NASA Contractor Report NASA/CR-2008-214746

Automated VAHIRR Product
Volume 3 (VAHIRR Installation Guide)

Joe H. Barrett III
NASA Applied Meteorology Unit
Kennedy Space Center, Florida

Juli Miller
Debbie Charnasky
Robert Gillen
ENSCO, Inc.

July 2008

NASA Contractor Report NASA/CR-2008-214746

Automated VAHIRR Product
Volume 3 (VAHIRR Installation Guide)

Joe H. Barrett III
NASA Applied Meteorology Unit
Kennedy Space Center, Florida

Juli Miller
Debbie Charnasky
Robert Gillen
ENSCO, Inc.

July 2008

Acknowledgments

The authors thank the Airborne Field Mill II program team for providing data and answering questions about the VAHIRR radar product.

Available from:

NASA Center for AeroSpace Information
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

This report is also available in electronic form at

http://science.ksc.nasa.gov/amu/

Table of Contents – Volume 3

1. Installation Guide Overview ... 5
2. Linux Installation ... 5
3. ORPG Installation ... 7
4. Running the ORPG ... 11
5. CODE Installation ... 12
6. Sample Algorithms Installation ... 13
7. LDM installation ... 14
8. LDM configuration for VAHIRR ... 16
9. VAHIRR configuration in CODE/ORPG software ... 17
10. Modifying the size and products in the product database ... 19
11. Purging old LDM and Archive II radar files ... 20
12. Entering environmental data for the Hail algorithms ... 21
13. Modifying the height of the cone-of-silence ... 21
14. Installing the test version of VAHIRR over a previous installation ... 21
15. Installing the operational version of VAHIRR over a previous installation ... 22
16. Installing the AMU ORPG-clone Archiver Application ... 23
17. User Guide for the AMU ORPG-clone Archiver Application ... 25

1. Installation Guide Overview

The VAHIRR Installation Guide describes how to generate, display, and configure the automated VAHIRR Product on an Open Radar Product Generator (ORPG)-clone (Sections 2-14). It also includes directions on installing and running the AMU ORPG-clone Archiver Application, useful for archiving radar products (Sections 15 and 16). The primary references for this guide were:

- WSR-88D CODE Guide for Build 8 v2 (April 2006), available with the CODE Build 8 v2 software download,
- WSR-88D CODE website: http://www.weather.gov/code88d/index.html, and
- CODE manual pages in $HOME/man directory on the ORPG-clone.

2. Linux Installation

Reference: Vol 1. Document 1 - CODE Specific ORPG Installation Instructions - April 2006 for ORPG Build 8, Section I Preparation for Installation, Linux PC Platform

1. Download the distribution file (**`pub_code_b8v2.tar`**) for Build 8, version 2 of the ORPG software. The distribution file can be obtained from the AMU. Do not use the more recent version of the ORPG software that is on the WSR-88D CODE website, since the VAHIRR product was developed and tested with Build 8, version 2. Make sure that the CODE System Requirements (see the system_requirements.html document in the distribution file) are met. CentOS (the open-source version of Red Hat Enterprise Operating System) can be substituted for Red Hat Enterprise Linux. Also, either the Server or Workstation versions of Red Hat Enterprise are acceptable. Make sure that the required compilers and software development tools are present:
   - **Compilers and utilities**
     - GCC 3.2.3 (includes gcc, g++, and g77), or more recent version,
     - GNU make 3.79.1, or more recent version,
     - GNU linker in binutils 2.14.90.0.4, or more recent version, and
     - glibc 2.3.2, or more recent version.
   - **Special bsd style install script** (provided by the distribution file)
2. Either perform a full installation of all packages on the operating system installation CDs, or make sure that all compilers and packages are present as listed in the CODE System Requirements and "Software Package Selection during Installation of Red Hat Enterprise 4 Workstation" (**`linux_rh_e_ws_pkg_sel.html`** in the distribution file) documents.
3. It is recommended that the ORPG not be installed over a prior installation. To reinstall the ORPG software, backup any necessary files (e.g. modified source code files). Then, reinstall Linux.
4. Use the 32-bit operating system installation CDs because the ORPG software is 32-bit only. Install either Red Hat Linux Enterprise 4 Workstation/Server or CentOS 4.
5. If the Red Hat Network is going to be used for automatic updates, it is recommended that all required packages be installed before registration with the Red Hat Network.
6. The following instructions will require administrative privileges:

   a) Installing any required software packages besides CODE, that are not already installed in the operating system.

   b) Modifying the **`/etc/sysct1.conf`** and **`/etc/inittab`** files.

   c) Creating the user account into which the ORPG will be installed and in which the ORPG will run.

   d) Providing file system space for the ORPG data storage directory.

7. The Linux machine must be set up on a functional TCP/IP network with a static IP address. The ORPG is not compatible with dynamic addresses with DHCP.
8. The following disk space is required in addition to the operating system files, software development packages and patches:
    a) The account into which the ORPG is being installed should have a minimum of 650 MB of space available (1 GB recommended).
    b) An ORPG data directory with 400 MB additional space. This is the directory referenced by the environment variable **`ORPGDIR`**, and does not need to be in the ORPG home directory.
9. Do not install more than one ORPG on a Workstation.
10. Do not install Linux over Windows or another version of Linux. There should only be one operating system installed on the machine.
11. To partition the hard disk, either use the default automatic partition configuration, or create a custom partition. The default partitioning is adequate.
12. If installing over an existing installation of Windows or Linux, wipe out the files/partitions from the previous installations.
13. It is not necessary to install the Linux firewall or use Security Enhanced Linux.
14. Install Red Hat Enterprise 4 (or CentOS 4):
    a) Put in CD 1 of the Red Hat Linux Enterprise 4 installation disks.
    b) Reboot the machine so that the installation screen comes up when rebooting.
    c) Select graphical mode for the installation.
    d) Select English for the installation language.
    e) Select U.S. English for the keyboard.
    f) Select either Automatically partition or Manually partition with Disk Druid.
    g) Choose to have automatic partitioning remove all previous partitions on this system.
    h) Select the drives to use for the installation.
    i) Enter a hostname. Enter IP address and netmask.
    j) Select English (USA) for the system.
    k) Select nearest city for time zone. It is not necessary to use UTC time.
    l) Enter a root password.
    m) Select packages to install.
    n) Note that a complete log of the installation can be found in **`/root/install.log`** and a kickstart file containing the installation options selected can be found in the file **`/root/anaconda-ks.cfg`**
    o) After installing the files, reboot the system when prompted to.
    p) The basic configuration screens come up after reboot.
    q) Accept the license agreement. Set date and time.
    r) Create a username for the ORPG user, e.g. codev8.
    s) It is not necessary to install the extra packages on CD5.
15. The package ncompress (containing the compress and uncompress programs) is required by the utility **`play_a2`** when playing back .Z or .gz files. To verify if this package is installed or not, use the command: **`rpm -qa | grep ncompress`**. If this package is not installed, copy the RPM file from the installation

CDs to **`/tmp`** and install it: **`rpm -Uvh /tmp/ncompress-4.2.4-40.rpm`**

16. Install any additional packages that are required for the ORPG development environment, as listed in the "Software Package Selection during Installation of RedHat Enterprise 4 Workstation" document.

17. Log in as the ORPG user. The user that is using the CODE/ORPG software should be the same user that installs it, because of the file ownerships and permissions. Obtain the public CODE distribution file, **`pub_code_b8v2.tar`**, from the AMU.

18. Place the tar file in a directory owned by the ORPG user. Extract the file with the command: **`tar xvf pub_code_b8v2.tar`**

19. Create symbolic links needed for Linux. Obtain the **`make_linux_links.tar`** file (in the **`~/pub_code_b8v2/guide/v1_setup_code1_19/files_support_linux/`** directory) and place into a convenient directory owned by the ORPG user.

20. Log in, or **`su`**, to root. While in the directory containing **`make_linux_links.tar`**, execute:

    ```
    tar xvf make_linux_links.tar
    ./make_linux_links
    ```

    The output of the script indicates the links that were created:

    ```
     create symbolic link `/usr/local/bin/gzip' to `/bin/gzip'
     create symbolic link `/usr/local/bin/gunzip' to `/bin/gunzip'
     create symbolic link `/usr/local/bin/bzip2' to `/usr/bin/bzip2'
     create symbolic link `/usr/bin/sh' to `/bin/sh'
     create symbolic link `/usr/bin/csh' to `/bin/csh'
     create symbolic link `/usr/bin/nawk' to `/usr/bin/gawk'
     create symbolic link `/usr/local/bin/perl' to `/usr/bin/perl'
     create symbolic link `/usr/sbin/ifconfig' to `/sbin/ifconfig'
     create symbolic link `/usr/sbin/ping' to `/bin/ping'
     create symbolic link `/usr/local/bin/cpp' to `/usr/bin/cpp'
    ```

    If the file is already present, it is used (the link is not made) and the corresponding message is not displayed.

    If the target is not present, the link cannot be made and a warning is displayed. For example:

    ```
    WARNING link /usr/bin/sh not made; target /bin/sh does not exist
    ```

    This indicates that a required software package was not installed during Linux installation.

21. Prior to running the ORPG, the shared memory segment size for the Linux kernel must be increased to 100 MB as follows:

    As root, edit the **`/etc/sysctl.conf`** file and add the following at the end:

    ```
    kernel.shmmax = 104857600
    ```

    Then, reboot the workstation.

22. Make sure that the user account for the ORPG has a minimum of 650 MB of space. The default shell for the account must be the C or tcsh shell.

23. The data directory for each ORPG installation should have 400 MB of space available. This directory is referred to by the variable **`$ORPGDIR`** in the configured ORPG. The location of the directory does not have to be within the ORPG account directory tree, but the ORPG account must have read, write, and execute permission in this directory. Do not use a directory named “data” in the ORPG account home directory.

3. ORPG Installation

Reference: Vol 1. Document 1 - CODE Specific ORPG Installation Instructions - April 2006 for ORPG Build 8, Section III Installation Instructions

Unless otherwise noted, all installation procedures should be accomplished while logged into the ORPG account.

1. Obtain the ORPG source code distribution archive file (in **~/pub_code_b8v2/guide/v1_setup_code1_19/files_orpg_sw/** directory) and place it into the ORPG account's home directory.
2. If the file is already uncompressed (103 MB rather than 30 MB), check to see if the file was renamed by dropping the .gz extension.
3. If the file is still compressed, uncompress the archive with the command: **gunzip rpg_b8v1_9_pub_src_tar.gz**
4. Extract the archive with the command: **tar xf rpg_b8v1_9_pub_src_tar**
5. Obtain the **code_orpg_files_b8v2.tar** file (in the **~/pub_code_b8v2/guide/v1_setup_codeN_nn/orpg_install/config_files/** directory) and place it into the ORPG account's home directory.
6. While in the home directory, extract the archive with the command:

   tar xf code_orpg_files_b8v2.tar

   The replacement files and an installation script are extracted into the subdirectory **code_config_b8v2**
7. In the subdirectory **~/code_config_b8v2/env**, install the development environment's configuration files with the following command: **./inst_env_config**

   a) The following files are replaced with modified CODE versions:
   ```
   $HOME/.cshrc
   $HOME/make_rpg
   $HOME/conf/make.common
   $HOME/conf/make.slrs_spk
   $HOME/src/cpc104/lib003/makefile
   $HOME/src/cpc104/lib005/makefile
   $HOME/src/code_util/tsk001/makefile
   $HOME/src/cpc104/lib006/makefile
   $HOME/src/src.make
   ```
   b) The following new files are installed:
   ```
   $HOME/build_env_cshrc
   $HOME/orpg_env_cshrc
   $HOME/test_make_cpc100
   $HOME/tools/bin/bsdinst
   ```
   c) The following files support the sh / ksh shells (not tested):
   ```
   $HOME/.profile
   $HOME/.dtprofile
   $HOME/.envfile
   $HOME/build_env_profile
   $HOME/orpg_env_profile
   ```
   d) WARNING: Even though the C shell is being used, do not delete ANY of these files. Some portions of the ORPG launch process require the presence of a basic environment from **.profile**.

   e) The files being replaced are saved with a .OLD extension.

   f) A copy of the new file is made with a .CODE extension.

   g) Required symbolic links are created for the precompiled libraries in the **~/lib/slrs_spk** directory.
8. Ignore the files in the directory **~/code_config_b8v2/orpg** for now. They will be installed after the ORPG is compiled.

   a) **<u>Configuration of the ORPG Account</u>**

   1. No changes are required in the sample **.cshrc** file installed with the script.

   2. In the **orpg_env_cshrc** file in the ORPG's home directory, the value of the **$ORPGDIR** variable must be set to the complete path to the ORPG data directory. This directory is not the same as the

~/data directory referred to by the **$ORPG_DATA_DIR** variable. Create a backup copy of the modified **orpg_env_cshrc** file.

b) **Configuration of the ORPG Build Script and Makefiles**

1. No changes are required for the Linux platform.

c) **Final steps of configuration**

1. Create a **$HOME/tmp** directory. Without this directory, the ORPG does not function properly.

2. To ensure the account environment reflects the changes just made, log out and back into the account before attempting to compile the ORPG.

3. Run **env** to check the output to verify that both the run environment file and the build environment file are being sourced at login.

4. If these environmental variables are not being set, the login may not be reading the **.cshrc** file. Try renaming the **.tcshrc** or **.profile** files, if they exist.

d) **Building the ORPG**

Patch 1:

1. Obtain **patch_b8.tar** (**~/pub_code_b8v2/guide/v1_setup_code1_19/files_orpg_sw/** directory) and place the file into the ORPG account's source code directory, **~/src** (or any convenient location).

2. While in this directory, extract the archive with the command: **tar xf patch_b8.tar**. The replacement files and an installation script are extracted into a subdirectory **patch_b8**.

3. While in the directory patch_b8, install the patched files with the following command: **patch_b8**

Patch 2:

1. Obtain **patch2_b8.tar** (**~/pub_code_b8v2/guide/v1_setup_code1_19/files_orpg_sw/** directory) and place the file into the account source code directory **~/src** (or any convenient location).

2. While in this directory, extract the archive with the command: **tar xf patch2_b8.tar**. The replacement file and an installation script are extracted into a subdirectory **patch2_b8**.

3. While in the directory **patch2_b8**, install the patched file with the following command: **patch2_b8**

Patch 3:

1. Obtain **patch3_b8.tar** (**~/pub_code_b8v2/guide/v1_setup_code1_19/files_orpg_sw/** directory) and place the file into the account source code directory **~/src** (or any convenient location).

2. While in this directory, extract the archive with the command: **tar xf patch3_b8.tar**. The replacement files and an installation script are extracted into a subdirectory **patch3_b8**.

3. While in the directory **patch3_b8**, install the patched files with the following command: **patch3_b8**

e) **Quick Test Compile**

A quick test compile of a portion of the source code should be conducted. This does not catch all possible configuration errors.

1. While in the **$HOME** directory, execute the test_make_cpc100 script with the following command: **test_make_cpc100 $HOME >& <outputfilename>**

2. After compilation is complete, compare the created log file with a typical output file to ensure that the build was successful. This level of compiler warning messages is normal. Ensure there are no Error messages. A good command for returning only the error messages is:

```
grep –e 'Error [1-9]' <output_file_name>
```

Comparing the section containing Error messages with the sample build output file may help evaluate the error. If a command cannot be found, the error is probably in the **$PATH** variable as defined in the **.cshrc** file and the build environment script. The output of the **env** command will show whether or not the account is successfully sourcing the build environment script. One cause of file access problems is not being logged in as the account owner when attempting to compile the ORPG.

3. With an unsuccessful build, look closely at the first unexpected Error message to determine the probable cause. If the cause is not obvious review all procedural steps taken to this point.

f) <u>Compiling the ORPG source code</u>

Executing the **make_rpg** script builds the entire ORPG and installs the binary executables and libraries in the appropriate directories. It takes approximately 15 minutes to build the ORPG on a typical Pentium 4 PC.

1. While in the **$HOME** directory, execute the make_rpg script with the following command:

```
make_rpg $HOME >& <outputfilename>
```

2. After compilation is complete, compared the created log file with a typical output file to ensure that the build was successful. This level of compiler warnings is normal. Ensure there are no unexpected Error messages.

The following errors in the compile output file are normal. Line numbers are approximate.

On the Linux Enterprise PC Platform:

```
===============================================
FIRST BUILD:
No errors.


SUBSEQUENT BUILD:

Ignored link error for existing library.
Line 4981 ======================================================
installing libs in ./src...
make[1]: Entering directory `/home/code8v2/src'
ln: `/home/code8v2/lib/lnux_x86/libbz2.so': File exists
ln: `/home/code8v2/lib/lnux_x86/libbzip2.so': File exists
make[1]: [libinstall] Error 1 (ignored)
Line 4985 ======================================================
```

3. With an unsuccessful build, look closely at the first unexpected Error message to determine the probable cause. If the cause is not obvious review all procedural steps taken to this point.

4. With a successful build, the procedures in the “Configuring the ORPG Installation” section should be accomplished even if the ORPG has been previously successfully built into this account.

g) <u>Configuring the ORPG Installation</u>

1. While in the directory **$HOME/code_config_b8v2/orpg**, install the ORPG configuration files with the following command: **inst_orpg_config**

The files being replaced are saved with a **.OLD** extension. A copy of the new file is made with a **.CODE** extension. In order to provide an immediate source of input data for testing the ORPG installation, three Archive II data files are installed in **$HOME/data/ar2data**.

2. The **$HOME/cfg/comms_link.conf** file does not require modification unless another source of base data (besides 8mm tape containing Archive II data or Archive II data from disk files using the CODE utility) is used or narrow band product distribution lines are configured.

3. The Remote System Services configuration file, **$HOME/.rssd.conf**, should be modified:

Make a client entry for the TCP/IP address (or hostname) of the workstation. For example:

```
# RPG Development Workstations
Client: stratus
```

Make a Pathname entry for the location of the ORPG data directory (**$ORPGDIR**), the account's home directory (**$HOME**), and the temporary directory for the operating system. For example:

```
# Pathnames
#      [$ORPGDIR]
Path: /code/data_orpg_b1
#      [$HOME]
Path: /code/home/orpg_b1
#   directory for temporary files on your workstation
Path: /tmp
#
#   not currently used for CODE
Path: /floppy/floppy0
```

Make a backup copy of this file.

4. The site adaptation data is set for Melbourne Florida.

5. The **$HOME/tools/bin/rm_orpg_data** script provides a safe means to erase all of the ORPG data files located in the data directory (**$ORPGDIR**) configured for the installation account.

6. To ensure the account environment reflects the changes just made, log out and back into the account before attempting to run the ORPG.

7. Test the installation by launching the ORPG following the instructions in Section IV of the document “Running the ORPG”.

4. Running the ORPG

Reference: Vol 1. Document 1 - CODE Specific ORPG Installation Instructions - April 2006 for ORPG Build 8 Section IV Running the ORPG

1. Log in as the ORPG user.
2. At a command prompt, enter: **mrpg -r -v startup**

   The -v option provides a verbose output. The -r option recreates all non-persistant global data stores and must be used after changing configuration files. Wait for the command prompt to return. Startup normally requires less than one minute.

3. If the systat package was not installed, the following iostat error message will be received after launching the ORPG:

```
15:18:15 mrpg:     Execute op process: vwindpro -v
15:18:15 mrpg: All operational processes started. Waiting for OP ready ...
15:18:15 mrpg: MGC_system iostat -c 2 60 > /home/code7v2/tmp/iostat.out
                                               & on  failed (ret -449)
15:18:16 mrpg: RPG State: OPERATE - mrpg_ma*e_rpg.c:1080
15:18:16 mrpg: RPG Operability Status: ONLINE - mrpg_ma*e_rpg.c:1122
15:18:16 mrpg: RPG system initialized - mrpg_pr*_cmds.c:130
15:18:16 mrpg: RPG startup completed
```

   As far as the CODE developers know, this message does not affect the ability to use CODE to develop radar products.

4. To check the status of running programs, enter **rpg_ps**.
5. Launch the ORPG user interface program with the **hci** command. When not connected to an operational RDA, it is normal for the hci to display a warning for Wideband Link Failure. The HCI does not need to be

run for the ORPG algorithm tasks to run. Documentation of the HCI is at: http://www.roc.noaa.gov/osteam/epss/epss_index.htm. There are also HCI man pages available on the ORPG-clone (hci, hci_status, etc.).

6. The ORPG utility "play_a2" is used for tape and archived data playback. In addition to the command line mode for Archive II disk files, play_a2 includes an interactive mode for both Archive II disk files and 8mm Archive II tapes.

   The play_a2 utility reads individual files, each containing a volume of Archive II data, and ingests the data into the ORPG. In order to provide a quick test of the ORPG, three files are included with the CODE ORPG configuration files and have been installed in **`~/data/ar2data`**.

   Execute the following command to ingest these files: **`play_a2 -d $HOME/data/ar2data`**

   If the HCI has been launched, observe the RDA radome indicating scanning in progress on the GUI window. If the variable **`AR2_DIR`** has been set to the **`$HOME/data/ar2data`** directory, executing play_a2 will suffice.

   Press [Ctrl]-[c] to stop the ingest of the Archive II files.

   In command mode, play_a2 can read volume files that are Unix Compressed (*.Z) as well as bzipped files. See the CODE Utility documentation contained in CODE Guide Volume 4 for additional information on play_a2. There is also a man page on play_a2.

7. Enter **`cvg`** at a command prompt, to launch the CODEview Graphics Display tool. Once the utility is launched, a product may be selected from the product database using the product list on the main CVG window. After the product is selected, the desired data packets for display are chosen from the Packet Selection popup-screen. See the CODE Utility documentation contained in CODE Guide Volume 4 for additional information.

8. To stop the ORPG tasks:

   Enter: **`mrpg shutdown`**

   Then enter: **`mrpg cleanup`**

   The **`mrpg cleanup`** command should always be executed when stopping the ORPG. The HCI and two other ORPG tasks (rssd and mrpg) are still active after shutdown and this command terminates these.

5. CODE Installation

Reference: Vol 1. Document 2 - Installing CODE Software Section II Instructions for CODE Utilities - April 2006

There are two main installation types for the CODE software – local and global. The local installation allows multiple versions of the utilities (such as CVG and CVT) to be run by separate users. A global installation type is not normally used. A global installation places the installed executables and default CVG configuration files into a location accessible by more than one account on a workstation. If using CVG 8.0 and CVT 4.0, a standalone installation option is provided. The standalone option permits CVT and CVG to be used outside of an ORPG account. The standalone option should not be used in conjunction with a local installation.

1. Perform a local installation. Verify that the **`~/tools/bin`** and **`~tools/bin/lnux_x86`** directories are in the **`$PATH`** environment variable.

2. Log in as the ORPG user.

3. Obtain the Code Utility Update **`util_update_c8v2a.tar`** file (in the **`~/pub_code_b8v2/guide/v1_setup_code1_19/files_code_sw/`** directory) and place it into the ORPG source code directory (**`$HOME/src`**) of the account.

4. While in this directory, extract the archive with the following command:

   ```
   tar xf util_update_c8v2a.tar
   ```

5. While in the **`~/src/util_update_c8v2`** directory, execute the following commands to install both CVT

and CVG updates:

```
make clean
make liball
make libinstall
make all
make install
```

6. With the local installation, the CVG default configuration files are placed in the directory **$HOME/tools/cvgN.N/.cvgN.N** (where N.N refers to the CVG version number). The environmental variable **$CVG_DEF_PREF_DIR** must be set to **$HOME/tools**, which is the value used in the account configuration file **.cshrc** supplied with CODE.
7. The environmental variable **$CV_ORPG_BUILD** must be set to the ORPG build number (for example, 8).
8. The environmental variable **$AR2_DIR** must be defined for each account that uses the play_a2 utility to replay Archive II disk files. This variable represents the default location of stored Archive II disk files. It is convenient to place each data set in individual directories under **$AR2_DIR**.
9. The cvt binary file is installed in the **~/tools/bin/lnux_x86** directory.
10. The cvg, cvg_read_db, and cvg_color_edit binary files are installed in the **~/tools/bin/lnux_x86** directory.

6. Sample Algorithms Installation

Reference: Vol 1. Document 2 - Installing CODE Software Section III Instructions for Sample Algorithms - April 2006

1. Log in as the ORPG user.
2. Obtain the CODE sample algorithm archive **code_alg_1_14.tar** (in the **~/pub_code_b8v2/guide/v1_setup_code1_19/files/code/sw directory**) and place into the ORPG's source code directory (**$HOME/src**).
3. While in this directory, extract the archive with the following command:
   tar xvf code_alg_1_14.tar
4. The archive contains the source code for four sample algorithms written in C. The source code for these sample algorithms is not included with the ORPG source code and must be installed and compiled separately. A description of each sample algorithm is provided in CODE Guide Volume 3, Document 3, Section II.
5. Because the new sample algorithm 4 produces a standard graphic product rather than a SATAP, CVG 8.0 and earlier must be reconfigured in order to display the final products. At the command prompt, enter **cvg**.

   The Product Message Type for Product ID's 1993 and 1994 must be changed from 2 - Stand alone Tabular to either 0 - Geographic or 1 - Non-Geographic. From the File->Preferences... menu dialog, go to the Product Specific Info Edit screen. See the CVG User Guide in Volume 4 of the CODE Guide for detailed instructions.
6. Prior to compiling the algorithms, the ORPG should be configured to add the tasks and products for the sample algorithms. As a convenience, the configuration information is provided in files included with the algorithm source code. So, these new products and tasks can be configured without editing the product_tables and task_table configuration files directly. To use this method:
   a) Create a subdirectory named **extensions** under the **~/cfg** directory.
   b) Copy the files **~/src/cpc305/product_tables.sample_snippet** and **~/src/cpc305/task_table.sample snippet** into that directory.
   c) In order to generate the product, the default generation table in the **product_generation_tables**

configuration file must be modified.

d) The following file configures the sample products for generation. Make a backup copy of `~/cfg/product_generation_tables` and copy the file `~/src/cpc305/product_generation_tables.sample_alg` to the `~/cfg/product_generation_tables` directory.

e) Algorithm adaptation data must be configured via specially named configuration files. The algorithms in the `tsk001` and `tsk004` directories require adaptation data. This data is provided with the source code. Copy the file `~/src/cpc305/tsk001/sample1_dig.alg` to the `~/cfg/dea/` directory. Copy the file `~/src/cpc305/tsk004/sample3_t2.alg` to the `~/cfg/dea/` directory.

7. The algorithms can be compiled by executing the following from within the `~/src/cpc305` directory:

```
make all
make install
```

Alternately, the algorithms can be compiled individually by executing the same commands from within each task subdirectory.

8. After editing the configuration tables and installing the new adaptation data, remember to start the ORPG with the -r switch to rebuild the binary configuration files.

7. LDM installation

Reference: LDM Reference, http://www.unidata.ucar.edu/software/ldm/ldm-6.4.6/basics/index.html

1. Go to the LDM webpage of Unidata: http://www.unidata.ucar.edu/software/ldm/.

2. Download the latest stable version of LDM 6.

3. It is recommended that the ORPG-clone have a monotonic clock accurate within one millisecond. The clock should be kept accurate by either running the NTP daemon or periodically executing the ntpdate utility.

4. Log in as root. Create the directory `/usr/local/ldm`. Change the account and group ownership (with the `chgrp` and `chown` commands) of this directory to the ORPG user.

5. Add ldm to `/etc/services` and `/etc/rpc`.

   a) Add the following line to `/etc/services`:

   ```
   ldm 388/tcp ldmd  # UCAR Unidata LDM
   ```

   b) Add the following line to `/etc/rpc`:

   ```
   ldm 300029 ldmd
   ```

6. Add the following entry to file `/etc/syslog.conf`:

```
local0.*      /usr/local/ldm/logs/ldmd.log
```

7. Add the string local0 to entries like the following so that LDM messages will not be written to the console (Note: entries might not be exactly like these, but they should be similar):

```
*.err;kern.notice;auth.notice;user.none;local0.none /dev/console
*.err;kern.debug;daemon.notice;mail.crit;user.none;local0.none
/var/adm/messages
```

8. Log in as the ORPG user. Add the following lines to the `.cshrc` file:

```
setenv LDMHOME /usr/local/ldm
set path=($LDMHOME/bin $path)
```

9. Make sure that `$HOME/man` is included in the `$MANPATH` variable. If not, add it.

10. Download the LDM distribution tar file and save it to the `/usr/local/ldm` directory.

11. Go to the **/usr/local/ldm** directory. Unpack the distribution using the zcat and tar utilities (assuming LDM version 6.4.6):

    ```
    zcat ldm-6.4.6.tar.Z | tar xvf -
    ```

12. Log out and back in, to reset environmental variables.
13. Go to the top-level source directory of the just-unpacked package (assuming LDM version 6.4.6): **cd /usr/local/ldm/ldm-6.4.6/src**
14. If re-building LDM, run: **make distclean**
15. Run the configure script: **./configure >&! configure.log**
16. This step customizes the installation process for the particular system.

    Inspect the file **configure.log** to verify the configuration process. If an error occurred, then the reason for the error can often be found in the file **config.log** (note the slightly different name).

17. Run the following to build the package, redirecting both standard output and standard error to the file make.log: **make >&! make.log**

    This step creates the package libraries and programs but doesn't install them. Inspect the file **make.log** to verify the build process. If something went wrong due to an environment variable, then return to the clean-up step above.

18. Run the following to install the package components, redirecting both standard output and standard error to the file install.log: **make install >&! install.log**
19. Install some components with superuser privileges: **su root -c 'make install_setuids'**

    This step is necessary for the LDM server to listen on port 388 and for the hupsyslog utility to notify the syslogd daemon when a new log file has been created.

20. Use the make utility to clean up the compilation files: **make clean**
21. Create subdirectories for data files and LDM log files. The LDM's product queue must be on a locally-mounted disk, and it is highly recommended that locally-decoded and stored data also be written to locally mounted disks. The data files and LDM log files are volatile, so they should not be automatically backed up on a regular basis.

    a) **mkdir /usr/local/ldm/data**

    b) **mkdir /usr/local/ldm/logs**

22. The symbolic-link subdirectories allow for easy switching between different versions of an LDM installation. Each symbolic-link subdirectory goes through a "runtime" symbolic-link, which, in turn, references the currently-active LDM installation.

    a) **cd /usr/local/ldm**

    b) **ln -s ldm-6.4.6 runtime**

    c) **ln -s runtime/* .** (Note there is a space before the period.)

    The directory structure should now look something like this (the data and log "directories" might be different):

    ```
    ls -lF
    ... bin -> runtime/bin/
    ... data -> /var/data/ldm/data/
    ... etc/
    ... include -> runtime/include/
    ... ldm-6.4.6/
    ... lib -> runtime/lib/
    ... logs -> /var/data/ldm/logs/
    ... man -> runtime/man/
    ```

```
... runtime -> ldm-6.4.6/
... src -> runtime/src/
```

8. LDM configuration for VAHIRR

Reference: LDM Reference, http://www.unidata.ucar.edu/software/ldm/ldm-6.4.6/basics/index.html

1. Log in as the ORPG user. Copy the following files to the **`/usr/local/ldm/etc`** directory: **`ldmadmin-pl.conf`**, **`ldmd.conf`**, **`pqact.conf`**, **`scour.conf`**. Create a new directory: **`/usr/local/ldm/scripts`**. Copy the **`mvLev2.pl`** file to the new directory. Make sure that the ORPG user has full access to the **`mvLev2.pl`** file: read, write, and execute permissions. Request these files from the AMU: **`ldmadmin-pl.conf`**, **`ldmd.conf`**, **`pqact.conf`**, **`scour.conf`**, and **`mvLev2.pl`**. Make sure the **`mvLev2.pl`** file is executable by the ORPG user account.

2. Make any necessary modifications to the **`ldmadmin-pl.conf`** file. The Perl script for administering the LDM is installed as **`$LDMHOME/ldm-6.4.6/bin/ldmadmin`** (assuming LDM version 6.4.6). The configuration file for this script should have been installed as the file **`$LDMHOME/etc/ldmadmin-pl.conf`**. In the configuration file, set the hostname to the local machine. Set the $pq_size variable to 500M.

3. Create the product queue with the following command: **`ldmadmin mkqueue -f`**

   where the -f option requests "fast" creation. Enter **`pqcheck –v`** to check the product queue. If the product-queue is invalid, then recreate it with the following commands:

   ```
   ldmadmin delqueue
   ldmadmin mkqueue -f
   ```

   If the product-queue doesn't have a writer-counter, then create the counter and set it to zero with the following command:

   ```
   pqcheck -F
   ```

4. Start the LDM by entering: **`ldmadmin start`**. If LDM is already running, a message may be received similar to "`start_ldm: There is another server running, start aborted.`" To restart LDM, enter **`ldmadmin stop`**, then enter **`ldmadmin start`**

5. To verify that the LDM is working, enter: **`ldmadmin watch`**

6. If data does not appear to be arriving, check the appropriate data directories to see if data is current. The LDM server can be checked to see if LDM is running there. The LDM server at the AMU is amu-dataserver. There should also be data arriving and being stored at the amu-dataserver. The amu-dataserver requests LDM data from Marshall Space Flight Center. The logs in **`$LDMHOME/logs`** can be checked if there are problems with LDM starting up or receiving data. Also, check the running processes by entering **`ps –aux`**. The IP address of the ORPG-clone must be accepted by the LDM server. In the **`$LDMHOME/etc/ldmd.conf`** file on the LDM server, there is a regular expression that lists the accepted IP addresses. There needs to be a route between the LDM client and server – the client should be able to ping the server and the server should be able to ping the client. The two machines should be able to resolve the IP addresses of each other – one way of testing this is to see if a directory can be NFS mounted from the server onto the client, assuming of course that the NFS configuration is performed correctly.

7. To test the LDM radar data feed into the ORPG software:

   a) Make sure that the LDM is running on the ORPG-clone.

   b) Start up the ORPG program: **`mrpg –r –v startup`**

   c) Start up the read_ldm program:
   `read_ldm –d $LDMHOME/data/KMLB $ORPGDIR/ingest/resp.0`

   d) Use the –p option in read_ldm to prevent old radar files from being deleted in the source directory. Use the –u option in read_ldm if the input radar files are uncompressed. These two options are rarely used.

   e) Start up the CVG program. Verify that radar products are being created, by selecting Update List and

Filter on the main CVG screen. Verify that new products continue to be created. Verify that products can be viewed, by selecting the product on the main CVG screen, and then selecting the Database Product button. Verify that the ORPG software is processing products by starting the HCI application. The elevation should be updating in the HCI.

f) Stop the ORPG software:
`mrpg shutdown`
`mrpg cleanup`

9. VAHIRR configuration in CODE/ORPG software

1. Go to the VAHIRR source code directory: **`cd $HOME/src/cpc305`**.
2. Create a new directory inside that directory: **`mkdir tsk007`**.
3. Request the **`vahirr_operational.tar`** file from the AMU and put it in the **`$HOME/src/cpc305/tsk007 directory`**.
4. While in the **`$HOME/src/cpc305/tsk007`** directory, untar the tar file: **`tar xvf vahirr_operational.tar`**. This will overwrite some of the files in the directory.
5. Verify that the owner/group of each file in the directory is equal to the username and group of the ORPG user.
6. Here are the locations of where the untarred files need to be. Do not move them now:

```
$HOME/src/cpc305/tsk007 directory
 format_product.c
 input_parms.c
 makefile
 vahirr_driver.c
 vahirr.h
 vahirr.mak
 MD5SUMS

$HOME/bin/lnux_x86 directory
 vahirr_prod

$HOME/cfg directory
 product_generation_tables
 vahirr_parms.cfg

$HOME/cfg/extensions directory
 product_tables.sample_snippet
 task_table.sample_snippet

$HOME/data/ar2data directory
 BL_2007_07_05_19_08_22.bz2
 tst6.1_2007_07_11_14_23_47.bz2
```

7. Verify the checksums of all files in the tarball, except for the **`vahirr_parms.cfg`** and **`MD5SUMS`** files. Rename **`MD5SUMS`** to **`MD5SUMS.old`**.
8. Create an output file containing the md5sum values of all the files, except **`vahirr_parms.cfg`** and **`MD5SUMS`** (Note – this command is entered on one line):

```
md5sum -b BL_2007_07_05_19_08_22.bz2 format_products.c input_parms.c makefile
product_generation_tables product_tables.sample_snippet
task_table.sample_snippet tst6.1_2007_07_11_14_23_47.bz2 vahirr_driver.c
vahirr.h vahirr.mak vahirr_prod | tee MD5SUMS.new
```

9. Verify that the **`MD5SUMS.new`** and **`MD5SUMS.old`** files are equal.

10. Delete the **`vahirr_prod`** file, which is the VAHIRR executable file. This file is only included in the tarball to test its checksum value. The **`vahirr_prod`** file is recreated when the VAHIRR source code is compiled and installed.

11. Move the **`product_generation_tables`** and **`vahirr_parms.cfg`** files to the **`$HOME/cfg`** directory. If prompted, answer ‘y’ to overwrite the old file.

12. Move the **`product_tables.sample_snippet`** and **`task_table.sample_snippet`** files to the **`$HOME/cfg/extensions`** directory. If prompted, answer ‘y’ to overwrite the old file.

13. Move the **`BL_2007_07_05_19_08_22.bz2`** and **`tst6.1_2007_07_11_14_23_47.bz2`** files to the **`$HOME/data/ar2data`** directory. If prompted, answer ‘y’ to overwrite the old file. These two files were developed for the VAHIRR Factory Acceptance Test procedures. They can be used to test the correctness of the VAHIRR radar product, but are not needed for creating the VAHIRR radar product with real-time Archive II data.

14. While still in the **`$HOME/src/cpc305/tsk007`** directory, enter: **`make all`**

15. View the output from the compiler and verify that no errors occurred.

16. At the command prompt, enter: **`make install`**

17. Again, view the compiler output and verify that no errors occurred.

18. Verify that the **`vahirr_prod`** file was updated in the **`$HOME/bin/lnux_x86`** directory, by comparing the file’s modification time with the current time.

19. Update the amount of memory allocated per process, from 40 MB to 60 MB. This is to prevent the VAHIRR product from crashing with an out-of-memory error. Edit the **`task_table`** file in the **`$HOME/cfg`** directory. Near the end of the file is the “Common_process_attributes” section. Change the value to the right of mem_limit from 40 to 60.

20. Test the production of the VAHIRR product using real-time data:

    a) Verify that LDM is running: **`ps –aux`**. Start up the ldmadmin program: **`ldmadmin start`**.

    b) Start up the ORPG software: **`mrpg –r –v startup`**.

    c) Start up the read_ldm program in order to process incoming LDM radar data:
    `read_ldm –d $LDMHOME/data/KMLB $ORPGDIR/ingest/resp.0`.

21. If the **`mrpg`** process does not start up correctly, verify that no backup files are in the **`$HOME/cfg/extensions`** directory.

22. After a couple of minutes of issuing the **`read_ldm`** command, the terminal window should display lines like the following:

    ```
    File  /usr/local/ldm/data/KMLB/20061109022502.raw removed
    ```

23. Start HCI in a different terminal window and verify that the elevation angles are changing when a new KMLB volume scan arrives. If in clear-air mode, then angles may be changing slowly.

24. Start up CVG and verify that VAHIRR products (both Product ID and Product Code are 153) are being created and are viewable. When viewing VAHIRR for the first time in CVG, there will be a configuration error: “The selected product has not been configured. Enter preferences for this product using the CVG Product Specific Info Menu”. In this case, select Preferences… in the File menu. In the Preferences dialog box, select “Edit Product Specific Info…”. Select Product ID 23, which is the 1km resolution Composite Reflectivity. Write down the settings for product 23, because the same settings will be entered for VAHIRR. Select the Add Product button and enter 153. Change the settings for product 153. Next, modify the Unit of Measure to be “dBZ-kft”. In the entry box beside Unit of Measure, enter “dBZ-kft”. Select the Apply Edit button. Select the Save Changes button. Close the Product Specific Info Edit dialog box. Select OK in the Preferences dialog box. Test the new VAHIRR settings, by viewing a VAHIRR product in CVG.

25. To stop the ORPG software:

a) exit out of CVG and HCI

b) press [Ctrl]-[C] in the terminal window running read_ldm

c) enter: **`mrpg shutdown`**

d) enter: **`mrpg cleanup`**

26. Verify the correct operation of the VAHIRR code by performing the pre-test procedure (section 2) and the Baseline Test Case (section 3.1) of the VAHIRR FAT Procedures. The **`BL_2007_07_05_19_08_22.bz2`** file from step 13 will be needed. However, instead of setting the freezing level height in line two of the **`vahirr_parms.cfg`** file, set the freezing level height in the HCI application. Directions on how to change the freezing level (height of 0 degrees Celsius) in HCI are in section 11. In the operational version of VAHIRR, only line one of the **`vahirr_parms.cfg`** file is read, for the cone-of-silence height. Note that it is unnecessary to check the checksums again. After the test has been completed, stop the ORPG software (see step 25). The ingest of LDM data can be stopped by entering: **`ldmadmin stop`**

10. Modifying the size and products in the product database

References: NOAA/NWS Radar Operations Center and Mitretek (now called Noblis)

It is possible to control the maximum size of the product tables, as well as which radar products are generated. The maximum size of the database (i.e. the maximum number of messages that the product database can hold) should be set to at least 2000, up to a maximum of 8000. Do not set the maximum size of the database to above 8000 messages. If the maximum number of messages in the product database is reached and the maximum size is set to less than 2000, then old products will not get purged. The product database will just fill up and new products cannot be written to the database. If this occurs, then enter **`mrpg shutdown`**, then **`mrpg cleanup`**. Enter **`mrpg -r -v startup`**, to clear the database. Obviously, if products are removed from the **`product_generation_tables`** file, then the database fills up slower. The size in bytes per message in the database varies with how much precipitation is occurring (i.e. the coverage of precipitation across the radar coverage area).

1. Modifying the size of the product database:

   To change the size of the ORPG product database, the file **`$HOME/cfg/data_tables`** needs to be modified. In the file there should be an entry like this:

```
Datastore {
 data_id         8014 ORPGDAT_PRODUCTS
 path            pdist/product_data_base.lb
 persistent
 Lb_attr {
     remark      “ORPGDAT_PROD_DATA_BASE”
     msg_size    0
     maxn_msgs   8000
     types       “LB_MSG_POOL”
 }
}
```

   The size is specified by the maxn_msgs field, representing the number of messages (i.e. products) that the database can hold. To decrease the size, set maxn_msgs to a smaller number. Once changed, the ORPG will need to be restarted with **`mrpg –p startup`**. When starting up CVG, the terminal window will still output “CVG Product Data Base Size 8000”, because CVG has a different configuration file for the maximum database size.

2. Adding or removing products from being generated:
   The product generation tables are stored in a binary form in a file defined by an entry in the **`$HOME/cfg/data_tables`** file:

```
Datastore {
 data_id         4000 ORPGDAT_PROD_INFO
 path            pdist/prod_info.lb
 persistent
 mrpg_init
 Lb_attr {
     remark      “ORPGDAT_PROD_INFO”
     msg_size    0
     maxn_msgs   7
     types       “LB_REPLACE”
 }
}
```

To permanently remove products from generation, modify the **`~/cfg/product_generation_tables`** file, shutdown the ORPG (**`mrpg shutdown`**, **`mrpg cleanup`**), delete the above listed file (**`rm $ORPGDIR/pdist/prod_info.lb`**), then restart the ORPG. Some products will be generated by default to support the alerting function, so even removing them from the default generation won't completely disable them. It is strongly recommended that the **`product_tables`** file not be modified to control which products are generated.

11. Purging old LDM and Archive II radar files

If the LDM is saving radar data to the local hard drive, then that data should occasionally get purged. Old LDM log files can also be purged, but this is of lesser concern due to the small size and number of log files that are created.

As a default, the read_ldm program writes an Archive II radar data file for every volume scan to the following six directories:

```
    $LDMHOME/data/KAMX
    $LDMHOME/data/KBYX
    $LDMHOME/data/KJAX
    $LDMHOME/data/KMLB
    $LDMHOME/data/KTBW
    $LDMHOME/data/KTLH
```

A crontab entry can be added to run the LDM's scour script once a day. Here is an example cron entry that is run once a day at noon (note – environmental variables should not be used in the cron entries):

```
    0 12 * * * /usr/local/ldm/bin/scour >/dev/null 2>&1
```

The scour script uses the **`scour.conf`** file in the **`$LDMHOME/etc`** directory. Here are the contents of that file:

```
# Configuration file for "scour" utility, to delete all files older than a
# specified number of days from specified directories and all their
# subdirectories.  Scour should be invoked periodically by cron(8).
#
# Each line consists of a directory, a retention time (in days), and
# (optionally) a shell filename pattern for files to be deleted.  If no
# filename pattern is specified, "*" representing all files not beginning
# with "." is assumed.
#
# A hash in column one indicates a comment line.

# Directory         Days-old    Optional-filename-pattern

#~ldm/data/dir1     2
#~ldm/data/dir2     2       *.foo
#~ldm/logs          1       *.stats
$LDMHOME/data/KAMX  2
$LDMHOME/data/KBYX  2
$LDMHOME/data/KJAX  2
$LDMHOME/data/KTBW  2
```

```
$LDMHOME/data/KTLH  2
```

As long as the –p option is not used, the read_ldm command will purge Archive Level II radar data files in the **$LDMHOME/data/KMLB** directory on a regular basis. To avoid storing files in the KAMX, KBYX, KJAX, KTBW, and KTLH directories in the first place, the AMU has the following two entries in its **$LDMHOME/etc/pqact.conf** file:

```
NEXRAD2 ^L2-BZIP2/KMLB/(..............)/*
    FILE    data/KMLB/\1.tmp

NEXRAD2 ^L2-BZIP2/KMLB/(..............)/([0-9])+/([0-9])+/E
    EXEC    /usr/local/ldm/scripts/mvLev2.pl /usr/local/ldm/data/KMLB/\1.tmp
/usr/local/ldm/data/KMLB/\1.raw
```

12. Entering environmental data for the Hail algorithms

The VAHIRR algorithm calculates the cloud thickness at and above 0 degrees Celsius. VAHIRR gets the height of the freezing level from the Hail Index product. The Hail Index product in turn uses the height of 0 and -20 degrees Celsius from the radar's Environmental table. The Environmental table can be set in the HCI application. First of all, obtain the height (in mean sea-level (MSL)) of 0 and -20 degrees Celsius by viewing the most recent upper-air sounding nearest to the radar. For KMLB, use the XMR sounding, or the TBW sounding as a backup. Next, start the HCI application. On the RPG Control/Status screen, select the Products button underneath RPG. Select the Algorithms button. Select the Hail Detection item in the Adaptation Item list box. Scroll down to the Height (0 Deg Celsius) and Height (-20 Deg Celsius) rows. Click on the entry box by 0 Deg Celsius and enter the value, which is in kft. Press Enter. The Save button should then be highlighted. Click the Save button. Click Yes in the popup window. Close the Algorithms dialog box.

If the freezing level is set to an invalid value, then the VAHIRR product will not be created each volume scan until the freezing level is set to a valid value. In the **$ORPGDIR/logs** directory, the following error message will be in the **vahirr_prod.output** file (in the example, the freezing level was set to -0.10 kft MSL):

```
0C height parameter - -0.100 kft
ERROR: 0C hght -0.10 kft (msl) = -0.03 km is out of range (0.0 – 7.0 km)
```

Fortunately, the HCI application will not accept negative values for the height of 0 or -20 degrees Celsius. If the freezing level is set to 0.0 kft MSL, then the VAHIRR product will be created. However, all grid points are disqualified from reporting a VAHIRR value, since the lowest elevation scan (i.e. 0.5 degrees) must be at or below the freezing level. This is to avoid underestimating the actual VAHIRR value. If the freezing level is set to 0.0 kft MSL, all of the pixels in the VAHIRR product will show up as white (No Data). In addition, the **$ORPGDIR/logs/vahirr_prod.output** file will contain the following message:

```
NOT GOOD: radar height 0.12 kft (msl) is above 0C height 0.00 kft (msl) set 0C height
to 0.0 km (agl)
```

13. Modifying the height of the cone-of-silence

The VAHIRR product will report "No Data" for pixels that are in or within 5 km of the cone-of-silence. The height of the cone-of-silence is set by the **~/cfg/vahirr_parms.cfg** file. This file should contain one value – the cone-of-silence height in kft above ground level. If this file is missing, then the default cone-of-silence height of 65.616 kft is used. In addition, the default cone-of-silence height is used if the value in the **vahirr_parms.cfg** file is invalid. The cone-of-silence height is invalid if it is negative or above 65.616 kft.

14. Installing the test version of VAHIRR over a previous installation

The test version of VAHIRR is used to run the VAHIRR Factory Acceptance Test procedures. The only difference with the operational version, is that the test version sets the height of the freezing level with the value in the second line of the **vahirr_parms.cfg** file. In the operational version, only the first line of the **vahirr_parms.cfg** file is read. In the operational version, the height of the freezing level is set with the HCI application.

If a previous version of VAHIRR has been installed and configured, then the rest of the instructions in this guide do not have to be followed. In this case, do the following in order to install the test version of VAHIRR:

1. Log in as the ORPG user.
2. Request the **`vahirr_test.tar`** file from the AMU. See step 3 of section 8.
3. Put the **`vahirr_test.tar`** file into the **`$HOME/src/cpc305/tsk007`** directory.
4. Untar the **`vahirr_test.tar`** file (**`tar xvf vahirr_test.tar`**). This will overwrite some of the files in the directory.
5. Verify that the owner/group of each file in the directory is equal to the username and group of the ORPG user.
6. Verify the checksums of all files in the tarball, except for the **`vahirr_parms.cfg`** and **`MD5SUMS`** files. Rename **`MD5SUMS`** to **`MD5SUMS.old`**. Create an output file containing the md5sum values of all the files, except **`vahirr_parms.cfg`** and **`MD5SUMS`** (the command should be entered on one line):

```
md5sum -b BL_2007_07_05_19_08_22.bz2 format_products.c input_parms.c makefile
product_generation_tables product_tables.sample_snippet task_table.sample_snippet
tst6.1_2007_07_11_14_23_47.bz2 vahirr_driver.c vahirr.h vahirr.mak vahirr_prod |
tee MD5SUMS.new
```

7. Verify that the **`MD5SUMS.new`** and **`MD5SUMS.old`** files are equal.
8. Delete the **`vahirr_prod`** file, which is the VAHIRR executable. This file is only included in the tarball, to test its checksum value. The **`vahirr_prod`** file is recreated when the VAHIRR source code is compiled and installed.
9. While still in the **`$HOME/src/cpc305/tsk007`** directory, delete the **`product_generation_tables`** file, since it has not been updated. Move the **`vahirr_parms.cfg`** file to the **`$HOME/cfg`** directory. If prompted, answer ‘y’ to overwrite the old file.
10. Delete the **`product_tables.sample_snippet`** and **`task_table.sample_snippet`** files, since they have not been updated.
11. Move the **`BL_2007_07_05_19_08_22.bz2`** and **`tst6.1_2007_07_11_14_23_47.bz2`** files to the **`$HOME/data/ar2data`** directory. If prompted, answer ‘y’ to overwrite the old file.
12. At the command prompt, enter: **`make all`**
13. View the compiler output and verify that no errors occurred.
14. At the command prompt, enter: **`make install`**
15. Again, view the compiler output and verify that no errors occurred.
16. Verify that the **`vahirr_prod`** file was updated in the **`$HOME/bin/lnux_x86`** directory, by comparing the file’s modification time with the current time.
17. Perform steps 19 through 25 of section 8. Note that in step 23, the VAHIRR product’s legend is changed in the CVG application from “dBZ” to “dBZ-kft”.

15. Installing the operational version of VAHIRR over a previous installation

If a previous version of VAHIRR has been installed an configured, do not follow all of the instructions in this guide. In this case, do the following in order to install the operational version of VAHIRR:

1. Log in as the ORPG user.
2. Request the **`vahirr_operational.tar`** file from the AMU.
3. Put the **`vahirr_operational.tar`** file into the **`$HOME/src/cpc305/tsk007`** directory.
4. Untar the **`vahirr_operational.tar`** file (**`tar xvf vahirr_operational.tar`**). This will overwrite

some of the files in the directory.

5. Verify that the owner/group of each file in the directory is equal to the username and group of the ORPG user.
6. Verify the checksums of all files in the tarball, except for the **`vahirr_parms.cfg`** and **`MD5SUMS`** files. Rename **`MD5SUMS`** to **`MD5SUMS.old`**. Create an output file containing the md5sum values of all the files, except **`vahirr_parms.cfg`** and **`MD5SUMS`** (the command should be entered on one line):

```
md5sum -b BL_2007_07_05_19_08_22.bz2 format_products.c input_parms.c makefile
product_generation_tables product_tables.sample_snippet task_table.sample_snippet
tst6.1_2007_07_11_14_23_47.bz2 vahirr_driver.c vahirr.h vahirr.mak vahirr_prod |
tee MD5SUMS.new
```

7. Verify that the **`MD5SUMS.new`** and **`MD5SUMS.old`** files are equal.
8. Delete the **`vahirr_prod`** file, which is the VAHIRR executable. This file is only included in the tarball, to test its checksum value. The **`vahirr_prod`** file is recreated when the VAHIRR source code is compiled and installed.
9. While still in the **`$HOME/src/cpc305/tsk007`** directory, delete the **`product_generation_tables`** file, since it has not been updated. Move the **`vahirr_parms.cfg`** file to the **`$HOME/cfg`** directory. If prompted, answer ‘y’ to overwrite the old file.
10. Delete the **`product_tables.sample_snippet`** and **`task_table.sample_snippet`** files, since they have not been updated.
11. Move the **`BL_2007_07_05_19_08_22.bz2`** and **`tst6.1_2007_07_11_14_23_47.bz2`** files to the **`$HOME/data/ar2data`** directory. If prompted, answer ‘y’ to overwrite the old file.
12. At the command prompt, enter: **`make all`**
13. View the compiler output and verify that no errors occurred.
14. At the command prompt, enter: **`make install`**
15. Again, view the compiler output and verify that no errors occurred.
16. Verify that the **`vahirr_prod`** file was updated in the **`$HOME/bin/lnux_x86`** directory, by comparing the file’s modification time with the current time.
17. Perform steps 19 through 26 of section 8. Note that in step 23, the VAHIRR product’s legend is changed in the CVG application from “dBZ” to “dBZ-kft”.

16. Installing the AMU ORPG-clone Archiver Application

Purpose of the application

The ORPG-clone's product database is limited in size. After the maximum number of messages has been reached, the oldest radar products get purged. The maximum number of messages that can be stored in the database is 8000. If all of the default radar products are stored in the product database, the database will fill up within a few hours. After the database becomes full, then old products start to get purged. In addition, the entire product database will be cleared if the -r option is used when starting the ORPG-clone software with the **`mrpg`** command. Finally, the products can be lost if the database file is deleted or becomes corrupt. For example, animating radar products with the CVG application will usually corrupt the product database. The Archiver application will enable the user to archive radar products in near real-time. Radar products are only archived when the ORPG-software is creating new products with incoming Archive Level II radar data.

What the application does

The Archiver is a Tcl script. When the Archiver is run from a cron job, the script first reads the current time. Next, the script opens a log file for writing. The list of radar product codes is then read from the script. For each archived radar product, the script determines the message id of the most recent version of the product, using the **`prod_query`** ORPG utility that is run from the command-line. If the **`prod_query`** application returns a valid message id, then the radar product is extracted from the product database using the **cvt** ORPG utility. Individual radar products are stored in the file system.

Installation of application
Follow these steps to install the application:

1. Log into the ORPG account on the ORPG-clone. Open up a terminal window. Verify that the date/time is accurate within one minute. The root user must normally be logged in to change the date/time.
2. Create the **$HOME/scripts** directory on the ORPG-clone if it does not already exist. If it does not, enter **cd $HOME** at the prompt. Then enter **mkdir scripts** at the prompt.
3. Create the **$HOME/cronLogs** directory on the ORPG-clone if it does not already exist. If it does not, enter **cd $HOME** at the prompt. Then enter **mkdir cronLogs** at the prompt.
4. Request the **archive_amu.exp** file from the AMU and copy it to the **$HOME/scripts** directory.
5. Determine which radar products to archive. The default list includes: VAHIRR, ET (Echo Tops), VIL, and CZ (Composite Reflectivity). For each product to archive, look up the product code and product ID using the CVG application. In the main page of the CVG application, the product code is in the column just to the left of the Product Description. The product ID is just to the left of the abbreviated product name. For example, the product code for Composite Reflectivity is 37, the product ID is 23, and the abbreviated name is CR. Create subdirectories for the archived products in the **$HOME/data** directory. The subdirectory names should be based on the product codes. For example if products 153 and 37 are archived, then the following directories should be created: **$HOME/data/153** and **$HOME/data/37**.
6. Edit the **archive_amu.exp** file to change from the default list of products. Edit the line that contains **set prodList {153 41 57 37}**. The numbers inside the braces represent the product code for each product to archive. For example, to archive Composite Reflectivity and VAHIRR products, then the line should contain **set prodList {153 37}**.
7. Determine how often to archive products. Select a time interval less than the time it takes the radar to perform one volume scan. For example, in VCP11, it takes the radar around five minutes to perform a volume scan. Therefore, archive products every four minutes or less.
8. Create the cron entry that will run the Archiver every x minutes, where x is how often to run the Archiver. Enter **crontab -e** at a prompt. An editor will open up the crontab file, which contains the ORPG user's cron jobs. The default editor is usually **vi** - if so enter **i** to go into insert text mode and press **[Esc]** to exit out of insert mode. In **vi**, enter **:q!** to exit without saving the file. Enter **:wq!** to exit and save the file. Enter **[Esc]** before trying to exit the editor, to exit out of text mode. Assuming that x is 4, enter the following line in the editor:

   ```
    0-59/4 * * * * csh -c '/home/codev8/scripts/archive_amu.exp' >
   /home/codev8/cronLogs/archive_amu.crontxt 2>&1
   ```

   This assumes that the ORPG home directory is **/home/codev8**. If it is not, then set it to the **$HOME** value. Avoid using environmental variables such as **$HOME** in the crontab entries. Enter **echo $HOME** at a command prompt to get the ORPG user's home directory. Comment out the cron job until the job is ready to run; enter a "#" character at the beginning of the line.
9. Create the cron entry that will delete old log files created by the Archiver. This entry should be run at a frequency less than the above cron entry. For example, if the cron entry to run the Archiver runs every four minutes, then the cron entry to delete old log files could be run every five minutes. Assuming that the cron is run every five minutes, enter the following line in the editor:

   ```
    0,5,10,15,20,25,30,35,40,45,50,55 * * * * rm
   /home/codev8/cronLogs/archiveprod_amu*.log
   ```

   This assumes that the ORPG home directory is **/home/codev8**. Comment out the cron job until the job is ready to run; enter a "#" character at the beginning of the line.

17. User Guide for the AMU ORPG-clone Archiver Application

Starting the Archiver

1. Start up the LDM, ORPG, and read_ldm programs.
2. Edit the cron entry so that the Archiver is run on a regular schedule. At a command prompt, enter **`crontab -e`** to edit the cron jobs. Either uncomment the cron entry that runs **`archive_amu.exp`**, or create/modify the cron entry. Uncomment an entry by removing the "#" character at the beginning of the line.
3. Edit the cron entry that deletes old log files on a regular schedule. Either uncomment the cron entry that deletes old files, or create/modify the cron entry. Uncomment an entry by removing the "#" character at the beginning of the line.

Viewing log files

Each time the Archiver is run, there is a log file called **`archive_amu.crontxt`** saved in the **`$HOME/cronLogs`** directory. This file indicates whether the Archiver ran successfully. The log file also contains the file name of a secondary log file. An example of a primary log file is (secondary log file shown in bold font):

```
Starting archive_amu.exp at Apr 20 2007 23:00:01 Z
Output of archive_amu.exp will go to
/home/codev8/cronLog/archive_amu.20070420_2300.log
Ending archive_amu.exp normally at April 20 2007 23:00:01 Z
Ending archive_amu.exp normally at April 20 2007 23:00:01 Z
Ending archive_amu.exp normally at April 20 2007 23:00:01 Z
Ending archive_amu.exp normally at April 20 2007 23:00:01 Z
```

In the primary log file, the statement "Ending archive_amu.exp normally at April 20 2007 23:00:01 Z" is repeated four times because there are four different radar products that are being archived. The secondary log filename contains the date/time when the radar products were generated. The secondary log file is also saved in the **`$HOME/cronLogs`** directory. The secondary log file contains detailed information about each radar product that was archived by the Archiver.

Viewing archived radar products

Open up a terminal window and enter **`cvg`** at the command prompt. Select the Product Disk Filter button (under Alternate Data Source) on the CODEview Graphics main screen. The Choose Product Binary File dialog box should open. Navigate to the **`/home/codev8/data/`** directory (assuming that **`/home/codev8`** is the ORPG's home directory). There will be a subdirectory for the product code of each radar product that is being archived. With the default configuration of the Archiver, products 153 (VAHIRR), 37 (Composite Reflectivity at 1 km resolution), 41 (Echo Top), and 57 (VIL) are archived. To view the archived VAHIRR products, for example, navigate to the "153" subdirectory. In the subdirectory, the archived files will be listed on the right side of the dialog box. The filenames are based on the time of the volume scan in which the products were generated. Select a file and make sure that the Input Type is set to "ICD Product (no hdr)". Next, select the OK button. The Specify Product ID dialog box should open up. Enter the Product ID of the product, not the Product Code. The Product Packets dialog box should then open up. Select the OK (Select All) button. The Graphic Product Display screen should appear with the archived radar product.

Stopping the Archiver

To stop the Archiver, remove the Archiver cron job from the cron scheduler. Also, remove the cron job that removes old Archiver log files. Open a terminal window. Enter **`crontab –e`** at the command prompt. An editor will start up with the file containing the scheduled cron jobs. Comment out the two lines that contain the cron jobs that run the Archiver and remove old log files. Comment out lines by inserting a "#" character at the beginning of the line. In the editor, commented out lines should look like the following (assuming the ORPG's home directory is **`/home/codev8`**):

```
#0-59/4 * * * * csh –c '/home/codev8/scripts/archive_amu.exp' >
/home/codev8/cronLogs/archive_amu.crontxt 2>&1
#0,5,10,15,20,25,30,35,40,45,50,55 * * * * rm
```

```
/home/codev8/cronLogs/archiveprod_amu*.log
```

After editing the cron jobs, save the file and exit out of the editor. To verify the cron jobs were edited correctly, enter **`crontab –l`** at the command prompt.