VWT How to Install

This Document

This document provides instructions for installing the Verification System.

Table of Contents

Background

The Verification System was written in such a way that it can be used for 2 different purposes. One purpose is to generate static graphics and/or ascii files on a cron. This mode (from here on known as command line mode) can be run entirely on command line. The other purpose is to be loaded as an applet in a browser, allowing the user to interact with the Verification System (from here on known as the Verification Web Tool [VWT]). Each mode has slightly different requirements and installation instructions. This will be addressed in each section below.

A particular installation of the Verification System can only be run in one mode at a time (eg. an installation compiled for running on command line must be recompiled to run as a web tool, and vice versa).

Requirements

The requirements vary depending on the mode the system will be run in

Command Line Direct Access Mode

Server-side

Hardware:

- XXX RAM
- XXX hard disk space

Software:

- MySQL >= 4.1.22

Client-side

Hardware:

- XXX RAM
- XXX hard disk space

Software:

- Java SDK >= 1.6 (to build the system)
- Java JRE >= 1.6 (to run the system)
- Ant (software used to build the system)
- Write permission in the ./output subdirectory of the process
- X11 (to generate images from the command line)
- Xvfb (to generate images on cron)

Command Line Web Services Mode

Server-side

Hardware:

- XXX RAM
- XXX hard disk space

Software:

- MySQL >= 4.1.22 (to serve the data)
- Tomcat >= 5.5.23 (to run the Java servlet)
- Java JRE >= 1.6 (to run the Java servlet)

Client-side

Hardware:

- XXX RAM

- XXX hard disk space

Software:

- Java SDK >= 1.6 (to build the system)
- Java JRE >= 1.6 (to run the system)
- Ant (software used to build the system)
- Write permission in the ./output subdirectory of the process
- X11 (to generate images from the command line)
- Xvfb (to generate images on cron)

Web Tool Mode

In this mode, it's assumed that the system will be built on the server, and the client is the user running the web tool.

Server-side

Hardware:

- XXX RAM
- XXX hard disk space

Software:

- MySQL >= 4.1.22 (to serve the data)
- Tomcat >= 5.5.23 (to run the Java servlet)
- Java JRE >= 1.6 (to run the Java servlet)
- Java SDK >= 1.6 (to build the system)
- PHP >= 5.0
- Apache >= 2.0

Client-side

Hardware:

- XXX RAM

Software:

- Web browser with Javascript enabled. The following browsers have been tested:
  - Mozilla Firefox 3.6.4 on Linux
  - Mozilla Firefox 3.0 on Windows
  - Firefox ? on Mac OSX
  - Safari ? on Max OSX
  - Internet Explorer

Setting Up the Database

The Verification System requires forecasts, observations, and other reference information to be stored in a MySQL database. More detailed information is available in the Data Handbook (which needs to be updated by Mike Charles).

On the Compute Farm (CF), the MySQL VM is vm-lnx-cpccfmysql. This is where the databases must be installed for the version of the VWT running operationally on the CF. These are the databases that are used for the VWT:

- cpc_observations
- cpc_forecasts
- cpc_climatologies
- cpc_reference
- cpc_vwt

Databases are already in place on the Compute Farm, and therefore do not need to be installed. If you are an account manager on the Compute Farm, you should skip the rest of the “Setting Up the Database” section.

Step 1 – Log into MySQL

```
mysql -h server -u username -p
```

Then put in your password.

This MySQL user needs to have permission to create databases and tables.

Step 2 – Running the SQL script

Running ./scripts/db_setup_createDBs.sql will create all the necessary databases and tables.

If any of the above databases already exist, THEY WILL BE DELETED! The same is true for the tables contained in them!

Type the following once logged into MySQL:

```
\. ./scripts/db_setup_createDBs.sql
```

'\.' followed by an SQL script name is how an SQL script is run from the MySQL prompt.

Downloading the Verification System

The Verification System is located in CPC's Subversion repository at the following URL:

https://cpc-devtools.ncep.noaa.gov/svn/dev/projects/verification/land_air/all_ranges/us/Verif_System

We strongly recommend downloading a tagged version, which are always stable versions. To browse all existing tagged versions, visit this URL:

https://cpc-devtools.ncep.noaa.gov/svn/dev/projects/verification/land_air/all_ranges/us/Verif_System/tags

To download the latest tag (eg. 1.0), type the following on the command line:

```
svn export https://cpc-devtools.ncep.noaa.gov/svn/dev/projects/
verification/land_air/all_ranges/us/Verif_System/tags/1.0
Verification_System
```

This will create a directory called Verification_System containing all the Verification System code.

Compiling and Deploying

The instructions for building the system depend on which mode you plan to run the system in. Again, there are 2 main ways of running the Verification System, on the command line and as a Java applet. On the command line, the Verification System can access the MySQL database both directly and using web services. When running on the CF, direct access is allowed. When running outside the CF (cpc-lw-work1 or a Linux machine), web services must be used. In summary there are 3 distinct modes of running the Verification System. The instructions for compiling and deploying follow for each mode.

Command Line Direct Access Mode

Use this mode if you're on the CF and want to run the Verification System on the command line or on cron. Note that when you compile for this mode, you can't use the Verification System in any other mode until you compile it specifically for that mode.

Step 1 – Setting up the config file

When running in Command Line Direct Access Mode, only the config file input/verif_direct_access.conf is used. Do the following to setup this config file:

1. Copy the example config file to a real config file:

   ```
   cd input
   cp verif_direct_access.conf.example verif_direct_access.conf
   ```

2. Edit the config file, following the instructions in the file.

Here are the MySQL settings that should be used if you're running on the CF:

- host – vm-lnx-cpccfmysql
- user – vwt_user
- password – contact Mike Charles
- db_fcst – cpc_forecasts
- db_obs – cpc_observations
- db_clim – cpc_climatologies
- db_ref – cpc_reference
- db_vwt – cpc_vwt
- dateFormat – yyyy-MM-dd

Here are the QC settings that should be used if you're running on the CF:

- goodDataThreshold – 95
- goodScoreThreshold – 95
- goodScoreThresholdReliability – 95

Step 2 – Compiling the system

Run the following command inside the Verification System root directory:

```
ant compile_command_line_direct_access
```

This will compile the Verification System and place a jar file in the ./build directory. This jar file is necessary to run the Verification System.

The command-line tool Ant is required for this step.

Step 3 – Running the system

More detailed instructions for running the system can be found in the Verification System README, however here is a quick way to run the system to make sure the installation worked:

1. The Verification System needs to have the $VERIF_HOME environment variable defined to run. Set it to the path of the main directory that you downloaded above.

2. The Verification System reads in settings (eg. variable, date range, score type, etc.) from an XML file in the ./input directory. Copy the quicktest settings XML file over the existing settings XML file:

   ```
   cd input
   cp settings_quicktest.xml settings.xml
   ```

3. Run the following command to run the Verification System using settings from this XML file:

   ```
   ant run_command_line_direct_access
   ```

If you used the quicktest XML file, you should now see a timeseries image in the ./output directory.

Command Line Web Services Mode

Use this mode if you're on the cpc-lw-work1 or a Linux machine and want to run the Verification System on the command line or on cron. Note that when you compile for this mode, you can't use the Verification System in any other mode until you compile it specifically for that mode.

Step 1 – Create a certificate to sign Java jar files

When running in this mode, it's necessary to "sign" the client jar file in order for it to run. Ant will sign the jar file for you, but Ant can only do this when there is a key for the VWT on the machine you're running it from. This can be done with the following steps:

1. Execute this command:

   ```
   keytool -genkey -keyalg rsa -alias vwt
   ```

2. Enter a password (on the CF, we use the password "cycl0nes" ← that's a zero)
3. Fill out the information as such:
   a. What is your first and last name? – Name of the project lead
   b. What is the name of your organizational unit? – NOAA/NWS/NCEP
   c. What is the name of your organization? – CPC
   d. What is the name of your City or Locality? – College Park
   e. What is the name of your State or Province? – MD
   f. What is the two-letter country code for this unit? – US
   g. Is CN=Your Name, OU=NOAA/NWS/NCEP, O=CPC, L=Camp Springs, ST=MD, C=US correct? – yes
4. Hit return (to use the same password you entered before)

Step 2 – Setting up the config file

When running in Command Line Web Services Mode, the config files input/verif_client.conf and input/verif_server.conf are used. Do the following to setup these config files:

1. Copy the example config file to a real config file:

   ```
   cd input
   cp verif_client.conf.example verif_client.conf
   cp verif_server.conf.example verif_server.conf
   ```

2. Edit the config files, following the instructions in the files.

Here are the MySQL settings in the server config file that should be used if you're using wwwdev1.ncep.noaa.gov as the development server (which is currently the case here at CPC):

- host – localhost
- public_user – vwt_user
- public_password – contact Mike Charles
- admin_user – vwt_admin
- admin_password – contact Mike Charles
- db_fcst – cpc_forecasts
- db_obs – cpc_observations
- db_ref – cpc_reference
- db_vwt – cpc_vwt
- dateFormat – yyyy-MM-dd

Here are the QC settings that should be used if you're running on the CF:

- goodDataThreshold – 95
- goodScoreThreshold – 95
- goodScoreThresholdReliability – 95

Here is the URL that should be specified under the "services" section in the client config file:

http://cpcintradev.ncep.noaa.gov/webapps/verif/services/VerificationSystemToolService

Step 3 – Compiling the system

Run the following command inside the Verification System root directory:

```
ant compile_command_line_web_services
```

This command will compile the Verification System and place client and server jar files in the ./build directory, as well as a war file. These files are necessary to run the Verification System.

This command also supports an optional argument that sets the name of the resulting WAR file:

```
-Dwar_file_name=[name.war]
```

If installing operationally on the Compute Farm, this option should *not* be used, which will result in the war file being named, by default, *vwt.war*.

The command-line tool Ant is required for this step.

Step 4 – Deploying the server-side code

In order to run the VWT in Command Line Web Services Mode, the war file must be placed on a server running Tomcat 5.5 or later.

1. Go to the Tomcat Manager page (get login info from Mike Charles) and see if there's an application called "vwt". If so, then click STOP, then UNDEPLOY, in that row on the right side. If not, continue to the next step.
2. In the Tomcat Manager page, click the Browse... button under WAR file to deploy
3. Select the vwt.war file created in the step above, then click Deploy.

The manager page will be refreshed and you should see a new application in the list matching the name of your .war file (without the .war).

For Windows users, you'll have to use WinSCP to copy the file from cpcwork1 to your Windows computer. Then you can upload the file using the Tomcat Manager.

Step 5 – Running the system

More detailed instructions for running the system can be found in the Verification System README, however here is a quick way to run the system to make sure the installation worked:

1. The Verification System needs to have the $VERIF_HOME environment variable defined to run. Set it to the path of the main directory that you downloaded above.
2. The Verification System reads in settings (eg. variable, date range, score type, etc.) from an XML file in the ./input directory. Copy the quicktest settings XML file over the existing settings XML file:

   ```
   cd input
   cp settings_quicktest.xml settings.xml
   ```

3. Run the following command to run the Verification System using settings from this XML file:

   ```
   ant run_command_line_web_services
   ```

If you used the quicktest XML file, you should now see a timeseries image in the ./output directory.

Web Tool Mode

Use this mode if you want to run the Verification Web Tool in a browser. Note that when you compile for this mode, you can't use the Verification System in any other mode until you compile it specifically for that mode.

Step 1 – Create a certificate to sign Java jar files

When running in this mode, it's necessary to "sign" the client jar file in order for it to run. This can be done with the following steps:

1. Execute this command:

   ```
   keytool -genkey -keyalg rsa -alias vwt
   ```

2. Enter a password (on the CF, we use the password "cycl0nes" ← that's a zero)
3. Fill out the information as such:
   a. What is your first and last name? – Name of the project lead
   b. What is the name of your organizational unit? – NOAA/NWS/NCEP
   c. What is the name of your organization? – CPC
   d. What is the name of your City or Locality? – College Park
   e. What is the name of your State or Province? – MD
   f. What is the two-letter country code for this unit? – US
   g. Is CN=Your Name, OU=NOAA/NWS/NCEP, O=CPC, L=Camp Springs, ST=MD, C=US correct? – yes
4. Hit return (to use the same password you entered before)

Step 2 – Setting up the config file

When running in Command Line Web Services Mode, the config files input/verif_client.conf and input/verif_server.conf are used. Do the following to setup these config files:

1. Copy the example config file to a real config file:

    ```
    cd input
    cp verif_client.conf.example verif_client.conf
    cp verif_server.conf.example verif_server.conf
    ```

2. Edit the config files, following the instructions in the files.

Here are the MySQL settings in the server config file that should be used if you're using wwwdev1.ncep.noaa.gov as the development server (which is currently the case here at CPC):

- host – localhost
- public_user – vwt_user
- public_password – contact Mike Charles
- admin_user – vwt_admin
- admin_password – contact Mike Charles
- db_fcst – cpc_forecasts
- db_obs – cpc_observations
- db_clim – cpc_climatologies
- db_ref – cpc_reference
- db_vwt – cpc_vwt
- dateFormat – yyyy-MM-dd

Here are the QC settings that should be used if you're running on the CF:

- goodDataThreshold – 95
- goodScoreThreshold – 95
- goodScoreThresholdReliability – 95

Here is the URL that should be specified under the "services" section in the client config file:

http://cpcintradev.ncep.noaa.gov/vwt/services/VerificationSystemToolService

Step 3 – Compiling the system

Run the following command inside the Verification System root directory:

```
ant compile_web_tool
```

This will compile the Verification System and place client and server jar files in the ./build directory, as well as a war file and a .tar.gz file. These files are necessary to run the Verification System.

This command also supports an optional argument that sets the name of the resulting WAR file:

```
-Dwar_file_name=[name.war]
```

If installing operationally on the Compute Farm, this option should *not* be used, which will result in the war file being named, by default, *vwt.war*.

The command-line tool Ant is required for this step.

Step 4 – Deploying the server-side Java code

In order to run the VWT in Command Line Web Services Mode, the war file must be placed on a server running Tomcat 5.5 or later.

1. Go to the Tomcat Manager page (get login info from Mike Charles) and see if there's an application called "vwt". If so, then click STOP, then UNDEPLOY, in that row on the right side. If not, continue to the next step.
2. Click the Browse... button under WAR file to deploy
3. Select the vwt.war file created in the step above, then click Deploy.

The manager page will be refreshed and you should see a new application called "vwt".

For Windows users, you'll have to use WinSCP to copy the file from cpcwork1 to your Windows computer. Then you can upload the file using the Tomcat Manager.

Step 5 – Deploying the server-side webpage files

In order to view the VWT in a web browser, the webpage files need to be placed on a server running an Apache web server. This also needs to be the same server that's running Tomcat (and that the server-side Java code was placed on in Step 3).

1. In order to set up the webpage files on the web server, you need to copy the .tar.gz file to the web server. CPC uses the Web Operations Center (WOC) as the public-facing webserver. To deploy this to the WOC, they must be contacted and most of these instructions would be followed by them. Internally we use the development webserver (wwwdev1.ncep.noaa.gov). Assuming this installation is internal:

   ```
   scp build/vwt.tar.gz user@wwwdev1.ncep.noaa.gov:/usr2/cpc/cpcintradev/htdocs/apps
   ```

   where user is a user account with write permission to the web area (/usr2/cpc/cpcintradev/htdocs)

2. Log into the web server:

   ```
   ssh user@wwwdev1.ncep.noaa.gov
   ```

3. Change into the web area:

   ```
   cd /usr2/cpc/cpcintradev/htdocs/apps
   ```

4. Unpackage the webpage files:

   ```
   tar -xzf vwt.tar.gz
   ```

   You should see a new directory /path/to/webroot/vwt

5. Move the config file outside of the apache Document Root. The Document Root is the top-level directory under which all web files are located on the web server. For example, if the web directory is /usr2/cpc/cpcintradev/htdocs (like it is on wwwdev1.ncep.noaa.gov):

   ```
   mv vwt/verif.conf /etc/verif.conf
   ```

6. Change the $config_file variable in the admin PHP scripts to point to the new location for verif.conf. This variable is located at the top of the PHP admin scripts.
7. Create (or add users to) an htpasswd file, which Apache users to store usernames and passwords. Type the following to create a new htpasswd file /etc/htpasswd_file (for example) and insert a user called vwt_admin (for example):

   ```
   htpasswd -c /etc/htpasswd_file vwt_admin
   ```

   You'll be prompted to create a password for vwt_admin.

   Leave off the -c if the file already exists and you're just adding a user.

   The htpasswd program is only available with Apache, so it's usually run on the webserver itself
8. The admin scripts should be password-protected using Apache's .htaccess file. There's an example one at verif/admin/.htaccess.example. Copy this to a real one:

   ```
   cd vwt/admin
   cp .htaccess.example .htaccess
   ```

   Edit the AuthUserFile option to point to the htpasswd file you created/edited in Step 7, and edit the AuthName option to specify a phrase to display when a user attempts to connect to the password-protected URL. You can test the password-protection by navigating to the URL associated with the verif files.

   http://cpcintradev.ncep.noaa.gov/apps/vwt/admin

   You should be prompted for a username and password. You should be able to enter the username and password you created in Step 7 and successfully get a listing of that directory.

The VWT itself will now be available at http://cpcintradev.ncep.noaa.gov/apps/vwt

You'll need the Java Plugin for your browser in order for the VWT to work