need to digitally sign the jar using a certificate that verifies the jars origin. MBARI provides a security certificate from [Verisign](#) that you can use for signing your Java applications.

Keystore Details

Parameter	value
Keystore Location	http://seaspray.shore.mbari.org/keystore/keystore.bin
Keystore Alias	MBARIjava
Keystore Password	java1700
Key Password	javaBeansMakeLousyCoffee

Ant Example

Here's an example of how to sign your jars using [Ant](#)

```
<target name="sign" description="Sign jar files" >
<property name="lib" value="lib" />
<property name="keystore.alias" value="MBARIjava" />
<property name="keystore.passwd" value="java1700" />
<property name="key.password" value="javaBeansMakeLousyCoffee" />
<property name="keystore" value="http://seaspray.shore.mbari.org/keystore/keystore.bin" />
<signjar alias="${keystore.alias}" storepass="${keystore.passwd}" keystore="${keystore}"
keypass="${key.password}">
<fileset dir="${lib}">
<include name="**/*.jar" />
</fileset>
</signjar>
</target>
```

MBARI Acronym Page

This page last changed on Jan 14, 2015 by [carueda](#).

1. AAI: Association for the advancement of artificial intelligence
2. ACT: Alliance for Coastal Technology. A NOAA-funded partnership of research institutions, resource managers, and private sector companies dedicated to fostering the development and adoption of effective and reliable sensors and platforms. <http://www.act-us.info/>
3. ADCP: Acoustic Doppler Current Profiler - a sonar that attempts to produce a record of water current velocities for a range of depths http://en.wikipedia.org/wiki/Acoustic_Doppler_Current_Profiler
4. AEITS: Autonomous Eye-in-the Sea - stand-alone deepwater video observatory for unobtrusive monitoring (not connected to a cable - gathers data autonomously)
5. AGU: American Geophysical Union
6. AIV: Analytical Instrument Valve (a valve on the MFB)
7. Alfresco: not an acronym. Open source content management system. <http://www.alfresco.com>
<http://alfresco.mbari.org>
8. ALOHA-MARS: <http://alohamooring.apl.washington.edu/> ALOHA (A Long-term Oligotrophic Habitat Assessment) is a place off the coast of Hawaii where HOT (Hawaiian Ocean Timeseries) Station 2 data is taken, at 22 45'N, 158W, located about 100 km north of Oahu, Hawaii. Station ALOHA has been the location of ~monthly shipboard observations of water column physics, biogeochemistry, and ecosystems for over 18 years. Nearby moorings have also been deployed (e.g. HALE-ALOHA). The NSF-funded cabled observatory deployed at that site is called the ALOHA observatory. MARS is the Monterey Advanced Research System (see below). The idea is to deploy a mooring in Monterey Bay connected to the MARS observatory, and in the future another one at the ALOHA site connected to the ALOHA observatory. The team hopes to deploy subsequent similar nodes at other observatory sites (H2O, Neptune, DEOS sites, etc.). <http://seahurst.apl.washington.edu/mars.php>
9. AMC: Autonomous Mission Controller
10. AMQP: Advanced Message Queuing Protocol - <http://www.amqp.org>
11. AOA: Angle of Attack
12. APL: Applied Physics Laboratory, University of Washington
13. ARI-WG: Access to Remote Instrumentation in a distributed environment - Working Group. OpenGrid Forum http://www.ogf.org/gf/group_info/view.php?group=ari-wg
14. ASLO: American Society of Limnology and Oceanography
15. ASTEP: Astrobiology Science and Technology for Exploring Planets
16. ATS: Autonomous Tissue Sampler
17. AVED: Automated Visual Event Detection <http://www.mbari.org/aved/>
18. AWW: All was well. <http://www.jkrowling.com/en/>
19. Beowulf cluster: scalable performance collection of independent computers based on commodity hardware, on a private system network, with open source software (Linux) infrastructure.
20. BIAUV: Benthic Imaging Autonomous underwater vehicle
21. BIOSA: Biophysical Oceanographic Sensor Array
22. Bogey: not an acronym. A score of one over par on any individual hole.
23. BOM: Bill of Materials
24. BRS/MRS: Benthic Respirometer System / Midwater Respirometer System <http://www.mbari.org/expeditions/Pulse53/gear.htm>
25. Bye: in sports and other competitive activities, most commonly refers to the practice of allowing a player or team to advance to the next round of a playoff tournament without playing
26. Calit2: California Institute for Telecommunications and Information Technology (at UCSD)
27. Caltech: California Institute of Technology <http://en.wikipedia.org/wiki/Caltech>
28. CDR: Critical Design Review ("Hey I'm starting to build something, should I stop?")
29. CI: Cyberinfrastructure <http://en.wikipedia.org/wiki/Cyberinfrastructure>
30. CeNCOOS: Central and Northern California Ocean Observing System <http://www.cencoos.org/>
31. CENIC: The Corporation for Education Network Initiatives in California <http://www.cenic.org/>
32. CFD: Computational Fluid Dynamics
33. CGSN: Coastal & Global Scale Nodes - http://www.whoi.edu/ooi_cgsn/home
34. CMORE: Center for Microbial Oceanography: Research and Education (a funding source)
35. CNAME'd: canonical name, also referred to as a CNAME record, a record in a DNS database that indicates the true, or canonical, host name of a computer that its aliases are associated with. an alias so that DNS can resolve some alias to a specific machine <http://www.webopedia.com/TERM/C/CNAME.htm>
36. COI: Common operating infrastructure (one of seven projects that together implement the Network for Ocean Research, Interaction and Application (NORIA) OOI CI.
37. Coolpix: Not an acronym. A registered trademark of Nikon. Digital cameras sold by Nikon, Inc. Coolpix images are digital images captured by a Coolpix camera. We often use the term Coolpix when we mean digital image captured by a digital still camera.

38. Condor: not an acronym. A specialized workload management system for compute-intensive jobs developed by the University of Wisconsin-Madison <http://www.cs.wisc.edu/condor/>
39. CREON: Coral Reef Environmental Observatory Network <http://www.coralreefeon.org/>
40. Cron: not an acronym. Cron is the name of program that enables UNIX users to execute commands or scripts (groups of commands) automatically at a specified time/date. <http://www.unixgeeks.org/security/newbie/unix/cron-1.html>
41. CTD: Conductivity, Temperature, and Depth <http://www.whoi.edu/instruments/viewInstrument.do?id=1003>
42. CUAHSI: Consortium of Universities for the Advancement of Hydrologic Science, Inc. - <http://www.cuahsi.org/>
43. CVS: Concurrent Versions System; maintains control of history of code/documents
44. DataNet solicitation: NSF 07-601, Sustainable Digital Data Preservation and Access Network Partners (DataNet).
45. DC: Dublin Core vocabulary
46. DEM: Digital Elevation Model
47. D-ESP: Deep ESP; the sphere and all it's accouterments
48. DI: De-ionized, often used with water
49. DMO: Division of Marine Operations
50. DMZ: Demilitarized zone. Computers placed into their own subnetwork (sometimes protected by a back-end firewall) in direct connection with the internet, protected by a front-end firewall. [http://en.wikipedia.org/wiki/Demilitarized_zone_\(computing\)](http://en.wikipedia.org/wiki/Demilitarized_zone_(computing))
51. Doc Ricketts: ROV, replacing Tiburon, carried by R/V Western Flyer.
52. DOEST: Day of Engineering, Science and Technology
53. Dribbling: not an acronym. The act of bouncing the ball continuously
54. DSL: Digital Subscriber Line; used to describe a type of modem
55. DStoNetCDF: SSDS post-processing service
56. DVL: Doppler Velocity Log
57. DWSM: Deep Water Sampling Module
58. EarthCube: EarthCube is a NSF activity to create a data and knowledge management system for the 21st Century. The scientific objective is to develop a framework over the next decade to understand and predict the Earth system from the sun to the center of the Earth <http://earthcube.ning.com/>
59. EE: Electrical Engineering
60. eGY: the Electronic Geophysical Year <http://www.egy.org/>
61. EITS: Eye-in-the Sea - deepwater video observatory for unobtrusive monitoring. <http://www.mbari.org/mars/general/eits.html>
62. EOM: Electro-Optical-Mechanical
63. EPA: Environmental Protection Agency - <http://www.epa.gov/>
64. EPIC: Embedded Platform for Industrial Computing
65. ERA: Earned Run Average
66. ESB: Enterprise Service Bus
67. ESIP Federation: Federation of Earth Science Information Partners
68. ESONET: European Seafloor Observatory Network <http://www.oceanlab.abdn.ac.uk/research/esonet.php>
69. ESP: Environmental Sample Processor - Chris Scholin et al. <http://www.mbari.org/muse/platforms/ESP.htm>
 - a. Darlene was the 1st generation ESP
 - b. GiGi The first 'second generation' ESP instrument
 - c. NEO 'Nasa-Earth-Observatory'; The second 'second generation' ESP instrument
 - d. Mack The third 'second generation' ESP instrument
 - e. Bruce Another 'second generation' ESP instrument
 - f. MOE A second gen. instrument with the MFB connected
 - g. Betty -another one
 - h. Gordon -another one (Betty & Gordon named after the founders of the Moore Foundation, who funded the builds)
70. ESSIN: Earth Sciences Semantic Interoperability Network. A multi-institutional proposal to NSF under the Community-based Data Interoperability Networks (INTEROP) Crosscutting program.
71. Et al.: and others
72. FC5: Fedora Core 5
73. FDR: Final Design Review
74. Ferret: plotting/analysis software package
75. FOCE: Free Ocean CO2 Enrichment
76. FPGA: Field Programmable Gate Array
77. FSM: Finite State Machine
78. FVGR: Free Vehicle Grab Respirometer

79. GAL: Gene Array Layout which is the file generated when printing a microarray for the ESP instrument. It tells you what and where things were printed on the filter paper.
80. GNB: Gaussian Naïve Bayes. GNB is a probabilistic classifier relying on Bayes rule to estimate posterior probabilities of labels given the data. Naive assumption in it is an independence of the features. See http://en.wikipedia.org/wiki/Naive_bayes for more information.
81. GEOS: Global Earth Observation System of Systems <http://www.epa.gov/geos/>
82. GRID Computing: Grid computing (or the use of a computational grid) is the application of several computers to a single problem at the same time - usually to a scientific or technical problem that requires a great number of computer processing cycles or access to large amounts of data. http://en.wikipedia.org/wiki/Grid_computing
83. GRIIDC: Gulf of Mexico Research Initiative Information and Data Cooperative <https://data.gulfresearchinitiative.org/about>
84. GUI: Graphical User Interface
85. GWT: Google Web Toolkit - <http://code.google.com/webtoolkit/>
86. HDF: Hierarchical Data Format
87. Hexakosioihexekontahexaphobia: not an acronym. Fear of the number six-hundred sixty-six.
88. HOBI: Harbor Branch Oceanographic Institution
89. HOTS: Hawaii Ocean Time-Series program
90. HPP: High-Pressure pump
91. ICAN: The International Coastal Atlas Network - <http://ican.science.oregonstate.edu/home>
92. ICD: Interface Control Diagram; block diagram of all system components and their interfaces
93. ICPR: International Conference on Pattern Recognition. This is the annual conference of the International Association for Pattern Recognition
94. IDE: Integrated Development Environment
95. iLab: Professor Laurent Itti's biologically inspired computational modeling group at the University of Southern California. They maintain the iLab Neuromorphic Vision C++ Toolkit that we use in AVED for visual event detection.
96. IMDIS: International Conference on Marine Data and Information Systems sponsored by <http://www.seadatanet.org/>
97. International metadata summit: This summit is formally titled the IODE/JCOMM Forum on Oceanographic Data Management and Exchange Standards. It is a weeklong meeting of a small group (17) of leaders in international oceanographic data and metadata systems, yet is intended to initiate international agreements on metadata standards. See http://www.iode.org/index.php?option=com_content&task=view&id=74 for details.
98. IO: Implementing Organization http://www.orionprogram.org/PDFs/OOI_CG_RFP_16Mar07.pdf
99. IOC: Intergovernmental Oceanographic Commission, <http://ioc.unesco.org/iocweb/index.php>
100. IODE: International Oceanographic Data Exchange, <http://www.iode.org>
101. IODP: Integrated Ocean Drilling Program <http://www.iodp.org>
102. IOOS: Integrated and Sustained Ocean Observing System <http://www.ocean.us>
103. ISUS: In Situ Ultraviolet Spectrophotometer
104. IWCE: International Wireless Communication Expo (<http://www.iwceexpo.com>)
105. JCOMM: Joint WMO-IOC Technical Commission on Oceanography and Marine Meteorology, <http://ioc.unesco.org/iocweb/index.php>
106. JEE: Java Enterprise Edition (formerly known as J2EE)
107. JIRA: A bug tracking, issue tracking, and project management application <http://www.atlassian.com/software/jira/>
108. JOI: Joint Oceanographic Institutions <http://www.joiscience.org/joi>
109. JVM: Java Virtual Machine - <http://en.wikipedia.org/wiki/Jvm>
110. KML: Keyhole Markup Language - A file format used to display geographic data in an Earth browser, such as Google Earth
111. LCM: Lightweight Communications and Marshalling <https://lcm-proj.github.io/>
112. LED: Light-emitting diode
113. LDEO: Lamont-Doherty Earth Observatory, Columbia University
114. LLNL: Lawrence Livermore National Laboratories
115. LRAUV: Long Range AUV (Autonomous Underwater Vehicle)
116. LST: Langrangian sediment trap
117. MATE: Marine Advanced Technology Education Center at Monterey Peninsula College (MPC)
118. MARS: Monterey Accelerated Research System - Monterey Cabled Observatory System
119. Maven: software project management and comprehension tool. <http://maven.apache.org/>
120. MBA: Monterey Bay Aquarium
121. MB/CA Current: a MBARI project in 2011
122. MEITS: Moored Eye-in-the Sea - deepwater video observatory for unobtrusive monitoring to be deployed on MARS

123. Meta-T: Water Temperature metadata pilot project, which is being developed under the umbrella of the Joint WMO-IOC Technical Commission for Oceanography and Marine Meteorology (JCOMM). (<http://marinemetadata.org/meta-t>)
124. MFB: Micro-Fluidics Block--added instrumentation to the ESP
125. MMI: Marine Metadata Initiative <http://marinemetadata.org/>
126. MMI ORR: MMI Ontology Registry and Repository <http://mmisw.org/orr/>
127. MOBB: Monterey Ocean Bottom Broadband project
128. Modbus: a serial [communications protocol](http://en.wikipedia.org/wiki/Modbus) <http://en.wikipedia.org/wiki/Modbus>
129. MODIS: Moderate Resolution Imaging Spectroradiometer is a key instrument aboard the Terra (EOS AM) and Aqua (EOS PM) satellites.
130. MOOS: Monterey Ocean Observing System <http://www.mbari.org/moos>
131. MRG: Red Hat Enterprise MRG is a next-generation IT infrastructure incorporating Messaging, Realtime, and Grid functionality. It offers increased performance, reliability, interoperability, and faster computing for enterprise customers. <http://www.redhat.com/mrg/>
132. MSE: MOOS Science Experiment
133. MTS: Marine Technology Society
134. MUCE: Monterey Upper Canyon Experiment
135. Mule: not an acronym. Open source ESB. <http://mule.codehaus.org/display/MULE/Home>
136. MV: Medium Voltage (converter); located in node power supply of MARS
137. MVC: Model-View-Controller software pattern - <http://en.wikipedia.org/wiki/Model%E2%80%933view%E2%80%933controller>
138. NAB: National Association of Broadcasters
139. Nagios: software to monitor all types of computer and network services. <http://www.nagios.org/>
140. Navproc: navigation processor
141. NCAR: The National Center for Atmospheric Research <http://www.ncar.ucar.edu/>
142. NCBO: The National Center for Biomedical Ontology
143. NCSA: National Center for Supercomputing Applications (at UIUC)
144. nctoolbox: nctoolbox is a Matlab toolbox that provides read-only access to common data model datasets. (See <http://code.google.com/p/nctoolbox/>)
145. NDBC: National Data Buoy Center
146. netCDF: file format used for MBARI processed data (stands for Network Common Data Form) <http://www.unidata.ucar.edu/software/netcdf/conventions.html>
147. NGDC: National Geophysical Data Center <http://www.ngdc.noaa.gov/>
148. NSF: National Science Foundation
149. NMI: NSF Middleware initiative
150. NOAA: National Oceanic and Atmospheric Administration. <http://www.noaa.gov>
151. NODC: National Oceanographic Data Center - <http://www.nodc.noaa.gov/>
152. NORIA: Network for Ocean Research, Interaction and Application. The OOI CI IO.
153. NTP: Network Time Protocol - a protocol to synchronize the clocks of computers over a network. <http://www.ntp.org/>
154. NTR: Nothing To Report
155. OA: Obstacle Avoidance
156. OBIS: Ocean Biogeographic Information System
157. OceanSITES: A worldwide system of long-term, deepwater reference stations measuring dozens of variables and monitoring the full depth of the ocean from air-sea interactions down to 5,000 meters. <http://www.oceansites.org/>
158. ODI: Ocean Design Inc.; makers of incredibly expensive underwater connectors
159. ODV: Ocean Data View <http://www.oceansites.org/>
160. ODSS: Oceanographic Decision Support System
161. OGC: Open Geospatial Consortium, Inc.® (OGC), a non-profit, international, voluntary consensus standards organization that is leading the development of standards for geospatial and location based services. <http://www.opengeospatial.org/>
162. OGC Oceans IE = Open Geospatial Consortium Oceans Interoperability Experiment. It is being hosted by the OOSTethys effort - <http://www.oostethys.org/ogc-oceans-interoperability-experiment>
163. OGF: Open Grid Forum, an open community committed to driving the rapid evolution and adoption of applied distributed computing. <http://www.ogf.org/>
164. OMF: Observatory Middleware Framework. A project funded by NSF. NCSA is the prime. MBARI is a subcontractor (at about 1/2 of the grant). UCSD (Frank Vernon) is a non-paid participant.
165. OMV: Ontology Metadata Vocabulary
166. OMV-MMI: MMI extensions to OMV
167. ONR: Office of Naval Research
168. OOI: Ocean Observatories Initiative (NSF) <http://www.orionprogram.org/OOI/default.html>
169. OOI CI: Ocean Observatories Initiative CyberInfrastructure: <http://ci.oceanobservatories.org/> (Developers site: <http://ooici.net/>)

170. OOR: Open Ontology Repository initiative - <http://ontolog.cim3.net/cgi-bin/wiki.pl?OpenOntologyRepository>
171. OOSTethys: not an acronym. A provider-to-user data systems framework, using interoperable standards, enabling discovery and use of data. <http://www.oostethys.org/>
172. OPeNDAP: The Open-source Project for a Network Data Access Protocol, the protocol used to deliver data directly from the web into user's application workspaces <http://opendap.org/>
173. ORION: Ocean Research Interactive Observatory Networks <http://www.orionprogram.org>
174. PC-104: small microprocessor for embedded systems (can run Linux) <http://www.pc104.org>
175. PCA: Principal component analysis. http://en.wikipedia.org/wiki/Principal_components_analysis
176. PCR: Polymerase Chain Reaction, procedure to make millions of copies of DNA from a few
177. PDR: preliminary design review ("Hey, I thought of something. Is it a stupid idea?")
178. PDU: Power Distribution Unit (? not sure about this though)
179. pH: a measure of the [acidity](#) or [alkalinity](#) of a solution. pH is a shorthand for its mathematical approximation: in chemistry a small p is used in place of writing log₁₀ and the H here represents H⁺, the concentration of hydrogen ions. <http://en.wikipedia.org/wiki/PH>
180. pl-navproc: the name of a computer

Name: pl-navproc.pl.mbari.org
Address: 134.89.22.11

we think navproc was the shortened name for navigation processor; we guess, "Point Lobos Navigation Processor".

181. PRDK: PUCK Reference Design Kit <http://www.mbari.org/pw>
182. PUCK: Pluggable Underwater Connector with Knowledge <http://www.mbari.org/pw>
183. PUP: Precision Underwater Positioner
184. QNX: A Unix-like real-time operating system
185. RAID: Redundant Array of Independent Drives <http://en.wikipedia.org/wiki/RAID>
186. RBI: Runs Batted In
187. RC: Radio controlled (airplane)
188. RCO: Regional Cabled Observatory <http://orionprogram.org/PDFs/RevisedOOICND08Mar07.pdf>
189. RDF: W3C Resource Description Framework
190. ROS: Remote Ocean Systems <http://www.rosys.com/>; Also Robot Operating System <http://www.ros.org/>
191. ROV: Remotely Operated Vehicle
192. RS-232 (Recommended Standard 232) is a standard for [serial](#) binary data signals connecting between a DTE ([Data terminal equipment](#)) and a DCE ([Data Circuit-terminating Equipment](#)). It is commonly used in [computer serial ports](#). <http://en.wikipedia.org/wiki/RS-232>
193. RSN: Regional Scale Node. See RCO. <http://www.ooi.washington.edu/rsn/secondary.jsp?keywords=WHATSR&title=What%20is%20RSN?>
194. SAP: Single Axis PUP
195. SCI: Semantic Collaboration Infrastructure. A multi-institutional proposal to NSF under SI2-SSI (Software Infrastructure for Sustained Innovation, Scientific Software).
196. SCPI: Self Contained Plankton Imager, an MBARI project in 2011
197. SDCI: Software Development for Cyberinfrastructure. A program at NSF.
198. SeaDataNet: Multi-national, multi-system data management project funded by the European Union, which will provide metadata, data and products as a unique virtual data centre. (<http://www.seadatanet.org/>)
199. SENSORIS: Sensor Data Integration System. A multi-institutional NSF project under Software Development for Cyberinfrastructure (SDCI) Data Improvement (Office of Cyberinfrastructure)
200. SensorML: not an acronym; markup language to describe sensors and their data. <http://vast.uah.edu/SensorML/>
201. SIFT: Scale-invariant feature transform http://en.wikipedia.org/wiki/Scale-invariant_feature_transform
202. SIIM: Science Instrument Interface Module
203. SIMM: Single In-line Memory Module
204. SIMPA: Seafloor Imaging of Marine Protected Areas. A software tool produced under MBARI projects 900821and 901016.
205. SIO: Scripps Institute of Oceanography
206. SMD: Soil Machine Dynamics Hydrovision <http://www.smdhydrovision.com>
207. SMP: Symmetric Multiprocessing
208. SNMP: (Scripps NCSA MBARI Prototype) a proof of concept project staffed at MBARI as part of SENSORS to integrate instrument network capabilities of SIO and MBARI with a policy management framework provided by NCSA and visualization capabilities provided by Calit2-Viz within an ESB-based integration infrastructure.
209. SOPAC: Pacific Islands Applied Geoscience Commission <http://www.sopac.org>

210. SOS: Sensor Observation Service, one of the SWE (Sensor Web Enablement) standards
211. SOSC: Smart Ocean Sensors Consortium <http://www.rbr-global.com/about-rbr/rbr-news/124-rbr-joins-sosc?lang=>
212. SPARQL - Simple Protocol and RDF Query Language
213. SPE: Solid-phase extraction; method from removing DNA or RNA from a 'soup'
214. SPS: Sensor Planning Service, one of the SWE (Sensor Web Enablement) standards
215. SRVI: Schmidt Research Vessel Institute
216. SSDS: Shore-Side Data System <http://www.mbari.org/ssds/> <http://new-ssds.mbari.org:8080/ssds/>
217. STOQS: Spatial Temporal Oceanographic Query System - product of the [2010 Upper Water Column Data Visualization project](#)
218. SURA: The Southeastern Universities Research Association
219. SURF Center: Sensors for Underwater Research of the Future
220. SVN: Subversion which is a source code repository.
221. SWE: Sensor Web Enablement set of standards (approved by OGC)
222. SWE Common: next-generation core for Sensor Web Enablement standards
223. TAMU: Texas A&M University
224. TDS: THREDDS Data Server, an application server used to aggregated and enhance the metadata of other OPeNDAP data sources <http://www.unidata.ucar.edu/projects/THREDDS/>
225. TERCOM: Terrain Contour Matching
226. THREDDS: Thematic Realtime Environmental Distributed Data Services <http://www.unidata.ucar.edu/projects/THREDDS/>
227. TREX: Autonomy group mission planning/scheduling software
228. TRN: Terrain-Relative Navigation. Using sensors that can "read" nearby terrain and comparing that with a known map of the area to determine where you are.
229. TVS: Transient voltage suppressor--special type of diode that protects against over-voltage conditions
230. TWIC: Transportation Worker Identification Credential (TWIC™) http://www.tsa.gov/what_we_do/layers/twic/index.shtm
231. UART: universal asynchronous receiver/transmitter, a type of "asynchronous receiver/transmitter", a piece of [computer hardware](#) that translates data between parallel and serial interfaces. <http://en.wikipedia.org/wiki/UART>
232. UCB: University of California-Berkeley
233. UCSB: University of California, Santa Barbara
234. UCSD: University of California, San Diego
235. UI: User Interface
236. UIUC: University of Illinois at Urbana-Champaign
237. UNOLS: University-National Oceanographic Laboratory System. An organization of [61 academic institutions](#) and National Laboratories involved in oceanographic research and joined for the purpose of coordinating oceanographic ships' schedules and research facilities.
238. UPS: Uninterrupted Power Supply
239. URI: Uniform Resource Identifier - <http://www.ietf.org/rfc/rfc2396.txt>
240. URN: Uniform Resource Name - <http://www.ietf.org/rfc/rfc2141.txt>
241. UUST: Unmanned Untethered Submersible Technology Symposium <http://www.ausi.org/uust/uust.html>
242. UW: University of Washington
243. VAAP: Video Annotation Analysis and Presentation <http://www.mbari.org/rd/projects/2010/information.htm>
244. VARS: Video Annotation and Reference System. <http://vars.sourceforge.net/>
245. VB: Variable Buoyancy
246. VB: Visual Basic
247. VCS: Vehicle Control System
248. VENUS: Victoria Experimental Network Under the Sea. A cabled ocean observatory. <http://www.venus.uvic.ca/>
249. VHDL: VHSIC (Very-High-Speed Integrated Circuit) Hardware Description Language
250. VINE: Vocabulary INtegration Environment
251. Voc2RDF: Vocabulary to RDF conversion tool
252. W3C: The World Wide Web Consortium. <http://www.w3.org/>
253. WATERS Network: WATER and Environmental Research Systems Network (WATERS Network)
254. WCS: Web Coverage Service
255. WFS: Web Feature Service
256. WHOI: Woods Hole Oceanographic Institute
257. WMO: World Meteorology Organization, <http://www.wmo.ch>
258. WMS: Web Map Service
259. WNS: Wet node simulator (MARS observatory simulation environment)

- 260. XML: Extensible Markup Language <http://en.wikipedia.org/wiki/XML>
- 261. XSD - XML Schema Definition http://en.wikipedia.org/wiki/XML_Schema

SVN at MBARI

This page last changed on Nov 10, 2010 by [salamy](#).

Subversion

SVN (Subversion) is a tool used by many software developers to manage changes within their source code tree. SVN provides the means to store not only the current version of a piece of source code, but a record of all changes (and who made those changes) that have occurred to that source code. Use of SVN is particularly common on projects with multiple developers, since SVN ensures changes made by one developer are not accidentally removed when another developer posts their changes to the source tree.

To use SVN first requires notification to the server administrator: Karen Salamy (salamy@mbari.org).

NOTE: In the rare event the administrator is on vacation or unavailable, please submit your SVN request to the IS Remedy page at: <https://ishelp.shore.mbari.org/helpdesk/WebObjects/Helpdesk> and it will be assigned by IS appropriately based on urgency level.

In order to access a Subversion repository, you must install a special piece of software called a Subversion client. Subversion clients are available for most any operating system. Information about the official subversion client can be found at <http://subversion.tigris.org/>. Excellent documentation on using Subversion can be found at <http://svnbook.red-bean.com/>

The MBARI SVN repository may also simply be "viewed" from a web browser for any users who wish to look at its existing source code tree structure at <http://kahuna.shore.mbari.org/viewvc/?root=svn>.

[SVN Client Software Downloads](#) - Windows, Macs, Linux Systems

[Setting up SVN Client to MBARI SVN Repository](#)

[Adding your code to MBARI's SVN Source Code Repository](#) - Non SVN client method

[View the SVN Repository](#)

Adding your code to MBARI's SVN repository

This page last changed on Jun 26, 2008 by [salamy](#).

Adding your code

Here's the standard method of adding a new directory to the subversion repository.

1) On your local machine create the following directory structure

```
tempdir
`~~ myproject      <-- Replace 'myproject' with the name of your project
`~~ tags
`~~ branches
`~~ trunk
`~~ agent
`~~ ctd2007
```

2) cd to *tempdir* and run the following command

```
svn import myproject svn+ssh://kahuna.shore.mbari.org/svn/repos/myproject
```

3) To check out *myproject's* main trunk on another machine you run the command

```
svn co svn+ssh://kahuna.shore.mbari.org/svn/repos/myproject/trunk myproject
```



Be Careful

Notice that we've added '/trunk' to the SVN URL above to check out only the trunk (main development line). If you omit '/trunk' you will check out all tags and branches as well as the trunk.

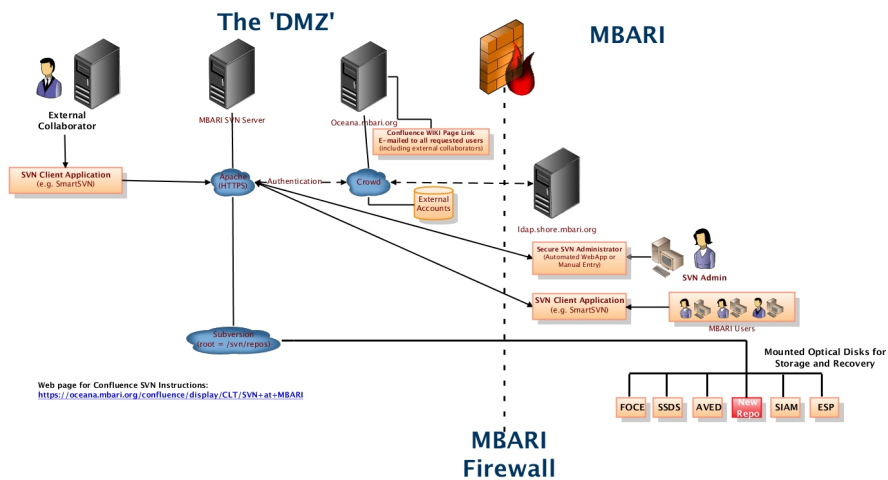
A free on-line guide to SVN is available at <http://svnbook.red-bean.com/>. The most useful svn commands are

```
svn status          <-- shows you the status of files in your project
svn add some/new/file  <-- add a new file to version control
svn del some/old/file  <-- remove a file from version control
svn update           <-- Get changes from the repository
svn commit -m "some commit message" <-- commit changes to the repository
svn help
```

If you find that you can't connect to the svn repository contact Karen Salamy (salamy@mbari.org). She may need to give you access permissions.

Secure SVN Development

This page last changed on Mar 12, 2009 by [kgomes](#).



Desired behavior

1. We can create accounts on Subversion (Crowd local store is where we can do this) for external collaborators. This will prevent them from having any access to MBARI resources.
2. It may eventually get to the point where users can create new repositories (projects), but for now, we will assume that the process of creating a new project is a manual one done by an admin.
3. There is one repository per project.

References

1. [Integrating Crowd with Apache](#)
2. [Integrating Crowd with Subversion](#)

SVN Client Software Downloads

This page last changed on May 17, 2012 by [salamy](#).

SmartSVN is the recommended SVN GUI client software to access the MBARI SVN Repository on kahuna.

SmartSVN GUI Installation Files

Windows

(Note: all require JRE 1.4.1 or higher)

- [SmartSVN-7.0.1.zip](#) - **NEW!**
-

Macintosh

(Note: all require Mac OS X, version 10.4 or higher)

- [SmartSVN-Mac-7.0.1.tar](#) - **NEW!**
-

Linux

(Note: all require JRE 1.4.1 or higher)

- [SmartSVN-Linux-7.0.1.tar.gz](#) - **NEW!**

Setting up SmartSVN

This page last changed on Nov 09, 2008 by [salamy](#).

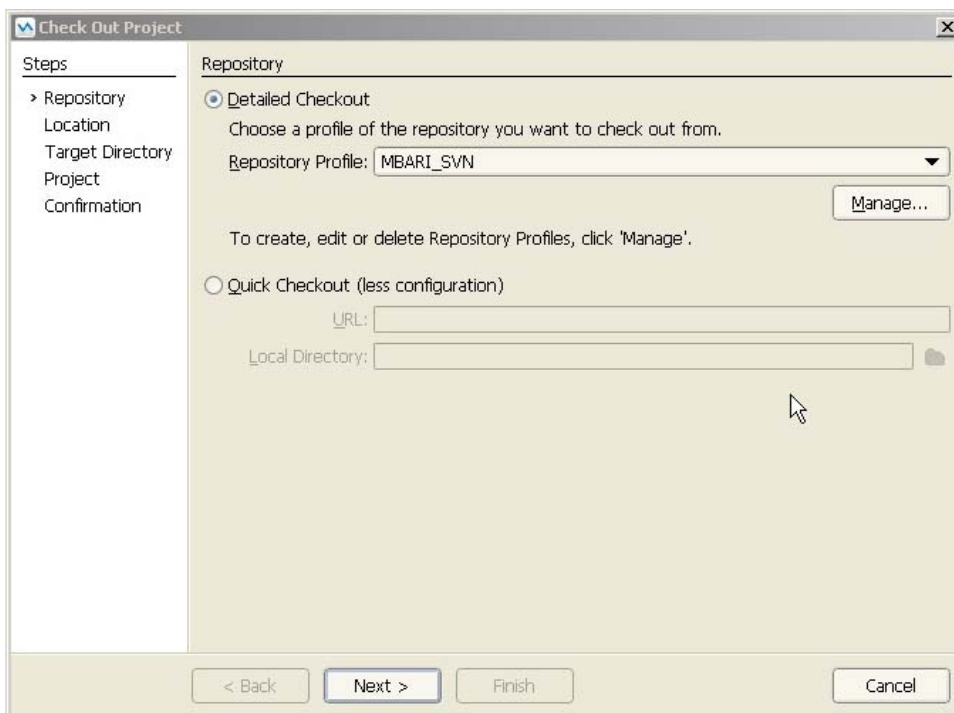
Download and Install Client

To install the SmartSVN client, first download the file from the [SVN Client Software Downloads](#) Page and install on your computer.

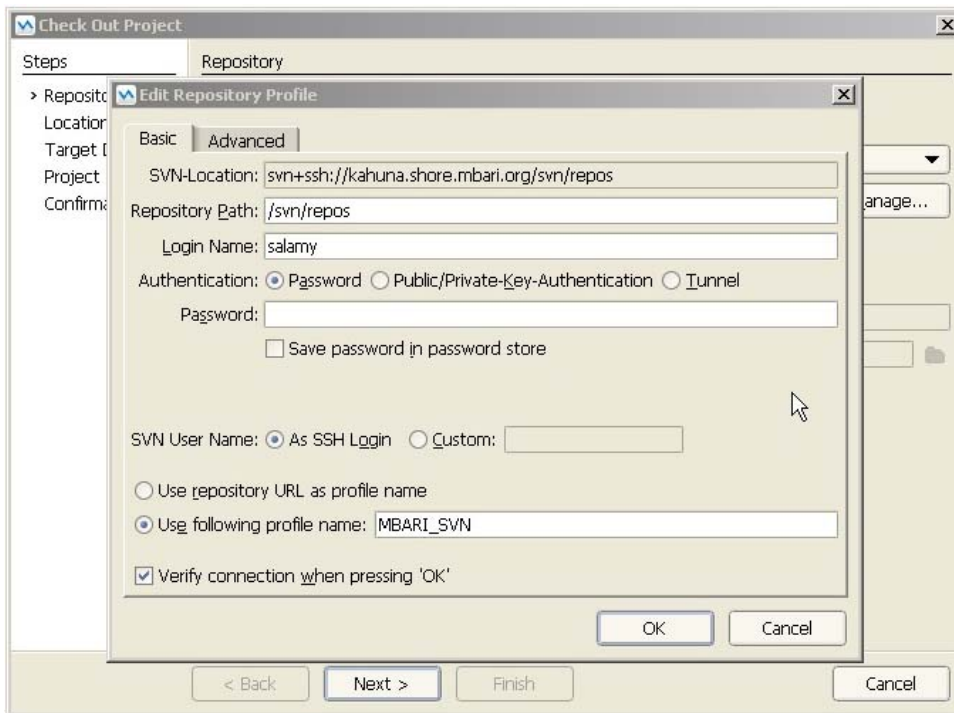
Setting up SmartSVN to see the SVN repository on Kahuna

To set up the SmartSVN client, run the application. Next, choose "**Project**" from the Main Menu and then click on "**Check Out...**"

This should bring you to the following page:



Click on the "**Manage**" button, create and edit a new profile called "**MBARI_SVN**" as follows:



NOTE!! Make sure you use your own personal **Login Name** when creating this profile and check that the **"Verify connection when pressing 'OK'"** box is clicked in order to verify that your computer is able to connect with the SVN server kahuna.

Once you have properly set up your Profile Name, when you click **"Project"** and **"Check Out..."** from the Main Menu, you should see a page to login to your account on the repository. Use your standard MBARI network account to log in. Once you have logged in successfully, you will see a list of all projects on the MBARI SVN Repository.

UML

This page last changed on Mar 03, 2009 by [kgomes](#).

We had some experience with a product called Together Control Center from TogetherSoft. TogetherSoft was then purchased by Borland and they really managed to mess it up. It was really behind (no UML 2.0) support and was in dire need of an update. Borland tried to release it as an Eclipse product and it really did not work well. So we started to look around. Right around the time I started investigating other UML Tools, Borland came out with Together 2006 For Eclipse. So to give it a fair shake, we tried several tools (I found a fairly good tool list at <http://www.softdevtools.com>):

- [Artisan Studio](#)
- Borland's Together
- [IBM Rational Software Architect](#)
- Oracle JDeveloper
- [MagicDraw](#)
- Omondo
- [VisualParadigm](#)

Turns out the OOI group was interested in a similar tool, but had to have some special features like SysML and State Diagrams. I looked up SysML and there was a short list of tools supporting SysML:

- MDG Technology for SysML (Sparx Systems)
 - SysML Support: Sparx provides a plug-in to their popular and award-winning UML 2.x modeling tool, Enterprise Architect (EA), that supports the OMG SysML 1.0 specification. Consequently, you will need Enterprise Architect 6.5 to run this plug-in. However, both EA and the SysML plug-in are reasonably priced, so you will likely find this combination an outstanding value when you compare it with the competition.
- Artisan Studio (Artisan Software)
 - SysML Support: Supports some early version of SysML; not clear when the product will be updated to support OMG SysML 1.0.
- SysML Toolkit (EmbeddedPlus)
 - SysML Support: Embedded Plus provides a SysML plug-in for IBM's Rational Software Development Platform. Supports some early version of SysML; not clear when the product will be updated to support OMG SysML 1.0.
- Rhapsody v. 6.1 (Telelogic)
 - SysML Support: Supports some early version of SysML; not clear when the product will be updated to support OMG SysML 1.0.
- TAU G2 v. 2.6 (Telelogic)
 - SysML Support: Supports some early version of SysML; not clear when the product will be updated to support OMG SysML 1.0.

Decision Matrix

Tool Name	Win	LINU	Mac	Version Tested	UML Version (1.X/2.X)	SysML Support	State Diagrams	Language	IDE Integration	Round Trip	Code Generation	Target Platform	Comments	Cost
Altova UModel				2.2			Java, C#, VB, etc.	None			Windows	The windows only thing is a big issue.		\$189
Apollo									Eclipse	Only for Java	Java, C++, etc.	Currently only supports Class diagrams ... ouch!	Subscription: \$56/year	

Artisan Studio 			2.X			C, C ++, Java, Ada, Spark Ada				VxWorks and Win32	Not a very professional install. Could not get working to test (see Artisan Studio).	
Borland Together 			2.0			Java, C ++, Eclipse		The best by far			After getting the license a hell worked out, I tried to just create some simple models but I could not change that associations by right clicking on them and if you use link templates and tried to change them, the links would disappear! OK, I figured a way around that	\$1500 (this has a back support penalty so the first round would cost \$3000, then \$1500 each year after that)

												and you need to delete the old and create a new. After that, everything looks pretty good.	
Embedded Plus													
IBM Rational Rose Technical Developer													
IBM Rational Software Architect												Still have not gotten license issues cleared up so have not been able to test	\$5500/ \$9625 float
JDeveloper													
MagicDraw				2.1			Java/J2EE, C#, C++, CORBA IDL, .NET, XML Schema, WSDL, Database Schema, DDL Generation		Yes for Java and C++	Yes, same list as Languages		I got it up and running. It does not generate all the clean of HTML and I could not get	This is a bit involved. I figured we would want Enterprise mobile licenses as this allows on user to have

											could just be a Mac thing, but ouch!). I also could not find an easy way to define associations and have them link to getter methods in the code.	for each license. SysML and DODAF plugins cost extra and would be about \$549 for both (each license).
Omondo												
Posiedon									Java only	Java, C++, C#, CORBA, IDL, Delphi, Perl, PHP4, SQL DDL, VB.NET	In using the interfaced/several menus and components, just disappear almost making it unuseable ... UG!	Standard: \$249, Professional: \$875, Embedded Enterprise: \$1549, just \$1590
PragmaDev												
Rhapsody												
SmartDraw												
Sparx				2.1	Yes, with MDG		C++, Java, C#, VB.Net, Visual Basic, Delphi, PHP, Python	Visual Studio and Eclipse don't think automatically	Yes, but I don't think automatically	Y		There are two pieces, Enterprise Architect(EA) and MDG Technology for SysML. EA= \$239, MDG= \$100

							and ActionScript				EA covers UML and MDG adds in SysML support. This one along with Borland one the Jolt award for 2006 for productivity. The EA tools did not really handle code well. Associations with multiplicity did not generate code correctly.	
TAU												
Umbrello												
Visual Paradigm	✓	✓	✓	2.X	Looks like they support certain aspects of SysML. ✓	Instant Reversal Engineering on Java, Java Class, C++ Source, DDL or EXE files, CORBA IDL Source, Ada 9x	Eclipse, NetBeans, IntelliJ, JDeveloper, Visual Studio, JBuilder, SunONE, WebLogic	✓	✓		I had some issues when running in Eclipse when I had other packages like MyEclipse pointing to the same	\$1750 (no maint.)

							Source, XML, XML Schema, JDBC, Hibernate, PHP 5.0 Source. Instant Code Generator on Java Source, C# Source, VB .NET Source, PHP Source, ODL, ActionScript Source, IDL Source, C++ Source, Delphi Source, Perl Source, XML Schema, Python Source, and Objective- C Source				Eclipse. When I used a clean Eclipse install, seemed to work great. Has a bunch of other tools like DB, etc. that I really did not try out. Associations did not generate code! Awaiting technical support. I did hit what might be a show stopper. When specifying associations of multiplicity, you cannot specify how you want the multiplicity generated in code (Collection, Vector, Array, etc.). This is a
--	--	--	--	--	--	--	---	--	--	--	--

												HUGE issue! Yep, I confirmed it. You can tell the code generator what kind of collection you wish to use in the UML tool, but in SDE you cannot specify, it always uses object arrays (ouch!). So much for VP.
--	--	--	--	--	--	--	--	--	--	--	--	---

Artisan Studio

This page last changed on Oct 30, 2006 by [kgomes](#).

My initial impression of the product is pretty poor. The installation was not very professional at all. The install requirements were very picky and if you had a PATH environment variable longer than 1000 characters (most people probably do), the installation would crash. Getting a license was very slow and their support seems like a pretty small, rag-tag bunch.

OK, got it installed and I am trying to create a new model. You would think that would be easy. I opened up Artisan Studio and selected File->New Model ... and then tried to create a model but kept getting "File already exists" no matter what I did. I then looked at their online help and it says to create a new model, I have to go to the "Model Repository" and select "New Model" ... but there are no "Repositories" in the repository neighborhood. OK, so I created a new repository on my computer. Tried to connect to the new repository, but it is telling me it is not started. Went to the admin tool again and started it. Now connecting it tells me that the repository is not initialized!!!! Wow, what a user friendly tool. OK, I have been beating on this thing, but it keeps telling me it is not initialized, but I can't find a way to initialize it. I give up.

IBM Rational Software Architect

This page last changed on Oct 30, 2006 by [kgomes](#).

Phone conversation with support guys. I have been working with Ahmad 1 (800) 656-3035 (option 1, then extension 7917).

Eric, Ahmad:

.NET: Microsoft has done their own visual modeling, IBM will not compete. Software architect does not support .NET, 3rd party can reverse .NET. Pretty loose coupling. Software architect, fairly close to Together (code visualization), harvesting (reverse), (forward) transformation: java, c++. ORM - Lightweight: visualization of databases - EJBs are supported (EE 5 will be supported). Serious data modeling (database architect, separate product). Difference with Together (very similar). Big difference is what tools it integrates with (testing, requirements). Rational Rose Technical Developer is the place to go for embedded systems. Moving into RSA (probably 2 years).

Dan Thompson: (Boulder, CO) is my regional sales rep.

Ahmad came back with quotes:

Gov't qualification: drop %25. 5 user licenses (\$5500/user if govt \$4637) or 2 floating (\$9625/license if govt. \$8114/license). Preferred business partners could bring it down some.

MagicDraw

This page last changed on Oct 30, 2006 by [kgomes](#).

Brian was trying to build a set of classes, but could not get it to work.

VisualParadigm

This page last changed on Oct 30, 2006 by [kgomes](#).

Nice that you can add custom stencils for using your own custom elements in a model. After creating a Java Project in Eclipse, I then opened a SDE project, and created diagrams, but could not get the diagram to display. I opened the same project in the straight Visual Paradigm for UML and it seemed to work nicely.

SearchNavigation

This page last changed on Nov 12, 2008 by [kgomes](#).

Searching Software Infrastructure

TreeNavigation

This page last changed on Nov 12, 2008 by [kgomes](#).

Software Infrastructure Home

This page last changed on Apr 25, 2016 by [kgomes](#).

Welcome to the Software Infrastructure page. This site is designed to catalog the different pieces of software infrastructure in use here at MBARI.

MBARI Central Authentication Service

This page last changed on Apr 25, 2016 by [kgomes](#).

This page documents the workings and usage of MBARI's central authentication service for applications. This page is primarily for developers that will be using the authentication service to assist in user management.

The basis for this service is OAuth 2.0. The idea is that this central service allows people to login using a variety of identity providers and then the applications that we are developing at MBARI can take a token that is provided to the client by the authentication service and query for information about the user and whether or not their login is valid.