The Common Operations and Development Environment (CODE) for the WSR-88D Open RPG

CODE B18.0r1.11: April 2018

Includes ORPG Build 18.0r1.11

Volume 1. Setting Up the ORPG Development Environment

- Installing and Compiling ORPG Source Code and CODE Utilities-

The **U.S. Government Edition** of CODE is the complete version. Distribution is limited to within the United States Government.

The **Public Edition** of CODE is intended for public release. Certain Copyrighted material has been removed to permit release outside the U.S. Government.

CODE provides:

- Instructions for setting up the development environment (includes ORPG source code)
- Guidance for compiling software and configuring new ORPG tasks & products
- Instructions for definition and use of algorithm adaptation data and algorithm dependent parameters
- API Programming Guide and the structure of WSR-88D algorithms (with sample algorithms)
- WSR-88D specific analysis tools
- A set of WSR-88D Archive II Data files and other special test case data.

CODE User provides:

- An Intel PC with Red Hat Enterprise Workstation or CentOS desktop

Volume 1. Guide to Setting Up the Development Environment

> CODE is produced in two versions:
>
> 1. **National Weather Service Edition** - This is the complete version of CODE. Distribution is limited to within the National Weather Service and other U.S. Government Agencies.
> 2. **Public Edition** - This version of CODE is intended for public release. Certain proprietary software components have been removed to permit release outside the U.S. Government.
> 3. Differences between the two CODE editions are described in Appendix K of this Volume.

Introduction

This version of CODE contains the source code for a September 2017 build of the ORPG software: Build 18.0r1.11.

Document 1. CODE Specific ORPG Installation Instructions

These instructions provide the basic information to configure the development environment including the installation of the ORPG source code and supplementary tools. Basic procedures for running the ORPG are also provided.

Document 2. Install CODE Software

This document includes instructions for the installation of the CODE development and analysis utilities and CODE sample algorithms.

Appendices

Other Documentation

HCI Guide - The NWS ROC maintains an on-line guide to the Human Computer Interface (HCI). `http://www.osf.noaa.gov/ops/epss_rpg.asp`

Archive II Disk Data Files - The CODE ORPG installation includes three volumes of Archive II data in the form of individual disk files in order to provide an immediate source of input data to test the ORPG. The CODE CD includes additional Archive II disk files in the `ar2data` directory.

Volume 1. Guide to Setting Up the Development Environment

Document 1. CODE Specific ORPG Installation Instructions

These instructions provide the basic information to configure the development environment including the installation of the ORPG source code and supplementary tools. The ORPG is designed to be installed into a standard Unix account.

Section I   **Preparation for Installation**

Section II   **Installation Instructions**

Section III   **Supplemental Information**

Section IV   **Running the ORPG**

Vol 1. Document 1 - CODE Specific ORPG Installation Instructions

Section I Preparation for Installation

WHAT'S NEW for Build 18

Although Red Hat 7 (64 bit) is the official operating system for ORPG, CentOS 7 (64 bit), which is free, has been tested for CODE installation. CentOS is an Enterprise-class Linux Distribution derived from sources freely provided to the public by a prominent North American Enterprise Linux vendor. The installation of CentOS 7 system is quite similar to the Red Hat 7 system.

Red Hat Enterprise 7 Workstation

These instructions are for preparing an Intel PC having Red Hat Enterprise 7 *Desktop with Workstation Option* as the operating system for use in the CODE environment. And can also be used in the similar way to prepare the CentOS 7 system if CODE users do not have Red Hat 7.

The installation script **install_rpg** (or any operational installation script included with the ORPG source code) should **not** be used. First, the scripts do not accomplish all configuration steps required for a development environment. Second, in most cases, some actions taken by these scripts must be accomplished in a different manner in order to accommodate your local environment. Third, some actions taken may not be appropriate for workstations that also serve users not involved with ORPG development.

Introduction

These instructions provide the basic information to configure the development environment including the installation of the ORPG source code and supplementary tools. The ORPG is designed to be installed into a standard Unix account.

> **Some experience using a Linux environment is assumed. Some aspects of installation and configuration require basic system administration knowledge.**

Note: All procedures should be accomplished while logged into the account into which the ORPG is being installed unless indicated otherwise. Steps requiring administrative privileges are flagged with **SA**

System Administration Summary

A knowledgeable system administrator is needed to assess whether the system prerequisites are met:

- If installing Red Hat Enterprise Desktop from scratch, be sure to use the 64-bit releases. It is recommended that the latest ISO CD or DVD images be obtained from the Red Hat web site.
- Determining whether the necessary Linux packages have been installed and updating the Red Hat installation if they have not.
- After installation, future updates can be installed automatically or manually.
    - If Red Hat Network is not used to automatically install updates;
        - Updates can be installed manually from the Red Hat Network web site, or
        - The latest ISO CD or DVD images can be downloaded from the Red Hat web site. Installation CDs and DVDs can be burned from these images and used to add additional packages and package updates.
    - If the workstations are going to be registered with the Red Hat Network for automatic updates, it is recommended that all required packages be installed before registration with Red Hat Network.
- Configure the disk storage file systems to provide necessary swap space and room for ORPG development accounts.

The following instructions include actions requiring administrative privileges:

- Installing any required software packages that are not included with the Red Hat Linux distribution.
- Creating the user account into which the ORPG will be installed and in which the ORPG will run.
- The workstation must be set up on a functional TCP/IP network with a static IP address. The ORPG is not compatible with dynamic address with DHCP.

In addition to the operating system files, software development packages and patches, the account into which the ORPG is being installed should have a minimum of 1 GB of space available.

Special Considerations

ORPG Previously Installed

It is recommended that the ORPG not be installed over a prior installation. The ORPG can be installed in more than one account. With special configuration and sufficient memory, more than one ORPG can be running simultaneously on a single workstation.

Do not reuse configuration files from the prior version of the ORPG. These files include: `.cshrc`, `orpg_env_cshrc`, `.bash_profile`, `orpg_env_profile`, `make_rpg`, `make.lnux_x86`, `make.common`, `task_tables`, etc. There may be subtle changes in these files that the instructions do not cover.

Installing more than one ORPG on a Workstation

More than one ORPG can be installed on a single workstation. There are several reasons for having multiple ORPG development environments. When upgrading to a new ORPG, development work can continue on the previous version until the new account is ready. Disk space permitting, it can be easier for more than one developer to share a workstation if not working on the same ORPG (useful if the development efforts are not related or not coordinated). Finally, it could be useful to have different versions of a single project installed at the same time.

When installing more than one ORPG on a single workstation keep the following in mind:

- Each ORPG must be installed in it's own account
- The "installation" script provided with the ORPG must NOT be used. Only scripts supplied with CODE should be used.
- In order to have more than one instance of an ORPG running simultaneously on a single workstation, each must have a different value for the variable **RMTPORT** (see *Editing the customized ORPG Account Files* in Part C of Volume 1, Document 1, Section II).
- ORPG shared memory segments for two data buffers may consume resources. See note below.

> **NOTE:** Four primary ORPG data buffers are configured to utilize individual shared memory segments that total over 50 MB in size. These memory segments are not released when the ORPG is shut down. The ORPG normally reuses the same allocated memory segments when restarted**. Each installed ORPG allocates its own memory segments. With multiple ORPGs installed on a single platform this could eventually consume sufficient memory space to prevent ORPG launch and development activities. Currently allocated segments can be listed with the **ipcs -a** command. Rebooting the operating system frees the memory.
>
> ** Sometimes the ORPG does not reuse these allocated shared memory segments when restarted. One observed cause of this is an ORPG start that is aborted. This can lead to loss of memory even with only one ORPG installed on a workstation.

Prerequisite Actions by System Administrator

1. Ensure the development workstation is properly configured 🆂🅰

- Properly configured Intel PC Workstation (see Appendix A CODE System Requirements).
- Installation of Red Hat Linux with all development utilities and desktop environments. Appendix B contains guidance in selecting software packages during installation.
- Appendix C contains guidance of installing CentOS 7 if CODE users would like to use the free OS system instead of Red Hat 7.

> **Red Hat Linux Pre-Installed**
>
> If Red Hat Enterprise is pre-installed on the platform:

o If you wish to partition the disk beyond the single "/" file system usually provided by the vendor, see 'Disk Partitioning' under ADVANCED INSTALLATION for guidance.

o With Red Hat Enterprise 7, the System -> Add/Remove Software tool does not provide the same interface as the installation program. **The only easy way to ensure all required packages are installed is to reinstall Red Hat enterprise 7 from distribution media using the guidance in this Volume.**

o Red Hat 4 is no longer supported since Build 12.

o When finished with the installation, continue with ' Software Not Installed during Red Hat Installation ', item 2 below.

Guidance for the Installation of Red Hat Linux

Red Hat 5 or 7 should be installed.

Before Installing Red Hat Linux

o Become familiar with the contents of the Red Hat *Installation Guide* relating to installation on an Intel PC.
o Red Hat Enterprise Linux 7 should be compatible with most hardware in systems that were factory built during the last two years. It is recommended that a list of hardware components present on the workstation (e.g., graphics display card) be compiled in case problems occur.
o When finished with the installation, continue with ' Software Not Installed during Red Hat Installation ', item 2 below.

Keep the following in mind when accomplishing the procedures in Part 1 of the Red Hat *Installation Guide*.

BASIC INSTALLATION: With limited experience in administration of Linux workstations, the following guidance will make installation relatively easy.

At the beginning of the installation process, recommend selecting the `Graphical` login type rather than the `Text` login. This causes the X-windows server to be automatically started when Linux is started.

Beginning with Red Hat 5, an 'installation number' obtained from Red Hat is required to get a successful installation. The installation number must reflect a license for Red Hat Desktop with workstation option. Red Hat is rather vague in describing the results of not supplying an installation number when prompted during installation. One result is that not

all software packages required for CODE will be installed.

Disk Partitioning -

Option 1: The basic installation creates a single file system where everything except the `/boot` and `swap` partitions is under the root '`/`'partition.

a. The easiest installation is to not share the PC with a Microsoft Windows installation or another Linux installation. To accomplish this
   ▪ Select all available drives for installation
   ▪ Select the option:
   'Remove all partitions on selected drives and create default layout'

b. If sharing the PC with a Microsoft Windows installation (not recommended for CODE):
   ▪ Use a second physical hard disk dedicated to Linux.
   ▪ In some situations it may be easier to not install a boot loader. A Linux boot floppy must then be created.

Make sure [x] Review and modify partitioning layout is checked so you can confirm the size of the swap space allocated. The sizes of the partitions are then displayed.

○ A '`/boot`' partition is created (approx. 100 MB),
○ `swap` space is allocated (confirm at least 1024 MB in size), and
○ the rest of space is placed into a single partition with a '/' mount point.

Option 2: If desired the definition and size of disk partitions can be customized. This is an advanced installation topic and should not be considered unless highly experienced in Linux administration.

Boot Loader -

The default Grub boot loader is satisfactory for most installations.

Network Configuration -

Note: The ORPG is not compatible with DHCP. The network interface can be configured at this time or after Red Hat installation. If configured during installation:

○ Make sure that [x] Active is checked for the primary interface `eth0`.

○ With the primary interface `eth0` selected, click "Edit" to modify device configuration
   ▪ Deselect "( ) Dynamic IP configuration (DHCP)" and select "(*) Manual configuration".
   ▪ Enter the workstation IP address and netmask (normally 255.255.255.0).

click OK when finished.

o Select "Set the hostname (*)manually" and enter the TCP/IP hostname.

o Enter the Gateway address and at least one DNS address.

Time Setup -

After setting the time zone, recommend deselecting system use UTC time.

Package Selection -

The default package list is **not** sufficient for the ORPG development environment.

Red Hat Enterprise 5

- When presented with Package Defaults, select "(*)Customize now". This is required in order to get all necessary packages installed.
- **The procedures in** Appendix B **must be followed when selecting packages to be installed.**

CentOS 5

- When presented with Package Defaults, select "(*)Customize now". This is required in order to get all necessary packages installed.
- **The procedures in** Appendix C **must be followed when selecting packages to be installed.**

After Software Installation and First Reboot -

When setting the date and time, do NOT check [ ] Enable Network Time Protocol. If there is a problem with the network configuration the next reboot can be significantly delayed.

The Red Hat firewall can be a little quirky to configure. Consider selecting Firewall: "Disabled" if you are on a protected network and this does not conflict with your organization's security procedures. The firewall can be enabled after installation.

When prompted, decline the offer for Red Hat Network Setup and Registration. If desired this can be accomplished later. When installing from media, it is best to ensure all required packages are installed with a successful compile and ORPG launch before obtaining automatic updates from the Red Hat Network.

2. Software Not Installed during the OS Installation SA

There are a few packages needed by CODE that are not installed during installation of Red Hat or CentOS. These packages cannot be individually selected by the package selection utility. Package `giflib-devel` is used by `CVG.` Package `ncompress` (contains Unix command compress and uncompress) is required by utility play_a2 when playing back .Z files. Packages `tcl-devel`, `tk-devel`, `gsl` and `gsl-devel` are required for the MIGFA algorithm.

To verify if these packages are installed, execute:

```
rpm -q giflib-devel
rpm -q ncompress
```

The following are required only for the NWS Edition of CODE (MIGFA algorithm):
```
rpm -q tcl-devel
rpm -q tk-devel
rpm -q gsl
rpm -q gsl-devel
```

If these packages are not installed, use yum to install them:

```
yum -y install giflib-devel
yum -y install ncompress
```

The following are required only for the NWS Edition of CODE (MIGFA algorithm):

```
yum -y install tcl-devel
yum -y install tk-devel
yum -y install gsl
yum -y install gsl-devel
```

After install all packages, it is better to install all available updates from Red Hat or CentOS.

```
yum -y update
```

3. Post Installation Red Hat Configuration Hints

- The GNOME desktop environment is the default if it has been installed. The GNOME desktop is used operationally but either GNOME or the KDE desktop can be used for CODE.
    - o To change from GNOME to KDE, prior to logging on select >Session at the bottom of the graphical login screen. Then select (*)KDE in the resulting pop-up menu and click on the Change Session button. Typically this must only be accomplished once for each user account.

- **SA** If you wish to remotely login to this workstation via XDM, it must be activated. XDM can be activated using the Logon Screen tool.

    KDE: Administration -> Login Screen
    GNOME: System -> Administration -> Login Screen

    In the Remote Tab, change the option from "Remote login disabled" to "Same as local".

- **SA** Many network communication services, for example **httpd** (web server), **rsh**, **rexec**, **telnet**, **nfs**, and the ftp server **gssftp** are not activated by default. On the other hand, some services are activated even if the software is not installed / configured. **For example, sendmail and snmp can prolong system boot if they are activated but the appropriate software was not installed / configured.**
    - These services can be activated or stopped using the Services menu.
        KDE: Administration -> Server Settings -> Services
        GNOME: System -> Administration -> Server Settings -> Services
    - Note that **xinetd** must also be activated for many of these services. Don't forget to select File -> Save Changes when finished.

- **SA** Even though DHCP is not selected during installation, the network configuration for TCP/IP is usually not complete. The ORPG will not launch with certain errors in the network configuration. The network device (ethernet card) can be configured using the Administration -> Network menu by selecting the desired device and pressing Edit.
    - A common configuration error resulting via the Red Hat administrative tools involves the `/etc/hosts` file. The tools place the host name on the line containing the `localhost` entry rather than on a separate line with the designated IP address.
    - If unfamiliar with Linux network configuration, the sample network configuration can be found in Appendix D Linux Network Configuration Files. Be sure to alias the hostname to rpg so that you do not need to modify .rssd.conf later.

- **SA** The firewall can be configured using the Administration -> Security Level and Firewall tool (from the KDE desktop).

4. Create the user account for each ORPG installation **SA**

The user account created for each ORPG installation must have a minimum of 1GB of space available in the home directory. The default shell for these accounts should be the C shell. In the future, bash will be supported.

Platform Preparation Complete - Next Steps

This completes preparation for ORPG installation. Proceed with the instructions in Volume 1, Document 1, Section II for installing an ORPG into a CODE account.

Vol 1. Document 1 - CODE Specific ORPG Installation Instructions

Section II Installation Instructions

Part A. Introduction

This document provides the basic information to configure the development environment on the **Linux PC** Platform.

Parts B - D of this document cover installation procedures for an ORPG in the CODE environment.

Section III contains supplemental information that is not required for installing a basic CODE development environment.

> **Some experience using a Unix environment is assumed. Some aspects of installation and configuration require basic system administration knowledge.**

Note:

- All procedures should be accomplished while logged into the account into which the ORPG is being installed unless indicated otherwise.
- Steps requiring administrative privileges are flagged with **SA**

The following procedures assume that the ORPG is installed directly into the account home directory.

Part B. Initial Steps

Special Consideration - ORPG Previously Installed

It is recommended that the ORPG not be installed over a prior installation. The ORPG can be installed in more than one account (assuming the installation script provided with the ORPG was not used). In order to have more than one instance of an ORPG running simultaneously on a single workstation, each must have a different value for the variable `RMTPORT` (see Editing the customized ORPG Account Files in `Part C` of `Volume 1, Document 1, Section II`).

Do NOT reuse any configuration files from the prior version of the ORPG. These files include: `.cshrc`, `orpg_env_cshrc`, `build_env_cshrc`, `make_rpg`, `make.lnux_x86`, `make.common`, `task_tables`, etc. There may be subtle changes in these files that the installation instructions do not cover.

Confirm System Admin Steps are Complete

Before proceeding, insure all system administration prerequisites described in Section I of this document have been completed.

1. Determine the user account and the account home directory provided for this ORPG installation.

    - **SA** The ORPG Installation Account
      The user account created for the ORPG installation should have a minimum of 1 GB in the home directory. The account default shell must be the C shell. Other shells will be supported in the future.

Installing the ORPG Source Code Software Distribution Files

The following procedures are based on a C shell environment. In the future, when other shells are supported, the appropriate changes must be made to the corresponding configuration files (i.e., `.bash_profile`, `.bashrc`, `.build_env_profile`, *and* `.orpg_env_profile`*).*

1. Obtain the ORPG source code distribution archive file (included with the CODE ORPG Software Distribution CD in directory `==/files_orpg_sw/`) and **place into the account home directory**.

2. Uncompress and extract the archive:

    If you have the NWS Edition:

    ```
    tar xvzf rpg_b18_0r1_3_nws_src.tgz
    ```

    If you have the Public Edition:

    ```
    tar xvzf rpg_b18_0r1_3_pub_src.tgz
    ```

Part C. ORPG Account Configuration Procedures (C shell)

The following procedures are based on a C shell environment. In the future, when other shells are supported, the appropriate changes must be made to the corresponding configuration files (i.e., `.bash_profile`, `.bashrc`, `.build_env_profile`, *and* `.orpg_env_profile`*).*

> This scheme of configuring the ORPG is equivalent to but departs somewhat from the method used for an operational ORPG. The reason is to provide more flexibility in establishing the ORPG / WSR-88D CODE environment. A generic

> `.cshrc` file is provided with comments documenting the contents and providing guidance on allowed changes.
>
> Rather than extensively editing the `.cshrc` file that is delivered with the ORPG software, we have consolidated the environment required to run the ORPG (`path`, `MANPATH`, `LD_LIBRARY_PATH`, and other environmental variables) into an ORPG *run environment file*. We also localize changes needed to build the entire ORPG software and to compile subsets of the software in a software *build environment file*. These files are sourced at the end of `.cshrc`.

Extracting the Archive Containing the CODE Customization Files

1. Obtain the `code_config_b18_0r1_3.tgz` file (included with the CODE distribution CD in directory `==/config_files/`) and place into the account home directory.
2. While in home directory extract the archive with the command:

   ```
   tar xvzf code_config_b18_0r1_3.tgz
   ```

   the replacement files and corresponding installation scripts are extracted into a subdirectory **code_config_b18_0r1_3**.

Installing the customized ORPG environment files

Several files must be customized in order to set up the account as an ORPG development environment. Some files set up the basic account environment and others modify makefiles. These files are installed **after** the ORPG source code archive has been extracted and **before** the ORPG is compiled.

1. While in the subdirectory `$HOME/code_config_b18_0r1_3/env`, install the development environment configuration files with the following command:

   ```
   ./inst_env_config
   ```

   - A list of the installed files is in Part A of Section III of this Document.
   - The files being replaced are saved with a `.OLD` extension.
   - A copy of the new file is made with a `.CODE` extension.

2. Ignore the files in the directory `~/code_config_b18_0r1_3/orpg` for now. They will be installed **after** the ORPG is compiled.

Editing the customized ORPG Account Files

Only those files that require additional manual edits are listed here. For a description of all the customized files and guidance on additional account configuration topics see Part A of Section III of this Document. Caution must be used when modifying the basic Unix account variables that are set in the CODE customized files (`.chsrc`, etc.).

Warning: After modifying any of the installed account configuration files be sure to make backup copies.

1. **The ORPG run environment file: `orpg_env_cshrc`**

   This file is sourced at the end of `.cshrc` and sets the environment for running the ORPG.

   If more than one installed ORPG is going to be run at the same time on a single workstation, the value of the variable `RMTPORT` must differ. Manually change the defined value of `RMTPORT` on each account. It is recommended that the first account have a value of 50000, the second 51000, etc.

   Create a backup copy of the modified `orpg_env_cshrc` file

2. **The software build environment file: `build_env_cshrc`**

   No changes required on the Linux platform. This file is sourced at the end of `.cshrc` and sets the environment for compiling the ORPG software and for compiling subsets of the software from within the source code tree.

   If you make any changes, create a backup copy of the modified `build_env_cshrc` file

Editing the customized ORPG Makefiles

Only those files that require additional manual edits are listed here. For a description of all the customized files and guidance on additional account configuration topics see Part A of Section III of this Document. There is no reason for additional customization of the ORPG makefile system.

Final Steps

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

2. At the command line, run `env` and check the output to verify that both the run environment file and the build environment file are being sourced at login.

Part D. Building (compiling) the ORPG

Replacing selected ORPG source code files

Currently there are no issues that require patches prior to compiling the ORPG.

Quick Test Compile

Before attempting to compile the whole ORPG, a quick test compile of a portion of the source code should be conducted (less than 1 minute). This does not catch all possible configuration errors. However, if the correct language compilers are being used with a properly installed operating system, an error free compile of the libraries in `~/src/cpc100` should indicate a properly configured environment.

1. While in the `$HOME` directory, execute the `test_make_cpc100` script with the following command if using `csh`:

   ```
   test_make_cpc100 $HOME >& <your output filename>
   ```

2. After compilation is complete, compare the created log file with a typical output file to ensure that the build was successful (see Appendix E CPC100 Compile Problems). This level of compiler warning message is normal. **Ensure there are no Error messages.**

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.

Compiling the ORPG source code

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 on a typical Pentium 4 PC.

IMPORTANT: Compiling the ORPG installs numerous configuration files. Unless backup copies of modified files are made, any changes or customization made will be lost if the ORPG is subsequently compiled again. This includes the `$HOME/.rssd.conf` file and other files in the home directory. Modifications to any file in the `$HOME/cfg` directory will be lost (e.g., `task_tables`, `comms_link.conf`, etc.).

1. While in the `$HOME` directory, execute the `make_rpg` script with the following command (using `csh`):

   ```
   make_rpg $HOME >& <your output filename>
   ```

2. After compilation is complete, compare the created log file with a typical output file to ensure that the build was successful (see Appendix F ORPG Compile Problems). This level of compiler warning message is normal. **Ensure there are no unexpected Error messages.**

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 following paragraphs, *Replacing missing ORPG binary files* and *Configuring the ORPG Installation* should be accomplished even if the ORPG

has been previously successfully built in this account. This is the easiest way to ensure the configuration is complete.

Replacing missing ORPG binary files

Currently there are no issues that require post build patches.

Configuring the ORPG Installation

These instructions provide the basic information to configure a standalone workstation running the ORPG with no communication managers (no wideband connection to an RDA or narrowband connections for product distribution). The configuration is based upon specific modifications to standard configuration files. More comprehensive configuration procedures for adding additional algorithm tasks and product data stores are provided in *CODE Guide Volume 2 - ORPG Application Development Guide*.

Warning: Many of the configuration files in `$HOME/cfg` are copied or otherwise derived from default versions located in `$HOME/src/cpc104`. Care should be taken to preserve copies of any manual changes to files in `$HOME/cfg` because they may be overwritten when the system is rebuilt or an installation script is run.

Installing the customized ORPG configuration files

1. While in the directory `$HOME/code_config_b18_0r1_3/orpg`, install the ORPG configuration files with the following command:

    ```
    inst_orpg_config
    ```

    The script prompts the user for the type of CODE distribution being installed (NWS or Public). This determines which version of the `task_tables` file is installed. If the wrong choice is made it can easily be corrected. See item 2 below.

    - A list of the installed files are in Part B of Section III of this Document.
    - 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/ar2data`

Editing the customized ORPG configuration files

Only those files that require additional manual edits are listed here. For a description of all the customized files and additional guidance on ORPG configuration topics see Part B of Section III of this Document. The only reason additional edits might be required would be if external interfaces for a base data source not normally used for algorithm development or a product distribution interface were desired.

1. **The Remote System Services configuration file: `$HOME/.rssd.conf`**

   Modification might be required. If the hostname has been aliased to `rpg`, there is no need to change this file. Otherwise variable `Client` needs to point to the hostname or `<ip_address>`. Open .rssd.conf from your `$HOME` directory with the editor of your choice. Modify the `Client` variable to be the `<ip_address>` of your machine.

   ```
   # RPG Development Workstations
   Client: rpg
   ```

2. **The `$HOME/cfg/task_tables` configuration file**

Two versions of the task_tables configuration file are installed: `task_tables.nws_code` and `task_tables.public_code`. Several tasks have been commented out in both versions: `cm_ping`, `wbserver`, `wbserver_ingest`, `convert_ldm`, `rpc.ldmd`, `manage_ldm`, `ldm_recomb`, `levelII_status_ICAO_ldmping`. The CODE script inst_orpg_config prompts the user for the type of CODE distribution being installed (Public or NWS). The script then copies the corresponding version of the file to the file `task_tables` which is used by the ORPG.

   - The NWS version of `task_tables` includes tasks associated with MIT/LL and NCAR supplied algorithms.

   - The Public version of `task_tables` disables the startup of tasks associated with MIT/LL and NCAR supplied algorithms because these tasks are not included with the Public Edition. The following commands in the `Operational_processes` list near the end of the public CODE version of `task_tables` have been commented out**.**

   ```
   Operational_processes {
       .        .
       .        .
       .        .
   #   data_qual
   #   hiresvil
       .        .
   #   hireseet
       .        .
   #   nexradMigfa
   #   ntda_alg
   #   ntda_fp
       .        .
   #   icing_hazard
   #   hail_hazard
       .        .
   ```

```
}
```

- **If you have the Public Edition of CODE, the tasks must remain disabled in order for the ORPG to start.**

- **After initial installation, if you make any changes, make a backup copy of this file.**

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

4. Test the installation by launching the ORPG following the instructions in Section IV of this document: *Running the ORPG*.

ORPG Installation Complete - Next Steps

This completes the ORPG installation into a CODE account. Proceed with the instructions in Volume 1, Document 2 to install CODE software (CODE utility updates and sample algorithms).

The next section of this document contains optional procedures to create additional CODE accounts from an existing account.

Vol 1. Document 1 - CODE Specific ORPG Installation Instructions

Section III Supplemental Information

This document is not required for setting up a basic CODE algorithm development environment on a stand-alone workstation. It contains additional information that would be useful for

- accomplishing additional customization of the Unix user account
- configuring the ORPG external interfaces
- maintaining the CODE distribution

WHAT'S NEW for Build 12

- In addition to `cm_ping`, two additional tasks related to wide band ingest and output and 5 LDM related tasks have been commented out of the task_tables configuration files.

Part A. ORPG and Unix Account Environment Files Modified for CODE

Several files must be customized in order to set up the account as an ORPG development environment. Some files set up the basic account environment and others modify makefiles. These files are installed **after** the ORPG source code archive has been extracted and **before** the ORPG is compiled.

Files installed with the `inst_env_config` script.

- The following files are replaced with modified CODE versions:

    ```
    $HOME/.cshrc
    $HOME/make_rpg
    $HOME/src/cpc104/lib001/makefile
    $HOME/src/cpc104/lib003/makefile
    $HOME/src/cpc104/lib005/makefile
    $HOME/src/cpc104/lib006/makefile
    $HOME/src/cpc104/lib009/makefile
    ```

- The following new files are installed:

    ```
    $HOME/build_env_cshrc
    $HOME/orpg_env_cshrc
    $HOME/test_make_cpc100
    ```

- The following files support the bash shells (not tested):

    ```
    $HOME/.bash_profile
    $HOME/.bashrc
    ```

```
$HOME/.envfile
$HOME/build_env_profile
$HOME/orpg_env_profile
```

- The files being replaced are saved with a **`.OLD`** extension.

- A copy of the new file is made with a **`.CODE`** extension.

Files defining the ORPG Account Environment

1. **The account's `.cshrc` file**

   Typically no change is required in the sample `.cshrc` file installed with the script. If you wish to customize your environment, **read the comments included in the file** (for example, changing the **`umask`** setting can adversely affect installation of CODE software and setting the **`noclobber`** shell variable will prevent ORPG launch).

2. **The ORPG run environment file: `orpg_env_cshrc`**

   This file is sourced and the end of **`.cshrc`** and sets the environment for running the ORPG.

   If more than one installed ORPG is going to be run at the same time on a single workstation, the value of the variable **`RMTPORT`** must differ. Manually change the defined value of **`RMTPORT`** on each account. It is recommended that the first account have a value of 50000, the second 51000, etc.

3. **The software build environment file: `build_env_cshrc`**

   No changes required on the Linux platform. This file is sourced at the end of **`.cshrc`** and sets the environment for compiling the ORPG software and for compiling subsets of the software from within the source code tree.

ORPG Build Scripts and Makefiles

1. **The `$HOME/make_rpg` script**

   No changes required. The customized **`make_rpg`** script file installed by the CODE script has been modified to provide some flexibility in setting up the development environment and to reduce unnecessary Error messages. The changes to this script are documented in Appendix G Files Modified for CODE.

2. **The `$HOME/test_make_cpc100` script**

   No changes required. The new **`$HOME/test_make_cpc100`** script that was installed requires no modification.

3. **The makefile in `$HOME/src/cpc104/lib005`**

   No editing required. In addition to other configuration files, the current makefile in this directory installs all the standard account configuration files (.cshrc, .bash_profile, .bashrc) at the end of every ORPG compile. This makefile was modified to not install these three files which avoids overwriting files customized for CODE.

4. **The makefile in `$HOME/src/cpc104/lib003`**

   No editing required. The current makefile in this directory does not install all of the required configuration files into the `~/cfg` and the `~/cfg/dea` directories (this is because another source of initial configuration information is used for the operational ORPG). The script substitutes a makefile that is similar to a previous version in order to install all required configuration files for the development environment.

5. **The makefile in `$HOME/src/cpc104/lib006`**

   No editing required. This file was modified to install additional files into the `$HOME/cfg/dea` directory.

6. **The makefile in `$HOME/src/cpc104/lib001`**

   No editing required. This file was modified to install 9 data files into the `$HOME/cfg` directory to support `rda_simulator` tool.

7. **The makefile in `$HOME/src/cpc104/lib009`**

   No editing required. This file was modified to install a file to `$HOME/tools/cfg/comms` directory so that RPG can be compiled without errors.

Other Files

1. **`.bashrc`**, **`.bash_profile`**, and **`.env_file`**

   Together these files set up the environment for the bash shell (not tested for CODE) as the .cshrc does for the csh shell.

2. **`orpg_env_profile`** and **`build_env_profile`**

   These files set up the ORPG run environment and the build environment for the bash shell (not tested for CODE) and are sourced at the end of **`.bash_profile`**

Part B. ORPG Configuration Files Modified for CODE

The installed configuration files provide the basic information to configure a standalone workstation running the ORPG with no communication managers (no wideband connection to an RDA or narrowband connections for product distribution). These files are installed **after** the ORPG is compiled.

Warning: Many of the configuration files in `$HOME/cfg` are copied or otherwise derived from default versions located in `$HOME/src/cpc104`. Care should be taken to preserve copies of any manual changes to files in `$HOME/cfg` because they may be overwritten when the system is rebuilt or an installation script is run.

Files installed with the `inst_orpg_config` script.

- The following files are replaced with modified CODE versions:

  ```
  $HOME/cfg/comms_link.conf
  $HOME/cfg/site_info.dea
  $HOME/cfg/blockage.lb
  $HOME/cfg/task_tables
  $HOME/cfg/task_attr_table
  ```

- The following new files are installed:

  ```
  $HOME/.rssd.conf
  $HOME/tools/bin/rm_orpg_data
  $HOME/cfg/version_rpg
  $HOME/cfg/task_tables.nws_code
  $HOME/cfg/task_tables.public_code
  ```

- The files being replaced are saved with an **.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`**

Files configuring basic aspects of the ORPG

1. **The Communications Link Configuration file: `$HOME/cfg/comms_link.conf`**

   Normally no editing required. The customized `comms_link.conf` file installed by the script normally requires no modification for a basic development environment. This file configures a stand-alone ORPG that is not controlling an RDA and does not have any narrowband product distribution lines. The file also assumes that the source of base data is either from an 8mm tape

containing Archive II data or is from reading Archive II data from disk files using the ORPG utility `play_a2`. This file does not require modification unless another source of base data is used or narrow band product distribution lines are configured. The critical entries for a development environment are documented in [Appendix G](#) Files Modified for CODE.

2. **The Remote System Services configuration file: `$HOME/.rssd.conf`**

   Modification might be needed. The sample `.rssd.conf` file installed into the account home directory by the script uses hostname rpg as Client. If the hostname of your computer has been aliased to rpg in file /etc/hosts, there is no need to change this file. *The examples provided indicate correct syntax.* Otherwise, replace rpg with the hostname or IP address of your computer.

   o Make a client entry for the TCP/IP address (or hostname) of your workstation.

   > **# RPG Development Workstations**
   > **Client: rpg**

3. **Site Adaptation Data: `site_info.dea` and `blockage.lb`**

   No editing required. The site adaptation data is set for Melbourne Florida. This is the site that is the source of the sample Archive II disk files provided with the initial ORPG installation. The following files are installed in the `$HOME/cfg` directory: `site_info.dea` and `blockage.lb`.

4. **The `$HOME/cfg/task_tables` configuration file**

Two versions of the task_tables configuration file are provided: `task_tables.nws_code` and `task_tables.public_code`. Several tasks have been commented out in both versions: `cm_ping, wbserver, wbserver_ingest, convert_ldm, rpc.ldmd, manage_ldm, ldm_recomb, levelII_status_ICAO_ldmping`. The CODE script inst_orpg_config prompts the user for the type of CODE distribution being installed (Public or NWS). The script then copies the corresponding version of the file to the file `task_tables` which is used by the ORPG.

   ▪ The NWS version of `task_tables` includes tasks associated with MIT/LL and NCAR supplied algorithms.

   ▪ The Public version of `task_tables` disables the startup of tasks associated with MIT/LL and NCAR supplied algorithms because these tasks are not included with the Public Edition. The following commands in the Operational_processes list near the end of the public CODE version of task_tables have been commented out**.**

```
Operational_processes {
    .       .
#   data_qual
#   hiresvil
    .       .
#   hireseet
```

```
         .          .
#    nexradMigfa
#    ntda_alg
#    ntda_fp
         .          .
#    icing_hazard
#    hail_hazard
         .          .
}
```

- If you have the Public Edition of CODE, the tasks must remain disabled in order for the ORPG to start.

5. **The `$HOME/cfg/task_attr_table` configuration file**

No editing required. Option -f is added to task control_rda so that the data being played back looks like coming from RDA.

Other Files

1. **The `$HOME/tools/bin/rm_orpg_data` script**

No editing required. This script provides a safe means to erase all of the ORPG data files located in the data directory (`$ORPGDIR`) configured for the installation account.

Vol 1. Document 1 - CODE Specific ORPG Installation Instructions

Section IV Running the ORPG

Preliminary Notes

- The following actions should be accomplished in the order presented. Specifically, **ORPG tasks should be running before beginning base data ingest (starting the 8mm tape reader or reading Archive II data files).**
- Having problems starting the ORPG? Refer to Appendix H **ORPG Launch Problems**.

TO START ORPG TASKS:

- Log in as an appropriate user, that is the account into which the ORPG is installed.

- Type: `mrpg -p -v startup`

    The **-v** option provides a verbose output.
    The **-p** option cleans up all data stores before starting up.

- Wait for the command prompt to return. Startup normally requires less than one minute.

    A sample output of this command is provided in Appendix I.

Note 1: In Build 14, a new error message is seen about syslog.lb. At the RPG startup with option –p, syslog.lb is deleted first thus can not be opened. It will be re-created by the RPG. User should ignore this error message.

```
18:40:10 mrpg: ORPGDA: RSS_orpgda_lb_open $(ORPGDIR)/mngrpg/syslog.lb failed
(ret = -43)
18:40:10 mrpg: ORPGDA_write ORPGDAT_SYSLOG failed (ret -43)
```

Note 2: If you have the Public Distribution Edition of CODE, a few products are not included and are disabled in the **task_tables** configuration file. For example, the **data_qual, hiresvil, hireseet**, **nexradMigfa**, **ntda_alg, ntda_fp, icing_hazard, hail_hazard** tasks are disabled.

Note 3: If the **sysstat package** was not installed on a Linux Platform, you will receive the

following **iostat** error message. You may see the following after launching the ORPG. This message does not affect using CODE to develop algorithms.

```
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
```

To Check Status of Running Programs:

- Type: `rpg_ps`

    A sample output of *rpg_ps* is provided in Appendix J.

Note: The *rpg_ps* command does not work unless certain ORPG tasks are running (it will not work after executing `mrpg cleanup`). In this case, the status of running tasks can be checked with the standard `ps -ef` command.

To Launch the ORPG User Interface Program:

- Type: `hci`

    When not connected to an operational RDA, it is normal for the *hci* to display a warning for *Wideband Link Failure*.

If the colors of the `hci` application window are washed out, it may be due to having other applications open (i.e. a web browser) on an 8-bit color display.

Note: ORPG algorithm tasks will run without launching the *hci*. Documentation of the *hci* is not included with this package.

Ingest a Source of Base Data

The ORPG utility "`play_a2`" is used for disk file and tape 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.

Method 1, Using Archive II data disk files:

The ORPG utility "`play_a2`" 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 `$HOME/ar2data`. The CODE CD contains additional Archive II disk files.

Execute the following command to ingest these files.

- Type: `play_a2 -d $HOME/ar2data`

- If you have launched the *hci*, observe the RDA radome indicate scanning in progress on the GUI window

If the variable `AR2_DIR` has been set to the `$HOME/ar2data` directory, executing '`play_a2`' will suffice.

See the CODE Utility documentation contained in CODE Guide Volume 4 for additional information the command line mode of `play_a2`.

Method 2, Using Archive II data tapes (requires an 8mm tape drive):

The ORPG `play_a2` utility has an interactive mode for reading radar data off of 8mm tape drives and ingesting into a running ORPG. This mode is selected with the `-a` option.

- Place the Archive II data tape into the 8mm tape drive at the console.
- Type: `play_a2 -a`
- Make selection from displayed menu:

```
Type:     0: Rewind tape
          1: Playback tape
          2: Search a volume and playback
          3: Skip volumes
```

> **4**: `Options`
>
> **5**: `Exit`

- If you have launched the *hci*, observe the RDA radome indicate scanning in progress on the GUI window

Note: The `play_a2` uses the default tape device configured for the workstation.

The CODE Utility documentation contained in CODE Guide Volume 4 does not yet cover the interactive modes of `play_a2`. See the man page.

For Graphic Display of a Subset of Legacy Products:

NOTE: The environmental variable `CVG_DEF_PREF_DIR` must be defined as the path of the location of the default preferences files (normally `$HOME/tools`) for `CVG` to function properly.

- Type: `cvg`   to launch the CODEview Graphics Display tool

Once the utility is launched,

a product must 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.

TO STOP ORPG TASKS:

- Type: `mrpg shutdown`

- Type: `mrpg cleanup`

IMPORTANT: Even though `mrpg cleanup` command is optional, it **should always be executed in a development environment when stopping the ORPG**. The *hci* and two other ORPG tasks (*rssd* and *mrpg*) are still active after shutdown and this command terminates these. It is important to execute `mrpg cleanup` if the ORPG is installed in more than one account. **If these tasks are not terminated during shutdown, an ORPG that is installed in another account will not launch unless the value of `RMTPORT` has been modified.** The `rpg_ps` command will not function after executing `mrpg` **cleanup**.

To Stop Ingest of Base Data:

If using method 1, Archive II disk files:

- Type **ctrl-C** in the terminal that started the 'play_a2' utility

If using method 2, 8mm Archive II tape:

- Type **ctrl-C** in the terminal that started the tape
- select item **5** for 'Exit'

<div style="background-color:red; color:white; text-align:center;">

Volume 1. Guide to Setting Up the Development Environment

</div>

Document 2. Installing CODE Software

This document includes instructions for the installation of the CODE development and analysis utilities and CODE sample algorithms.

WHAT'S NEW for Build 14

- Instructions to generate dual pol test products 340-344, 600-605 and 700-705 are provided in Section IV.

Vol 1. Document 2 - Installing CODE Software

Section I   Software Requisites for CODE Utilities

New for Build 12

CVG 9.0 and later requires additional software packages from the Red Hat Enterprise installation. This is because CVG uses libgd and libgif from Red Hat rather than internal libraries libgd and libungif.

CVG 9.0 and later requires packages `gd`, `gd-devel`, `giflib`, and `giflib-devel`.

If the platform preparation instructions (Red Hat Linux installation) in Volume 1 Document 1 and Appendix B are followed, all of the packages will be installed.

To verify if the required packages are installed, execute:

```
rpm –q gd
rpm –q gd-devel
rpm -q giflib
rpm -q giflib-devel
```

The following procedures require Administrative permissions.   **SA**

If these packages are not installed, use yum to install them:

```
yum -y install gd
yum -y install gd-devel
yum -y install giflib
yum -y install giflib-devel
```

Vol 1. Document 2 - Installing CODE Software

Section II   Instructions for CODE Utilities

Part A. Introduction

NOTE: The CVG 9.2 and CVT 4.4.3 utilities are integrated into the Build 13 ORPG source code tree and compiled with the ORPG.

The major enhancements included in CVG 9.2 are:

Product Display Related changes
   • Improved display of radial products to greatly reduce the number of black pixels between radials, artifacts of the display resolution and the X-windows drawing primitives.
User Interface Enhancements
   ▪ NONE.
Misc Bug Fixes
   ▪ BUG Fixed: The product database size in CVG was smaller than the maximum possible in the RPG. This would cause the display of product other than the product selected for display when using larger product databases.

Other:
   ▪ NONE.

The major enhancements included in CVT 4.4.3 are:

   • A Build 12 change in the radial header for the generic radial component changed the azimuth from center azimuth to beginning azimuth.

• Part B. contains installation instructions for code utility updates (if provided). These procedures accomplish a local installation.
• Part C. provides the optional global (and standalone) installation procedures for CVT and CVG.
• Part D. contains special instructions for the CODE utilities CVG & CVT and the ORPG utility 'play_a2'. The environmental variables that must be set before using `cvt`, `cvg`, and `play_a2` are described.
• Part E. lists the installed file locations for the CODE utilities.

Installation Types

1. The **local installation**, which is accomplished when the ORPG is compiled, has one advantage. With multiple accounts on a workstation, each account could run different versions of a utility. This may be required for CODEview Graphics if some accounts have different ORPG Builds. The local installation requires that `~/tools/bin` and `~/tools/bin/lnux_x86` are in the `path` environmental variable.

2. A second installation type, a 'global' installation, is provided for CVT and CVG. A **global installation** is not required and normally not used. A global installation places the installed executables (and the default CVG configuration files) into a location accessible by more than one account on a workstation. As a convenience, scripts are provided (run with root privileges) which place the executables in `/usr/local/bin`. These scripts must be run after CVG / CVT are compiled. The global installation requires that the installed location (e.g., `/usr/local/bin`) be in the `path` environmental variable. Through modification of each utility's global installation script, another location could be chosen. For CODEview Graphics (`cvg`), an environmental variable must be set to load default preferences.

   A 'standalone' installation option is provided. The **standalone option** permits CVT and CVG to be used outside of (not logged into) an ORPG account. CVT and CVG must be compiled with the standalone option set. **This option should not be used for a local installation.**

Prerequisite Software

All requisite software is provided with a full installation of Red Hat Linux, using the Linux ORPG installation instructions.

Part B. Compiling CODE Utility Updates

A **local installation** of CVG 9.2 and CVT 4.4.3 has been included in baseline of Build 13 source code. For a global installation the procedures in Part C. must be accomplished.

There are no utility updates with this CODE distribution.

Part C. Global Installation Instructions (Optional)

A global installation is not required and normally not used because the CODE utilities are installed when the ORPG is compiled. The purpose of a global installation would be to install the utilities only once and have them accessible by more than one user account.

IMPORTANT: **For best operation, CVG should be compiled with the same ORPG Build with which it will be used.** If multiple accounts on a workstation have a mixture of ORPG Builds (12, or 13, etc.), if the global installation option is used, CVG should be compiled on the most recent build.

All global installation scripts must be executed with **SA** administrative privileges.

The standalone option can be used with a global installation if there is a need to use CVT or CVG outside of an ORPG configured account.

CVT - Global Installation

1. CVT must first be compiled with the executable in the local source code subdirectory `lnux_x86`. If a standalone option is desired the environmental variable `STANDALONE_CVT` must be set prior to compiling CVT (see the **CODE-specific** section of the `.cshrc` file).

2. If accomplishing a global installation of the utility included with the ORPG, the source code directory is `~/src/code_util/tsk004/`.

3. To install **CVT** under `/usr/local/bin`:
   From within the applicable source code directory execute the global installation script: **SA**

        ./cvt_global_install

4. The executable **cvt** can be manually copied to another location (in the **PATH**). In this case ensure the binary is executable by all intended users.

CVG - Global Installation

1. CVG must first be compiled with the executables in the local source code subdirectory `lnux_x86`. If a standalone option is desired the environmental variable `STANDALONE_CVG` must be set prior to compiling CVG (see the **CODE-specific** section of the `.cshrc` file).

2. If accomplishing a global installation of the utility included with the ORPG, the source code directory is `~/src/code_util/tsk001/`.

3. To install **CVG** under `/usr/local/bin`:
   From within the applicable source code directory execute the global installation script: **SA**

        ./cvg_global_install

4. The executables `cvg`, `cvg_read_db`, and `cvg_color_edit` can be manually copied to another location (in the **PATH**). In this case ensure the binary is executable by all intended users.

5. To install the default configuration files under **/usr/local/share**:
   execute the following script: **SA**

        ./cvg_install_config

6. The default configuration files can manually copied to another location. In this case ensure all intended users have both read and write permissions of the configuration files at the top level (e.g., `cvgN.N/.cvgN.N/`) and read permission of all other files.

Part D. Special Instructions - Before using cvt, cvg, and play_a2

CVG - Special Instructions

> The CVG configuration files usually differ significantly from prior versions of CVG. These files are installed in directories associated with a specific CVG release.
>
> **WARNING**: Any locally developed products which were used with prior versions of CVG must be reconfigured from scratch. Do not attempt to reuse any configuration files from previous CVG installations other than locally developed color palette files / digital legend files.

1. The option menu on the CVG main window should be used to set the ORPG Build number (for example '**8**') of the ORPG from which the products were produced. For CVG, the environmental variable `CV_ORPG_BUILD` only affects the initial value in this menu when CVG is first launched.

2. For all installations, the CVG default configuration files must be placed into a directory named `cvgN.N/.cvgN.N` (where `N.N` refers to the CVG version number). These default configuration files must not be modified by the user. The environmental variable `CVG_DEF_PREF_DIR` must point to the parent of that directory (see the **CODE-specific** section of the **.cshrc** file). For example, if the default configuration files are placed in `/mytools/cvg8.0/.cvg8.0`, then `CVG_DEF_PREF_DIR` must be set to `/mytools`. **CRITICAL STEP**
    o   With the **local installation** that is accomplished when the ORPG is compiled, the CVG default configuration files are placed in the directory: `$HOME/tools/cvgN.N/.cvgN.N`. The environmental variable `CVG_DEF_PREF_DIR` must be set to `$HOME/tools` which is the value used in the account configuration file `.cshrc` supplied with CODE.
    o   If the optional **global installation** is accomplished with the `cvg_global_install` and `cvg_install_config` scripts (instructions in Part C.), the CVG default configuration files are placed in the directory: `/usr/local/share/cvgN.N/.cvgN.N`. The environmental variable `CVG_DEF_PREF_DIR` must be set to `/usr/local/share` (see the **CODE-specific** section of the `.cshrc` file).
    o   If the configuration files were manually copied into a custom location or the `cvg_install_config` script was modified during a **global installation**, the definition of `CVG_DEF_PREF_DIR` must reflect this new location (see the **CODE-specific** section of the `.cshrc` file).

CVG Notes - Actions during startup

The configuration files in the directory configured with `CVG_DEF_PREF_DIR` serve as a repository for default configuration files. When launched, `CVG` accomplishes the following:

- Searches for the local configuration directory when launched. If the local configuration directory corresponding to the version of CVG does not exist,
    - o it is created and default configuration files are copied from the default configuration directory. CVG uses a directory named **$HOME/.cvgN.N** - according to the version number (e.g., **$HOME/.cvg9.2**).
    - o **CVG** also checks to see if the installed map data file are installed and decompressed, then creates several sample map files.

CVG uses the local configuration files which can be customized. These local files can be replaced by the defaults by either renaming or deleting the local configuration directory (**$HOME/.cvgN.N**) and subsequently launching **CVG**.

CVG Notes - Standalone Installation

If the **standalone** option is used with a global installation to permit using CVG from any account:

- The environmental variable **STANDALONE_CVG** must be set before CVG is compiled.
- The CVG File->Preferences menu on the main window must be used to locate the product database linear buffer file.

CVT - Special Instructions

1. **IMPORTANT:** To ensure proper operation, the environmental variable **CV_ORPG_BUILD** must be set to the ORPG build number (for example '**11**') of the ORPG from which the products were produced (see the **CODE-specific** section of the **.cshrc** file).

2. In order to configure a new product for decoding the data levels in an unsigned integer array (data packet 16 or the generic radial component), the configuration file containing the Scale-Offset parameters must be placed in the **$HOME/.cvt** directory. A sample configuration file **decode_params.1992** is installed in this directory.

3. If the **standalone** option is used with a global installation to permit using CVT from any account:

- The environmental variable **STANDALONE_CVT** must be set before CVT is compiled.
- The environmental variable **CVT_DB** must be used to locate the product database linear buffer file.

Archive II Disk File - Special Instructions

The ORPG utility **play_a2** is used for all Archive II disk file ingest capability.

1. The environmental variable **AR2_DIR** must be defined for each account using the **play_a2** utility to replay the Archive II disk files (see the **CODE-specific** section of the **.cshrc** file). This variable

represents the default location of stored Archive II disk files. This directory can be used to contain the most commonly used ingest data set. It is convenient to place each data set in individual directories under **AR2_DIR**.

Note: The CODE ORPG installation places three volumes of data into **$HOME/ar2data** as a convenience in order to easily test the ORPG installation. You may wish to install the more extensive collection of disk files into another location. In any case, the **AR2_DIR** environmental variable must be defined for proper operation of the **play_a2** utility.

Part E. Installed file locations

CVT - Installed Files

The following files are installed in the indicated location, based upon whether a local or global installation was performed.

Installed File		Installed Location	
Name	**type**	**local**	**global**
cvt	binary	**~/tools/bin/lnux_x86**	**/usr/local/bin**
decode_params.1992	text	**~/.cvt**	**~/.cvt**

> For a global installation, the location of the executable binary can be changed through modification of the **cvt_global_install** script and appropriate modification of the **path** environmental variable. For a standalone installation, the executable can be manually copied anywhere in the **path**.

CVG - Installed Files

CVG Binary Files

The following files are installed in the indicated location, based upon whether a local or global installation was performed.

Installed File		Installed Location	
Name	**type**	**local**	**global**

`cvg`	binary	`~/tools/bin/lnux_x86`	`/usr/local/bin`
`cvg_read_db`	binary	`~/tools/bin/lnux_x86`	`/usr/local/bin`
`edit_cvgplt`	binary	`~/tools/bin/lnux_x86`	`/usr/local/bin`
`map_cvg`	binary	`~/tools/bin/lnux_x86`	`/usr/local/bin`

> For a global installation, the location of the executable binaries can be changed through modification of the `cvg_global_install` script and appropriate modification of the `path` environmental variable. For a standalone installation, the executables can be manually copied anywhere in the `path`.

CVG Map Data Files

The background map data files are installed into the following locations.

Installed File		Installed Location	
Name	**type**	**local**	**global**
`us_map.dat.bz2`	binary	`~/tools/cvg_map`	`/usr/local/share/cvg_map`
`ak_map.dat.bz2`	binary	`~/tools/cvg_map`	`/usr/local/share/cvg_map`
`hi_map.dat.bz2`	binary	`~/tools/cvg_map`	`/usr/local/share/cvg_map`

These map data files are automatically uncompressed when accessed for the first time by the CVG map utility or the associated scripts.

CVG Configuration Files

The default configuration files are installed into the following locations. The directory name `cvgN.N` represents a version-specific directory name (e.g., `cvg9.2`).

Installed File		Installed Location	
Name	**type**	**local**	**global**
`colors`	dir	`~/tools`/cvgN.N/.cvgN.N	`/usr/local/share`/cvgN.N/.cvgN.N
`help`	dir	`~/tools`/cvgN.N/.cvgN.N	`/usr/local/share`/cvgN.N/.cvgN.N
`legends`	dir	`~/tools`/cvgN.N/.cvgN.N	`/usr/local/share`/cvgN.N/.cvgN.N
`prefs`	text	`~/tools`/cvgN.N/.cvgN.N	`/usr/local/share`/cvgN.N/.cvgN.N
`radar_info`	text	`~/tools`/cvgN.N/.cvgN.N	`/usr/local/share`/cvgN.N/.cvgN.N
`prod_config`	text	`~/tools`/cvgN.N/.cvgN.N	`/usr/local/share`/cvgN.N/.cvgN.N
`resolutions`	text	`~/tools`/cvgN.N/.cvgN.N	`/usr/local/share`/cvgN.N/.cvgN.N
`site_data`	text	`~/tools`/cvgN.N/.cvgN.N	`/usr/local/share`/cvgN.N/.cvgN.N
`prod_db_size`	text	`~/tools`/cvgN.N/.cvgN.N	`/usr/local/share`/cvgN.N/.cvgN.N

sort_method	text	~/**tools**/cvgN.N/.cvgN.N	/**usr**/**local**/**share**/cvgN.N/.cvgN.N
prod_names	text	~/**tools**/cvgN.N/.cvgN.N	/**usr**/**local**/**share**/cvgN.N/.cvgN.N
descript_source	text	~/**tools**/cvgN.N/.cvgN.N	/**usr**/**local**/**share**/cvgN.N/.cvgN.N

Note 1: The configuration files in the above installed locations must not be modified by the user.

Note 2: When CVG is launched, if a local directory **$HOME/.cvgN.N** (where **N.N** is the CVG version number) does not exist, it is created and these configuration files (and directories) are copied into this local configuration directory for use by CVG. Some of the files in this local directory are modified when new products are configured for display.

> For a global installation, the location of the default configuration files can be changed through modification of the **cvg_install_config** script and use of an environmental variable. For a standalone installation, the default configuration files can be placed into any **cvgN.N/.cvgN.N** directory as long as the environmental variable **CVG_DEF_PREF_DIR** points to the parent of that directory and the account has read permissions.

Note: If using the C shell, the added commands can be executed based upon the **path** environmental variable after updating the hash table with the **rehash** command.

Vol 1. Document 2 - Installing CODE Software

Section III Instructions for Sample Algorithms

This 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.

Change History

NEW FOR Sample Algorithms 1.22a (CODE B12r1.7)

- Eliminated debug output from output file.

NEW FOR Sample Algorithms 1.22 (Build 12 x1.116)

- Added a sample algorithm default generation file that modifies the product_generation_tables from the Build 12 ORPG.

Sample Algorithm Installation Instructions

The following procedures should be accomplished while logged into the applicable ORPG account.

1. Obtain the CODE sample algorithm archive `code_alg_1_22a.tar` (included with the CODE software CD in the `==/files_code_sw/` directory) **and place into the ORPG source code directory** (`$HOME/src`) of the applicable account.

2. While in this directory, extract the archive with the following command:
   `tar xvf code_alg_1_22a.tar`

3. The archive is extracted into `$HOME/src/cpc305`. This cpc contains 6 tasks. `tsk001` and `tsk002` sub directories contain the source code and makefiles for Sample Algorithms 1 and 2: Digital Reflectivity and Radial Reflectivity. `tsk003` and `tsk004` contain the source code and makefiles for Sample Algorithm 3 (a two-task chained algorithm) and `tsk005` and `tsk006` contain Sample Algorithm 4.

4. While in the `$HOME/src/cpc305` directory run the script:
   `./install_sample_alg`

5. Restart the ORPG with the `-p` switch to rebuild the binary configuration files.

Overview: Configuration and Installation of Algorithms (optional)

This portion of the document contains on overview of algorithm configuration and installation. **The installation script `install_sample_alg` used above accomplishes all actions required to configure and compile the sample algorithm automatically**. It accomplished items 2 - 4 below. Item 1 lists the configuration attributes for the tasks and products associated with the sample algorithms.

1. **Configuration Parameters**

Prior to compiling the algorithms, the ORPG should be configured to add the following tasks and products. **The sample algorithms are easily configured by following the procedures in the next step.** The following chart lists all of the configuration parameters used for the sample algorithms. The CODE Guide Volume 2 Document 2 - *The ORPG Application Development Guide* contains a detailed explanation of algorithm configuration.

	Sample Algorithm 1 Base Reflectivity	Sample Algorithm 1 Raw Reflectivity
Product Code [Note 1]	1990	1995
Buffer Number [Note 1]	1990	1995
Buffer Name [Note 2]	SR_DIGREFLBASE	SR_DIGREFLRAW
Executable Name [Note 3]	sample1_dig	sample1_dig
Task Name [Note 4]	sample1_base	sample1_raw
Path to linear buffer file	base/sample1_base_refl.lb	base/sample1_raw_refl.lb
Input Buffer Number	78	66
Input Buffer Name	SR_REFLDATA	REFL_RAWDATA
This algorithm is a special case. There are two instances of an executable task 'sample1_dig' with different configuration parameters. Each uses a different input and creates a different product.		

	Sample Algorithm 2 Radial 16-level Reflectivity
Product Code [Note 1]	1991
Buffer Number [Note 1]	1991
Buffer Name [Note 2]	RADREFL
Executable Name [Note 3]	sample2_rad
Task Name [Note 4]	sample2_rad
Path to linear buffer file	base/sample2_radrefl.lb
**Input Buffer Number **	79
**Input Buffer Name **	REFLDATA

* An option is provided to change input buffer to 307 DUALPOL_REFLDATA.

	Sample Algorithm 3 Task 1 Intermediate Product	Sample Algorithm 3 Task 2 Final Product
Product Code [Note 1]	0	1992
Buffer Number [Note 1]	1989	1992
Buffer Name [Note 2]	SAMPLE3_IP	SAMPLE3_FP
Executable Name [Note 3]	sample3_t1	sample3_t2
Task Name [Note 4]	sample3_t1	sample3_t2
Path to linear buffer file	base/sample3_ip.lb	base/sample3_fp.lb
Input Buffer Number	79	1989
Input Buffer Name	REFLDATA	SAMPLE3_IP

	Sample Algorithm 4 Task 1 Intermediate Product 1	Sample Algorithm 4 Task 1 Intermediate Product 2
Product Code [Note 1]	0	0
Buffer Number [Note 1]	1988	1987
Buffer Name [Note 2]	SAMPLE4_IP1	SAMPLE4_IP2
Executable Name [Note 3]	sample4_t1	sample4_t1
Task Name [Note 4]	sample4_t1	sample4_t1
Path to linear buffer file	base/sample4_ip1.lb	base/sample4_ip2.lb
Input Buffer Number	79	79
Input Buffer Name	REFLDATA	REFLDATA
Opt Input Buffer Number	1989	1989
Opt Input Buffer Name	SAMPLE3_IP	SAMPLE3_IP

	Sample Algorithm 4 Task 2 Final Product 1	Sample Algorithm 4 Task 2 Final Product 2
Product Code [Note 1]	1993	1994
Buffer Number [Note 1]	1993	1994
Buffer Name [Note 2]	SAMPLE4_FP1	SAMPLE4_FP2
Executable Name [Note 3]	sample4_t2	sample4_t2
Task Name [Note 4]	sample4_t2	sample4_t2
Path to linear buffer file	base/sample4_fp1.lb	base/sample4_fp2.lb
Input Buffer Number	1988	1987
Input Buffer Name	SAMPLE4_IP1	SAMPLE4_IP2

Note 1: The primary include file for each algorithm no longer defines these values. This is not needed because all algorithms use the "by_name" data access functions. The product code is defined in the product_attr_table file and the buffer numbers are in both the product_attr_table and task_attr_table files.

Note 2: The buffer name is defined in the product_attr_table file. Though not required, it is highly recommended that the buffer name be used as the registration name in the task_attr_table file.

Note 3: The individual task binary makefile (`*.mak`) determines the name of the binary executable. If a different name is used these files must be modified. Normally the task name and executable name are the same.

Note 4: Normally the task name and executable name are the same. One reason for using a different task name and executable name is to have multiple instances of a task running using the same source code with each having different inputs and/or outputs. Sample algorithm 1 demonstrates this capability.

2. Configuration Files

As a convenience, the configuration information is provided in files included with the algorithm source code. The installation script `install_sample_alg` accomplishes the following.

- These new products and tasks can be configured without editing the `product_attr_table`, `task_attr_table`, and `task_tables` configuration files directly. To use this method:
    - Create a subdirectory named `extensions` under the `~/cfg` directory.
    - Then copy the following files from the `~/src/cpc305` directory to the `~/cfg/extensions` directory.
      `product_attr_table.sample_snippet`,
      `task_attr_table.sample_snippet`, and
      `task_tables.sample_snippet`.
- In order to generate the product, either the default generation table in the `product_generation_tables` configuration file must be modified or generation must be selected through the `hci`. The following file configures the sample products for generation.
    - Make a backup copy of `~/cfg/product_generation_tables`.
    - Copy the file `~/src/cpc305/product_generation_tables.sample_alg` to `~/cfg/product_generation_tables`.

3. Adaptation Data

Algorithm adaptation data must be configured via specially named configuration files. The algorithms in `tsk001` and `tsk004` require adaptation data. This data is provided with the source code. The installation script `install_sample_alg` accomplishes the following.

- 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.

4. Compilation Procedures

The algorithms can be compiled by executing the following from within the **~/src/cpc305** directory: The installation script **install_sample_alg** accomplishes the following.

```
make clean

make all

make install
```

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

Vol 1. Document 2 - Installing CODE Software

Section IV Instructions for Dual Pol Test Products

This archive contains the configuration files to generate dual pol test products 340-344, 600-605, 700-705. These configuration files should be installed to ~/cfg/extensions.

<div style="border:1px solid">

New for Build 14

The dual pol test product configuration files are no longer installed to ~/tools/cfg by default. To generate these test products, the configuration files have to be installed to ~/cfg/extensions.

</div>

Dual Pol Test Products Installation Instructions

The following procedures should be accomplished while logged into the applicable ORPG account.

1. Obtain the CODE archive `dp_test_prod.tar` (included with the CODE software CD in the `==/files_code_sw/` directory) **and place into the ORPG cfg directory** (`$HOME/cfg`) of the applicable account.

2. While in this directory, extract the archive with the following command:
    ```
    tar xvf dp_test_prod.tar
    ```

3. The archive is extracted into `$HOME/cfg/dp_test_prod`. There are 3 files in this directory:

    `product_generation_tables.dualpol8bit_test`

    To generate products 340-344

    `product_generation_tables.test_base_prods_8bit`

    To generate products 600-605 and 700-705.

    `install_dual_pol_test_prod.sh`

    To install the above 2 configuration files from this directory and below configuration files to `$HOME/cfg/extensions`:

    `$HOME/src/cpc024/tsk001/`

    `product_attr_table.dualpol8bit_test`

    `task_attr_table.dualpol8bit_test`

    `$HOME/src/cpc102/tsk001`

```
product_attr_table.test_base_prods_8bit_combbase

product_attr_table.test_base_prods_8bit_refldata

task_attr_table.test_base_prods_8bit_combbase

task_attr_table.test_base_prods_8bit_refldata

task_tables.test_base_prods_8bit
```

4. While in the **$HOME/cfg/dp_test_prod** directory run the script:
   ./**install_dual_pol_test_prod.sh**

   Verify all configuration files have been installed to **$HOME/cfg/extensions/:**

   cd ~/cfg/extensions

   ls

5. Restart the ORPG with the **-p** switch to rebuild the binary configuration files.

Volume 1. Guide to Setting Up the Development Environment

Appendices

Appendix A. CODE System Requirements

Appendix B. Red Hat Enterprise Desktop Package Selection

Appendix C. CentOS 5.8 Desktop Installation Guidance

Appendix D. Linux Network Configuration Files

Appendix E. CPC100 Compile Problems

Appendix F. ORPG Compile Problems

Appendix G. Files Modified for CODE

Appendix H. ORPG Launch Problems

Appendix I. Outputs of ORPG Start up

Appendix J. Outputs of rpg_ps

Appendix K. Software Removed for the Public Edition

Appendix A. CODE System Requirements

Build 16 - What's New?

ROC is fielding Build 18 with Red Hat Enterprise Linux 7 (64 bit).

Only the Linux PC platform running Red Hat or CentOS is currently supported for CODE.

Workstation Platform

The *Operational Configuration* provides a development platform that is essentially the same as the operational system. This is not required for algorithm development or implementation. The *Development Configuration* provides an acceptable platform for running an ORPG clone and developing ORPG algorithms but does not match the performance criteria of the operational system.

Performance Testing. Any desktop PC with a current processor and 2 GB of RAM would be sufficient in order to determine the relative performance of an algorithm.

	Operational Configuration	Development Configuration	Notes
Workstation	Dual AMD 6348 CPUs	Any PC with a dual core or quad core CPUs	1
Operating System	Red Hat Enterprise Linux 7 Desktop with Workstation Option (64-bit)	Red Hat Enterprise 7 Desktop with Workstation Option (64-bit) or CentOS 7 Desktop (64-bit)	2
Physical Memory	16 GB	4 GB RAM minimum 16 GB recommended	
Swap Space	TBD	1 GB minimum	3
Disk Drive	1 TB SATA III hard drives	1 GB plus for each ORPG account	4

Display Capability	N/A	24-Bit color, 1024x768 min, 1280x1024 recommended	

Note 1: **With the amount of overhead in the operational system there is little reason to replicate it for development. Any recent quality desktop PC with 4 GB of RAM can be used to obtain a good idea of an algorithms relative performance.**

Note 2: **Red Hat Enterprise 7 Workstation has been selected as the operating system for the deployed ORPG. CentOS 7 has been tested to be a good alternative Operating System of Red Hat 7.**

Note 3: **Currently 1 GB of swap space is sufficient for the CODE development environment.**

Note 4: Does not include space for compilers and other development tools.

Software Language Compilers

The CODE Linux platform uses libraries and software development tools that are provided with the basic distribution of Red Hat Enterprise 5 Desktop with Workstation option.

Compilers used to build the Operational ORPG

- **Compilers and utilities provided with Red Hat Enterprise Workstation**
  - GCC 4.8.5 (includes gcc, g++, and gfortran)
  - GNU make 3.82
  - GNU linker in binutils 2.25.1-32
  - glibc 2.17-196

Appendix B. Red Hat Enterprise Desktop Package Selection

Red Hat Enterprise 7 Desktop with Workstation Option

Red Hat Enterprise Desktop is pre-installed

The required packages must be confirmed if Linux has not been installed using the CODE guidance. Because the current version of the System -> Add/Remove Software tool does not provide the same interface as the installation package selection tool, the easiest way to ensure all required package are installed is to reinstall Linux using distribution media (CDs).

Installing Red Hat Enterprise from Distribution Media

1. Custom Installation must be selected

During the Linux installation process, an option is provided to customize the installation. The default package list is insufficient for CODE so the (*) Customize now option must be selected instead of the ( ) Customize later option

Prior versions of Red Hat Enterprise Workstation included an option to install all packages on the distribution media. This option is no longer provided.

2. Package Group Selection

The selection of packages involves ensuring that needed major package groups under several major areas are selected for installation.

If the (*) Customize now option was checked, the Package Group Selection dialog appears. *If you do not see this dialog before package installation begins, do not continue this installation.*

a. Depending upon the type of Red Hat Enterprise being installed, package groups are divided into 6 or 7 major areas such as: Desktop Environments, Applications, Development, Servers, Base System, and Languages.

b. Within each area,
    o   Ensure the following major groups of packages are selected for installation. [x] indicates selected.

c. Then within each group,
    o   Check the required optional packages are selected as indicated by clicking "Optional Packages". There are a few areas (indicated in the following list) where additional optional packages must be selected for CODE.

NOTE: These instructions make no attempt to identify the minimum required set of packages for the CODE algorithm development environment. Deselecting a package group or un checking an optional package could cause a problem in the development environment for CODE.

The chart below is representative of Red Hat 7. However the details can change between releases. The existence of groups within each area and the division of packages between groups may vary slightly.

The brackets indicate the number of optional packages selected / available in each area. The actual number of packages will vary between Red Hat Releases

The selection of packages involves ensuring that needed major package groups under several major areas are selected for installation.

Since no hard comparison is available between CentOS5.8 and CentOS 7 and package groups are different, installed package groups that looked the closest to those under 5.8. Additional packages to be installed as needed.

Gnome Desktop

```
[x] Legacy X Window System Compatibility

[x] Compatibility Libraries

[x] Development Tools
```

Infrastructure Server

```
[x] FTP Server

[x] Java Platform
```

Vol 1 Appendix B – Red Hat Enterprise Desktop Package Selection

Server with GUI

```
[x] KDE
```

GNOME Desktop

```
[x] Gnome Applications
```

KDE Plasma Workspaces

```
[x] Internet Applications

[x] KDE Applications
```

Development and Creative Workstation

```
[x] Platform Development
```

Appendix C. CentOS 7 Desktop Installation Guidance

CentOS is an Enterprise-class Linux Distribution derived from sources freely provided to the public by a prominent North American Enterprise Linux vendor.

Load Instructions CentOS 7

1. Download CentOS 7 64bit ISO DVD image from:
http://centos.servint.com/centos-7/7/isos/
CentOS-7-x86_64-DVD-1611.iso

2. Boot from CDROM, at the prompt choose **Install CentOS 7** and press [Enter] key.

3. The system will start loading media installer and a Welcome screen should appear. Select your Installation Process Language and click on Continue.

4. The next screen prompt is Installation Summary. Choose time settings (UTC). Click on **Date & Time** and select the options; hit on upper **Done** button to apply configuration.

5. Choose the **Language Support** settings and click the **Done** button.

6. Choose the **Keyboard** and click the upper **Done** button to apply changes.

7. Choose the system software to load (see last two pages for selection listing). Select "Desktop – Gnome".

8. Click on **Installation Destination** menu and choose "I will configure partitioning". Items having issues will be highlighted with an orange triangle icon with an exclamation point. Click on each alert icon to resolve the issues. In case of spg17, the system was allowed to automatically configure partitions because it was a new installation on a system without a previous OS load.

9. Set the system hostname and enable networking. Click on **Network & Hostname** (hostname set to spg4), then enable the Network interface, switching the top **Ethernet** button to ON. Manual configuration was done after the install (see step 13).

10. Click on **Begin Installation** button and set up the password for root account.

11. After installation completes, reboot and log back in as root.

12. Disable SELinux: in /etc/sysconfig/selinux, set SELINUX=disabled; then reboot and log back in as root again.

13. Complete manual network configuration, and activate both network adapters (via GUI under Applications->System Tools->Settings->Network), or by hand editing /etc/sysconfig/network-scripts files ifcfg-em1 and ifcfg-em2 to ensure the following settings. NOTE: DUE TO RECENT SECURITY HARDENING EM1 INTERFACE SHOULD BE DEACTIVATED VIA ifdown em1 COMMAND.

```
ifcfg-em1:

TYPE=Ethernet
    BOOTPROTO=none
    DEFROUTE=no
    IPV4_FAILURE_FATAL=no
    IPV6INIT=yes
```

```
        IPV6_AUTOCONF=yes
        IPV6_DEFROUTE=yes
        IPV6_FAILURE_FATAL=no
        IPV6_ADDR_GEN_MODE=stable-privacy
        NAME=em1
        UUID=e19561ce-4b13-43c3-98a5-0ec928840646
        DEVICE=em1
        ONBOOT=no
        DNS1=10.201.19.9
        ZONE=public
        IPADDR=140.90.23.116
        PREFIX=24
        GATEWAY=140.90.23.1
        IPV6_PEERDNS=yes
        IPV6_PEERROUTES=yes
        IPV6_PRIVACY=no

ifcfg-em2:

TYPE=Ethernet
        BOOTPROTO=none
        DEFROUTE=yes
        IPV4_FAILURE_FATAL=no
        IPV6INIT=no
        IPV6_AUTOCONF=yes
        IPV6_DEFROUTE=no
        IPV6_FAILURE_FATAL=no
        IPV6_ADDR_GEN_MODE=stable-privacy
        NAME=em2
        UUID=3be168e4-c730-4fb7-bf22-2d13b3962b35
        DEVICE=em2
        ONBOOT=yes
        IPADDR=192.168.26.217
        PREFIX=24
        DNS1=10.201.19.9
        DNS2=10.201.19.47
        IPV6_PEERDNS=yes
        IPV6_PEERROUTES=yes
        IPV6_PRIVACY=no
```

Installing CentOS 7 from Distribution Media

Package Group Selection

The selection of packages involves ensuring that needed major package groups under several major areas are selected for installation.

Since no hard comparison is available between CentOS5.8 and CentOS 7 and package groups are different, installed package groups that looked the closest to those under 5.8. Additional packages to be installed as needed.

Gnome Desktop

```
[x] Legacy X Window System Compatibility

[x] Compatibility Libraries
```

```
[x] Development Tools
```

Infrastructure Server

```
[x] FTP Server

[x] Java Platform
```

Server with GUI

```
[x] KDE
```

GNOME Desktop

```
[x] Gnome Applications
```

KDE Plasma Workspaces

```
[x] Internet Applications

[x] KDE Applications
```

Development and Creative Workstation

```
[x] Platform Development
```

Appendix D. Linux Network Configuration Files

Example contents of four files required for TCP/IP networking:

--- /etc/hosts ---
This is the file which typically is in error if attempting to set
up the non DHCP configuration during installation.
The first line of the hosts file contains the local host loopback.
The second line of the hosts file contains the IP address and the
Hostname (nimbus) of the workstation. Alias the hostname to rpg.

```
127.0.0.1        localhost.localdomain     localhost
172.16.15.199    nimbus rpg
```

--- /etc/resolv.conf ---
The only required entry in resolv.conf is the IP address of at
least one DNS server. For example (two nameservers):

```
nameserver 172.16.49.25
nameserver 172.16.49.46
```

--- /etc/sysconfig/network ---
The network file must contain the hostname. On a system with only
one network device card, this file may contain the GATEWAY
(default router) address instead of the 'ifcfg-eth0' file.
For example:

```
NETWORKING=yes
NETWORKING_IPV6=no
HOSTNAME=nimbus
```

--- /etc/sysconfig/networking/devices/ifcfg-eth0 -------------------
The eth0 file is the configuration file for the primary (or only)
network interface card. The entries that must be customized for
the workstation are: IPADDR (IP address), GATEWAY (default router
address), NETMASK (typically 255.255.255.0), HWADDR (the MAC
address of the device card. On systems with only one network
device card, the GATEWAY may be identified in the 'network' file
rather than here. ONBOOT should be yes and DEVICE corresponds to
the filename. For example:

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=172.16.15.199
NETMASK=255.255.255.0
NETWORK=172.16.15.0
BROADCAST=172.16.15.255
HWADDR=00:13:72:D5:8D:A3
GATEWAY=172.16.15.254
TYPE=Ethernet
```

Place copies of the above files into the
/etc/sysconfig/networking/profiles/default directory.

Appendix E. CPC100 Compile Problems

Evaluating the Compilation of CPC 100 Libraries

If the CODE specific instructions are followed, the output of the `test_make_cpc100` script contains both the standard output and standard error of the ORPG build attempt.

Error Messages

- The first step in checking the saved output of the `test_make_cpc100` script to search the file for the string "Error" (match the case). The output of a good compile will not contain either of these words as **message types**. A good command for returning only the error messages is:

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

  The output of the grep command should be blank (no errors).

- If Error messages are found, focus on the first few messages in the file. Subsequent errors are often misleading because they are a result of previous errors. Resolve one or two errors and recompile the libraries.
- Comparing the section containing Error messages with the sample build output file may help evaluate the error. The test output file (`test_make_cpc100.out`) is located in directory `code_b18_0r1_11/output_files/`.

Typical Configuration Problems

Assuming the correct compilers and tools are installed and the correct version of the operating system is being used, failure to successfully build the ORPG is normally due to some problem in the build environment. The error messages contained in the output from the `test_make_cpc100` script will provide an indication of the problem. The following list provides several hints where to look.

- 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*.
- Is the account successfully sourcing the *build environment script*? This can be determined by looking at the output of the `env` command.
- Experiencing file access problems? One cause is not being logged in as the account owner when attempting to compile the ORPG.

Warning Messages

It is not necessary to evaluate every Warning message in the output file, however it is generally a good idea to make a general comparison of the types of Warnings to this sample build output file. The test output file (`test_make_cpc100.out`) is located in directory `code_b18_0r1_11/output_files/`.

Appendix F. ORPG Compile Problems

Evaluating the Compilation of the ORPG Software

If the CODE specific instructions are followed, the output of the `make_rpg` script contains both the standard output and standard error of the ORPG build attempt.

Error Messages

- The first step is checking the saved output of the `make_rpg` script to search the file for the string "Error" (match the case). The output of a good compile normally does not contain either of these words as **message types** (the word 'Error' is included as part of a few names, which is normal). A good command for returning only the error messages is:

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

  The output of the grep command should be blank (no errors).

- If unexpected Error messages are found, focus on the first few messages in the file. Subsequent errors are often misleading because they are a result of previous errors. Resolve one or two errors and recompile the ORPG.
- Comparing the section containing Error messages with the sample build output file may help evaluate the error. The output file `make_rpg.out` (2.2 MB) is located in directory `code_b18_0r1_11/output_files/`.

Meeting Basic System Requirements

If multiple tasks fail to compile, the problem could be not having all the required software packages installed with Red Hat Linux.

Beginning with Red Hat 5, an 'installation number' obtained from Red Hat is required to get a successful installation. The installation number must reflect a license for Red Hat Desktop with workstation option. Red Hat is rather vague in describing the results of not supplying an installation number when prompted during installation. One result is that not all software packages required for CODE will be installed.

Typical Configuration Problems

Vol 1 Appendix F. ORPG Compile Problems

Assuming the correct compilers and tools are installed and the correct version of the operating system is being used, failure to successfully build the ORPG is normally due to some problem in the build environment. The error messages contained in the output from the `make_rpg` script will provide an indication of the problem. The following list provides several hints where to look.

- 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***.
- Is the account successfully sourcing the ***build environment script***? This can be determined by looking at the output of the `env` command.
- Experiencing file access problems? One cause is not being logged in as the account owner when attempting to compile the ORPG.

Warning Messages

- It is not necessary to evaluate every Warning message in the output file, however it is generally a good idea to make a general comparison of the types of Warnings to this sample build output file. The output file `make_rpg.out` (2.2 MB) is located in directory `code_b18_0r1_11/output_files/`.

Appendix G. Files Modified for CODE

The two files with the most significant modifications are included in this appendix. Other files modified for CODE are Volume 1, Document 1, Section III parts A and B.

make_rpg

The following changes have been included in the modified **make_rpg** script for the development environment.

- o The definition of the **makecmd** macro was modified for the Solaris platform using the default location for GNU make as follows:

```
###### changed for CODE ### correct path to Gnu make
#makecmd="/cm/tools/bin/$ARCH/make"
makecmd="/usr/local/bin/make"
######
```

- o ALL occurrences of "**/$RPG_LEVEL**" were changed to "**$RPG_LEVEL**" (i.e., removed the leading "/"). (This change is made because we pass the complete **path** to the ORPG installation directory (**$HOME**) to the **make_rpg** script. The original development environment passed a simple directory name without the leading "/").
- o A command setting the **LD_LIBRARY_PATH** variable was commented out because the variable is set in the build_env files in the CODE environment.
- o The method of defining symbolic links in the **cpc904** directory was changed in order to support cloning (or copying) of development accounts.

```
###### RELATIVE PATH LINKS FOR CODE ACCOUNT CLONING PROCEDURE ###
##ln -s $RPG_LEVEL/src/cpc904 $RPG_LEVEL/src/cpc904/lib001/sys/snet
##ln -s $RPG_LEVEL/src/cpc904 $RPG_LEVEL/src/cpc904/tsk002/sys/snet
##ln -s $RPG_LEVEL/src/cpc904 $RPG_LEVEL/src/cpc904/tsk003/sys/snet
##ln -s $RPG_LEVEL/src/cpc904 $RPG_LEVEL/src/cpc105/tsk002/sys/snet
cd $RPG_LEVEL/src/cpc904/lib001/sys
rm -f snet
ln -s ../.. snet
cd $RPG_LEVEL/src/cpc904/tsk002/sys
rm -f snet
ln -s ../.. snet
cd $RPG_LEVEL/src/cpc904/tsk003/sys
rm -f snet
ln -s ../.. snet
cd $RPG_LEVEL/src/cpc105/tsk002/sys
rm -f snet
ln -s ../../../cpc904 snet
cd $workdir
######
```

- o The following modification corrects a problem if accomplishing more than one Build attempt for the ORPG on the Linux platform.

```
###### Changed for CODE, eliminates error messages after first build
### existing links stopped subsequent install
if [ -f $MAKETOP/bin/$ARCH/cm_tcp1  ]
then
    rm -f $MAKETOP/bin/$ARCH/cm_tcp1
fi
######
```

- o The following modification eliminates error messages on subsequent compiles (if accomplished).

```
###### Changed for CODE, eliminates error messages after first build
### set directory permission to permit subsequent installs
if [ -d $MAKETOP/orda ]
then
    chmod -R u+w $MAKETOP/orda/doc
    chmod -R u+w $MAKETOP/orda/images
fi
######
```

- o A command was modified to permit the removal of a temporary shared library without prompting the user.

```
###### CODE CHANGE: allow removal without prompting the user ###
#rm /$RPG_LEVEL/lib/$ARCH/libinfrlb.*
rm -f $RPG_LEVEL/lib/$ARCH/libinfrlb.*
######
```

comms_link.conf

The following entries in `comms_link.conf` are required for a basic development environment that uses Archive II tape or Archive II disk files as the data input.

- The name of the wideband comm manager for the RDA link must be a dummy name (such as `player`). The RDA link is identified by comparing the value in the first column labeled with the value set for `RDA_link` in later this file. In the example below, the name of the RDA comm manager is `player`.

```
# LN   UN   CN   DN   PN   LT            LR  CS          MPS  NS   LS DEN CLASS TOUT AW
   0    0    0    0    0  Dedic   1536000 player    4096   1    0   0    1
   .    .    .    .    .    .          .       .          .    .    .    .  . .
   .    .    .    .    .    .          .       .          .    .    .    .  . .


   RDA_link 0
```

- All other communication line entries must be commented out.

```
# LN   UN   CN   DN   PN   LT            LR  CS          MPS  NS   LS DEN CLASS TOUT AW
##  0    0    0    0    0  Dedic   1536000 cm_tcp_     4096   1    0   0    1
    0    0    0    0    0  Dedic   1536000 player      4096   1    0   0    1
# the following line can be used for nbtcp distribuion
#   1    1    1    1    0  Dedic    100000 cm_tcp_      128   2    0   1   99    0
##   1    1    1    1    0  D-in      14400 cm_uconx_    128   2    0   1    2   60   LINE
##   2    1    1    1    1  D-in      14400 cm_uconx_    128   2    0   1    2   60   LINE
##   3    9    9    0    0  Dedic   1500000 cm_tcp_      128   2    0   1   99   60
##   4    9    9    0    0  Dedic   1500000 cm_tcp_      128   2    0   1   99   60
##   5    9    9    0    0  Dedic   1500000 cm_tcp_      128   2    0   1   99   60
```

- The number of links must be set to 1 (for the remaining RDA link).

```
   number_links 1
```

- Make a backup copy of the new `comms_link.conf` file if you make any changes.

Appendix H. ORPG Launch Problems

Meeting Basic System Requirements

If multiple tasks fail to launch, the problem could be not having all the required software packages installed with Red Hat Linux.

Beginning with Red Hat 5, an 'installation number' obtained from Red Hat is required to get a successful installation. The installation number must reflect a license for Red Hat Desktop with workstation option. Red Hat is rather vague in describing the results of not supplying an installation number when prompted during installation. One result is that not all software packages required for CODE will be installed.

Typical Configuration Problems

Initially, failure to completely launch the ORPG is normally due to some problem in the run environment. The error messages contained in the output from the **mrpg -v startup** command will provide an indication of the problem. The following list provides several hints where to look.

- Recheck the configuration instructions for missed steps. For example: Are the **.cshrc**, **.profile**, and **.dtprofile** files the same versions provided with the CODE distribution? Was a **~/tmp** directory created with appropriate permissions? Were all of the configuration files modified as instructed and saved? Were both of the CODE installation scripts executed (one before compiling the RPG and one after)?
- Is the account successfully sourcing the ***ORPG run environment script***? This can be determined by looking at the output of the **printenv** or **env** command and examining the values of **path**, **LD_LIBRARY_PATH**, **ORPGDIR**, etc.
- Experiencing file access problems? Unable to create linear buffer files? One cause is not being logged in as the account owner when attempting to run the ORPG.
- Unable to find an executable file or a command? Check the value of the **path** variable.
- One source of a failure of the ORPG to launch is a missing **.rssd.conf** file.

```
code17_0r1_13:code17_0r1_13/  60 > mrpg -p -v startup
01/26/10 18:48:00 Checking permanent file
/home/code17_0r1_13/data/logs/mrpg.log
18:48:00 mrpg: start_rssd failed (0) - mrpg_main.c:124
code17_0r1_13:code17_0r1_13/  61 >
```

- Another source of a failure is due to certain errors in the TCP/IP network setup. During a Red Hat Linux installation, the **/etc/hosts** file is often misconfigured by the Linux administration tools. For example, the host name is mixed with the localhost:
  ```
  127.0.0.1    dev2 localhost.localdomain localhost
  192.168.x.x  dev2
  ```
  To fix the problem, just remove the host name (**dev2**) from the localhost line. See Appendix D.

```
code17_0r1_13:code17_0r1_13/  64 > mrpg -p -v startup
01/26/10 18:55:00 Checking permanent file
/home/code17_0r1_13/data/logs/mrpg.log
18:55:00 mrpg: mrpg goes to background
18:55:00 mrpg: Reading task tables
18:55:00 mrpg:     Reading task attr table file
/home/code17_0r1_13/cfg/task_attr_table
18:55:00 mrpg:     Reading task table file /home/code17_0r1_13/cfg/task_tables
18:55:00 mrpg: Empty shutdown commands table
18:55:00 mrpg: Cleaning up all data stores...
18:55:01 mrpg: Start up RPG - Non-operational
18:55:01 mrpg: Reading data table
18:55:01 mrpg:     Reading data table file
/home/code17_0r1_13/cfg/data_attr_table
18:55:01 mrpg: Reading product table
18:55:01 mrpg:     Reading product table file
/home/code17_0r1_13/cfg/product_attr_table
18:55:01 mrpg: Generating system configuration file
18:55:01 mrpg:     Use old system config file
18:55:01 mrpg: Reading comms configuration
18:55:01 mrpg: RMT: authentication failed in connecting to dev2
18:55:01 mrpg: RMT: authentication failed in connecting to dev2
18:55:01 mrpg: LB_open (create) nds LB
dev2:/home/code17_0r1_13/data/infr/nds.lb failed (ret -1010) -
mrpg_pr*_info.c:132
18:55:02 mrpg: mrpg exits with 1
code17_0r1_13:code17_0r1_13/  65 >
```

Not Related to Initial Configuration

If the ORPG had been running previously and is now experiencing launch problems:

- If there is more than one ORPG installed on this workstation, check for ORPG tasks still running from the other account. Typically these could include **mrpg** and **rssd** which would remain if the other account was shutdown without executing **mrpg cleanup**. These tasks must be killed before an ORPG in a different account can be launched. The launch failure occurs quickly in this case with an **EN_register** failure:

```
code17_0r1_13:/home/code17_0r1_13:  65>mrpg -p -v startup
01/26/10 19:08:40 Checking permanent file
/home/code17_0r1_13/data/logs/mrpg.log
19:08:40 mrpg: mrpg goes to background
19:08:40 mrpg: RMT: connect to the local rssd failed (errno 111)
19:08:40 mrpg: Reading task tables
19:08:40 mrpg:     Reading task attr table file
/home/code17_0r1_13/cfg/task_attr_table
19:08:40 mrpg:     Reading task table file /home/code17_0r1_13/cfg/task_tables
19:08:40 mrpg: Empty shutdown commands table
19:08:40 mrpg: Cleaning up all data stores...
19:08:40 mrpg: Start up RPG - Non-operational
19:08:40 mrpg: Reading data table
19:08:40 mrpg:     Reading data table file
/home/code17_0r1_13/cfg/data_attr_table
```

```
19:08:40 mrpg: Reading product table
19:08:40 mrpg:     Reading product table file
/home/code17_0r1_13/cfg/product_attr_table
19:08:40 mrpg: RMT: connect to the local rssd failed (errno 111)
19:08:40 mrpg: EN_register failed (ret -1007) - mrpg_ge*onfig.c:62
19:08:42 mrpg: mrpg exits with 1
code17_0r1_13:/home/code17_0r1_13:  66>
```

- Another reason for the ORPG launch to fail is a syntax error in the **task_attr_table**
  configuration file. In this case the **mrpg cleanup** command may also fail.

```
code17_0r1_13:cfg/ 75 > mrpg -p -v startup
01/26/10 19:12:23 Checking permanent file
/home/code17_0r1_13/data/logs/mrpg.log
19:12:23 mrpg: mrpg goes to background
19:12:23 mrpg: Reading task tables
19:12:23 mrpg:     Reading task attr table file
/home/code17_0r1_13/cfg/task_attr_table
19:12:23 mrpg: CS_entry CS_NEXT_LINE failed (-798) - mrpg_re*_tats.c:286
19:12:23 mrpg: Failed in first call to MRT_init
19:12:25 mrpg: mrpg exits with 1
code17_0r1_13:cfg/ 76 >
```

- If the ORPG launch stops at the point of Initializing the Product Generation and Distribution
  function, the **product_attr_table** configuration file may have a syntax error or incorrect
  information entered. The following output of the **mrpg -p -v startup** command is one
  example of this error:

```
code17_0r1_13:cfg/ 86 > mrpg -p -v startup
01/26/10 19:21:27 Checking permanent file
/home/code17_0r1_13/data/logs/mrpg.log
19:21:27 mrpg: mrpg goes to background
19:21:27 mrpg: Reading task tables
19:21:27 mrpg:     Reading task attr table file
/home/code17_0r1_13/cfg/task_attr_table
19:21:27 mrpg:     Reading task table file /home/code17_0r1_13/cfg/task_tables
19:21:27 mrpg: Empty shutdown commands table
19:21:27 mrpg: Cleaning up all data stores...
19:21:27 mrpg: Start up RPG - Non-operational
19:21:27 mrpg: Reading data table
19:21:27 mrpg:     Reading data table file
/home/code17_0r1_13/cfg/data_attr_table
19:21:27 mrpg: Reading product table
19:21:27 mrpg:     Reading product table file
/home/code17_0r1_13/cfg/product_attr_table
19:21:27 mrpg: CS: unclosed { (file /home/code17_0r1_13/cfg/product_attr_table)
- mrpg_main.c:825
19:21:27 mrpg: CS: unclosed { (file /home/code17_0r1_13/cfg/product_attr_table)
- mrpg_main.c:825
19:21:27 mrpg: CS_entry CS_NEXT_LINE failed (-798) - mrpg_re*_dats.c:650
19:21:27 mrpg: Failed in reading product attributes - mrpg_re*_dats.c:117
19:21:29 mrpg: mrpg exits with 1
code17_0r1_13:cfg/ 87 >
```

- Depending upon the manner in which the **site_info.dea** file is corrupted, there may be no apparent problem at launch. However, if some parameters of the station_type are missing, the corrupted data could lead to the following message. If the binary adaptation data is corrupted after launch, this can prevent shutdown of the ORPG as well.

```
code17_0r1_13:cfg/ 96 > mrpg -p -v startup
01/26/10 19:26:13 Checking permanent file
/home/code17_0r1_13/data/logs/mrpg.log
19:26:13 mrpg: mrpg goes to background
19:26:13 mrpg: Reading task tables
19:26:13 mrpg:      Reading task attr table file
/home/code17_0r1_13/cfg/task_attr_table
19:26:13 mrpg:      Reading task table file /home/code17_0r1_13/cfg/task_tables
19:26:13 mrpg: Empty shutdown commands table
19:26:13 mrpg: Cleaning up all data stores...
19:26:13 mrpg: Start up RPG - Non-operational
19:26:13 mrpg: Reading data table
19:26:13 mrpg:      Reading data table file
/home/code17_0r1_13/cfg/data_attr_table
19:26:13 mrpg: Reading product table
19:26:13 mrpg:      Reading product table file
/home/code17_0r1_13/cfg/product_attr_table
19:26:13 mrpg: Generating system configuration file
19:26:13 mrpg:      Use old system config file
19:26:13 mrpg: Reading comms configuration
19:26:14 mrpg: RPG state file /home/code17_0r1_13/data/rpg_state created
19:26:14 mrpg: Removing all RPG operational tasks
19:26:14 mrpg: Removing all RPG tasks ...
19:26:14 mrpg: Checking/creating/clearing RPG data stores - startup
19:26:14 mrpg: Checking permanent file /home/code17_0r1_13/data/config_device.*
19:26:14 mrpg: Checking permanent file /home/code17_0r1_13/data/trap.log
19:26:24 mrpg: Executing init commands - startup
19:26:24 mrpg: --->Initialize Adaptation Data
19:26:24 mrpg: RPG init command (init_adapt_data) failed (exit 1) -
mrpg_pr*_cmds.c:559
19:26:24 mrpg: mrpg exits with 1
code17_0r1_13:cfg/ 97 >
```

- A corrupted **product_generation_tables** file may produce the following output.

```
code17_0r1_13:cfg/ 103 > mrpg -p -v startup
01/26/10 19:28:52 Checking permanent file
/home/code17_0r1_13/data/logs/mrpg.log
19:28:52 mrpg: mrpg goes to background
19:28:52 mrpg: Reading task tables
19:28:52 mrpg:      Reading task attr table file
/home/code17_0r1_13/cfg/task_attr_table
19:28:52 mrpg:      Reading task table file /home/code17_0r1_13/cfg/task_tables
19:28:52 mrpg: Empty shutdown commands table
19:28:52 mrpg: Cleaning up all data stores...
19:28:52 mrpg: Start up RPG - Non-operational
19:28:52 mrpg: Reading data table
19:28:52 mrpg:      Reading data table file
/home/code17_0r1_13/cfg/data_attr_table
19:28:52 mrpg: Reading product table
```

```
19:28:52 mrpg:     Reading product table file
/home/code17_0r1_13/cfg/product_attr_table
19:28:52 mrpg: Generating system configuration file
19:28:52 mrpg:     Use old system config file
19:28:52 mrpg: Reading comms configuration
19:28:53 mrpg: RPG state file /home/code17_0r1_13/data/rpg_state created
19:28:53 mrpg: Removing all RPG operational tasks
19:28:53 mrpg: Removing all RPG tasks ...
19:28:53 mrpg: Checking/creating/clearing RPG data stores - startup
19:28:53 mrpg: Checking permanent file /home/code17_0r1_13/data/config_device.*
19:28:53 mrpg: Checking permanent file /home/code17_0r1_13/data/trap.log
19:29:02 mrpg: Executing init commands - startup
19:29:02 mrpg: --->Initialize Adaptation Data
19:29:03 mrpg: --->Initialize the Binary Task Attribute Table
19:29:03 mrpg: --->Initialize Critical Data Stores
19:29:03 mrpg: --->Initialize RDA Alarms Table
19:29:03 mrpg: --->Initialize the ITCs
19:29:03 mrpg: --->Check HYDROMET Files.
19:29:03 mrpg: --->Initialize HYDROMET Files.
19:29:03 mrpg: --->Initialize GSM
19:29:04 mrpg: --->Initialize Binary Product Attributes Table
19:29:04 mrpg: --->Initialize Product Distribution
19:29:04 mrpg: --->Initialize Routine Request Product Generation
19:29:04 mrpg: --->Initialize Product Generation Tables
19:29:04 mrpg: RPG init command (mnttsk_pgt -t startup) failed (exit 1) -
mrpg_pr*_cmds.c:559
19:29:04 mrpg: mrpg exits with 1
code17_0r1_13:cfg/ 104 >
```

- A launch failure could be due to the binary configuration files having become corrupt. They can be replaced by erasing the contents of the **$ORPGDIR** directory by using the **-p** option for **mrpg startup**.

- Another cause of launch failure is the consumption of addressable memory by unreleased shared memory segments. Each installed ORPG allocates four shared memory segments, totaling over 50 MB. These are not released on shutdown. Normally these segments are reused the next time the ORPG is started. However, there are times when the ORPG allocates new memory segments when restarted. Allocated memory segments can be listed by executing the **ipcs -a** command. If more than four are listed for any account into which the ORPG is installed, they can be eliminated by rebooting the operating system.

- **When all else fails, reboot the workstation**. There have been other instances where restarting the operating system has corrected an unknown problem that prevented ORPG launch.

Problems launching the X-windows components of the ORPG (the **hci** and **xpdt**):

- Check the value of the SCREEN environmental variable.
- Running from a remote X terminal? If so, check the operation of these applications from the console.

Appendix I. Outputs of ORPG Start up

```
code17_0r1_13:/home/code17_0r1_13:  28>mrpg -p -v startup
03/04/15 18:51:46 mrpg: mrpg goes to background
18:51:46 mrpg: Reading task tables
18:51:46 mrpg:     Reading task attr table file
/home/code17_0r1_13/cfg/task_attr_table
18:51:46 mrpg:        cpu_limit set to 80 for veldeal
18:51:46 mrpg:        mem_limit set to 2000 for veldeal
18:51:46 mrpg:        Reading task table file /home/code17_0r1_13/cfg/task_tables
18:51:46 mrpg:        Common cpu_limit set to 50
18:51:46 mrpg:        Common mem_limit set to 60
18:51:46 mrpg:        Common cpu_window set to 40
18:51:46 mrpg: Empty shutdown commands table
18:51:46 mrpg: Cleaning up all data stores...
18:51:46 mrpg: Start up RPG - Non-operational
18:51:46 mrpg: Reading data table
18:51:46 mrpg:     Reading data table file
/home/code17_0r1_13/cfg/data_attr_table
18:51:46 mrpg: Reading product table
18:51:46 mrpg:     Reading product table file
/home/code17_0r1_13/cfg/product_attr_table
18:51:46 mrpg: Generating system configuration file
18:51:46 mrpg:     New system config file generated
18:51:46 mrpg: Reading comms configuration
18:51:46 mrpg: RPG state file /home/code17_0r1_13/data/rpg_state created
18:51:46 mrpg: RPG System is STARTING UP
18:51:46 mrpg: ORPGDA: RSS_orpgda_lb_open $(ORPGDIR)/mngrpg/syslog.lb failed
(ret = -43)
18:51:46 mrpg: ORPGDA_write ORPGDAT_SYSLOG failed (ret -43)
18:51:46 mrpg: Removing all RPG operational tasks
18:51:46 mrpg: Removing all RPG tasks ...
18:51:47 mrpg: Checking/creating/clearing RPG data stores - startup
18:51:47 mrpg: Checking permanent file /home/code17_0r1_13/data/config_device.*
18:51:47 mrpg: Checking permanent file /home/code17_0r1_13/data/trap.log
18:51:47 mrpg: Checking permanent file /home/code17_0r1_13/data/owr_server.log
18:51:47 mrpg: Checking permanent file
/home/code17_0r1_13/data/syslog_shadow.lb
18:51:57 mrpg: Executing init commands - startup
18:51:57 mrpg: --->Initialize Adaptation Data
18:51:58 mrpg: --->Initialize the Binary Task Attribute Table
18:51:58 mrpg: --->Initialize Critical Data Stores
18:51:58 mrpg: --->Initialize RDA Alarms Table
18:51:58 mrpg: --->Initialize the ITCs
18:51:58 mrpg: --->Check HYDROMET Files.
18:51:58 mrpg: --->Initialize HYDROMET Files.
18:51:58 mrpg: --->Initialize GSM
18:51:58 mrpg: --->Initialize Binary Product Attributes Table
18:51:58 mrpg: --->Initialize Product Distribution
18:51:58 mrpg: --->Initialize Routine Request Product Generation
18:51:58 mrpg: --->Initialize Product Generation Tables
18:51:58 mrpg: --->Initialize Alert Requests/Alert Thresholds
18:51:58 mrpg: --->Initialize Loadshed Information
18:51:59 mrpg: --->Initialize Clutter
18:51:59 mrpg: --->Initialize RDA Adaptation Data
```

Vol 1 Appendix I. Outputs of ORPG Start up

```
18:51:59 mrpg: --->Initialize ISDP Estimate
18:51:59 mrpg: Starting operational processes
18:51:59 mrpg:     Execute op process: recomb -T a_recomb -A -l 1000
18:51:59 mrpg:     Execute op process: alerting
18:51:59 mrpg:     Execute op process: basrflct -T basrflct
18:51:59 mrpg:     Execute op process: basspect -T basspect
18:51:59 mrpg:     Execute op process: basvgrid
18:51:59 mrpg:     Execute op process: basvlcty -T basvlcty
18:51:59 mrpg:     Execute op process: bref8bit -T bref8bit
18:51:59 mrpg:     Execute op process: bvel8bit -T bvel8bit
18:51:59 mrpg:     Execute op process: clutprod
18:52:00 mrpg:     Execute op process: cmprfape
18:52:00 mrpg:     Execute op process: cmprfcg -T cmprfcg
18:52:00 mrpg:     Execute op process: cmprflct
18:52:00 mrpg:     Execute op process: combattr
18:52:00 mrpg:     Execute op process: control_rda -d -l 5000 -v
18:52:00 mrpg:     Execute op process: cpcntalg
18:52:00 mrpg:     Execute op process: crapeprd
18:52:00 mrpg:     Execute op process: data_qual
18:52:00 mrpg:     Execute op process: dp_dua_accum -T dp_dua_accum
18:52:00 mrpg:     Execute op process: dp_elev_prod
18:52:00 mrpg:     Execute op process: dp_lt_accum
18:52:01 mrpg:     Execute op process: dp_precip_4bit
18:52:01 mrpg:     Execute op process: dp_precip_8bit
18:52:01 mrpg:     Execute op process: dp_s2s_accum
18:52:01 mrpg:     Execute op process: dpprep
18:52:01 mrpg:     Execute op process: dqa_elev
18:52:01 mrpg:     Execute op process: dualpol4bit
18:52:01 mrpg:     Execute op process: dualpol8bit
18:52:01 mrpg:     Execute op process: ecotppro
18:52:01 mrpg:     Execute op process: elev_prod -T elev_prod
18:52:01 mrpg:     Execute op process: epre
18:52:02 mrpg:     Execute op process: hail_hazard
18:52:02 mrpg:     Execute op process: hailalg
18:52:02 mrpg:     Execute op process: hailprod
18:52:02 mrpg:     Execute op process: hca
18:52:02 mrpg:     Execute op process: hci_agent
18:52:02 mrpg:     Execute op process: hhc8bit -T hhc8bit
18:52:02 mrpg:     Execute op process: hireseet
18:52:02 mrpg:     Execute op process: hiresvil
18:52:02 mrpg:     Execute op process: hybrprod
18:52:02 mrpg:     Execute op process: icing_hazard
18:52:02 mrpg:     Execute op process: itwsdbv
18:52:03 mrpg:     Execute op process: lcrap
18:52:03 mrpg:     Execute op process: lcrappg
18:52:03 mrpg:     Execute op process: lcrflct
18:52:03 mrpg:     Execute op process: recomb -T ldm_recomb -A -l 1000
18:52:03 mrpg:     Execute op process: mda1d
18:52:03 mrpg:     Execute op process: mda2d
18:52:03 mrpg:     Execute op process: mda3d
18:52:03 mrpg:     Execute op process: mdaprod
18:52:03 mrpg:     Execute op process: mdattnn
18:52:03 mrpg:     Execute op process: mlda
18:52:04 mrpg:     Execute op process: mlprod -T mlprod
18:52:04 mrpg:     Execute op process: mngdskerr
18:52:04 mrpg:     Execute op process: nexradAmda
18:52:04 mrpg:     Execute op process: nexradMigfa -N
```

```
18:52:04 mrpg:        Execute op process: ntda_alg
18:52:04 mrpg:        Execute op process: ntda_fp
18:52:04 mrpg:        Execute op process: owr_server -l
18:52:04 mrpg:        Execute op process: pbd -v -l 5000
18:52:04 mrpg:        Execute op process: pcipdalg
18:52:04 mrpg:        Execute op process: prcpadju
18:52:04 mrpg:        Execute op process: prcpprod
18:52:05 mrpg:        Execute op process: prcprtac
18:52:05 mrpg:        Execute op process: prcpuspt
18:52:05 mrpg:        Execute op process: prfbmap
18:52:05 mrpg:        Execute op process: prfselect
18:52:05 mrpg:        Execute op process: ps_onetime -v 3
18:52:05 mrpg:        Execute op process: ps_routine -v 3 -l500
18:52:05 mrpg:        Execute op process: qia
18:52:05 mrpg:        Execute op process: qperate
18:52:05 mrpg:        Execute op process: radcdmsg -T radcdmsg
18:52:05 mrpg:        Execute op process: recclalg
18:52:05 mrpg:        Execute op process: recclprods
18:52:06 mrpg:        Execute op process: recomb -T recomb -l 1000
18:52:06 mrpg:        Execute op process: basrflct -T replay_basrflct
18:52:06 mrpg:        Execute op process: basspect -T replay_basspect
18:52:06 mrpg:        Execute op process: basvlcty -T replay_basvlcty
18:52:06 mrpg:        Execute op process: bref8bit -T replay_bref8bit
18:52:06 mrpg:        Execute op process: bvel8bit -T replay_bvel8bit
18:52:06 mrpg:        Execute op process: cmprfcg -T replay_cmprfcg
18:52:06 mrpg:        Execute op process: dp_dua_accum -T replay_dp_dua_accum
18:52:06 mrpg:        Execute op process: radcdmsg -T replay_radcdmsg
18:52:06 mrpg:        Execute op process: srmrmrv -T replay_srmrmrv
18:52:07 mrpg:        Execute op process: user_sel_LRM -T replay_user_sel_LRM
18:52:07 mrpg:        Execute op process: vad -T replay_vad
18:52:07 mrpg:        Execute op process: vertxsct -T replay_vertxsct
18:52:07 mrpg:        Execute op process: rpgdbm -v
18:52:07 mrpg:        Execute op process: saaprods
18:52:07 mrpg:        Execute op process: saausers
18:52:07 mrpg:        Execute op process: segmtalg
18:52:07 mrpg:        Execute op process: snowaccum
18:52:07 mrpg:        Execute op process: elev_prod -T sr_elev_prod
18:52:08 mrpg:        Execute op process: srmrmrv -T srmrmrv
18:52:08 mrpg:        Execute op process: status_prod
18:52:08 mrpg:        Execute op process: stmtrprd
18:52:08 mrpg:        Execute op process: strucprod
18:52:08 mrpg:        Execute op process: superes8bit
18:52:08 mrpg:        Execute op process: superob_vel
18:52:08 mrpg:        Execute op process: tda1d
18:52:08 mrpg:        Execute op process: tda2d3d
18:52:08 mrpg:        Execute op process: tda2d3dru
18:52:08 mrpg:        Execute op process: tdaruprod
18:52:08 mrpg:        Execute op process: trfrcalg
18:52:09 mrpg:        Execute op process: tvsprod
18:52:09 mrpg:        Execute op process: update_alg_data
18:52:09 mrpg:        Execute op process: user_sel_LRM -T user_sel_LRM
18:52:09 mrpg:        Execute op process: vad -T vad
18:52:09 mrpg:        Execute op process: veldeal -I
18:52:09 mrpg:        Execute op process: vertxsct -T vertxsct
18:52:09 mrpg:        Execute op process: viletalg
18:52:09 mrpg:        Execute op process: vilprod
18:52:09 mrpg:        Execute op process: vwindpro
```

Vol 1 Appendix I. Outputs of ORPG Start up

```
18:52:09 mrpg:      Execute op process: wideband_agent
18:52:09 mrpg: All operational processes started. Waiting for OP ready ...
18:52:11 mrpg: RPG State: OPERATE
18:52:11 mrpg: RPG Operability Status: ONLINE
18:52:11 mrpg: RPG System Startup Completed
18:52:11 mrpg: RPG startup completed
code17_0r1_13:/home/code17_0r1_13:  29>
```

Appendix J. Outputs of rpg_ps

```
code17_0r1_13:/home/code17_0r1_13:   29>rpg_ps
              name   pid   cpu     mem   life command
          a_recomb  21980    0m    284K    47s recomb -T a_recomb -A -.
          alerting  21982    0m    308K    47s alerting
          basrflct  21984    0m    316K    47s basrflct -T basrflct
          basspect  21986    0m    316K    47s basspect -T basspect
          basvgrid  21988    0m    288K    47s basvgrid
          basvlcty  21990    0m    320K    47s basvlcty -T basvlcty
           bref8bit  21992    0m    300K    47s bref8bit -T bref8bit
           bvel8bit  21994    0m    304K    47s bvel8bit -T bvel8bit
          clutprod  21996    0m    732K    47s clutprod
          cmprfape  21998    0m    416K    46s cmprfape
           cmprfcg  22000    0m    476K    46s cmprfcg -T cmprfcg
          cmprflct  22002    0m    288K    46s cmprflct
          combattr  22004    0m    308K    46s combattr
       control_rda  22006    0m    228K    46s control_rda -d -l 5000 .
          cpcntalg  22008    0m    388K    46s cpcntalg
          crapeprd  22010    0m    476K    46s crapeprd
         data_qual  22012    0m    492K    46s data_qual
       dp_dua_accum  22014    0m    300K    46s dp_dua_accum -T dp_dua_.
       dp_elev_prod  22016    0m    296K    46s dp_elev_prod
        dp_lt_accum  22018    0m    292K    45s dp_lt_accum
      dp_precip_4bit  22020    0m    320K    45s dp_precip_4bit
      dp_precip_8bit  22022    0m    324K    45s dp_precip_8bit
       dp_s2s_accum  22024    0m    300K    45s dp_s2s_accum
            dpprep  22026    0m    284K    45s dpprep
          dqa_elev  22028    0m    296K    45s dqa_elev
         dualpol4bit  22030    0m    304K    45s dualpol4bit
         dualpol8bit  22032    0m    304K    45s dualpol8bit
          ecotppro  22034    0m    448K    45s ecotppro
         elev_prod  22036    0m    280K    45s elev_prod -T elev_prod
              epre  22038    0m    304K    44s epre
       hail_hazard  22040    0m    296K    44s hail_hazard
           hailalg  22042    0m    384K    44s hailalg
          hailprod  22044    0m    396K    44s hailprod
               hca  22046    0m    304K    44s hca
         hci_agent  22048    0m    512K    44s hci_agent
           hhc8bit  22050    0m    304K    44s hhc8bit -T hhc8bit
          hireseet  22052    0m    296K    44s hireseet
          hiresvil  22054    0m    296K    44s hiresvil
          hybrprod  22056    0m    316K    44s hybrprod
       icing_hazard  22058    0m   1156K    44s icing_hazard
           itwsdbv  22060    0m    316K    43s itwsdbv
             lcrap  22062    0m    432K    43s lcrap
            lcrappg  22064    0m    424K    43s lcrappg
            lcrflct  22066    0m    444K    43s lcrflct
        ldm_recomb  22068    0m    280K    43s recomb -T ldm_recomb -A.
             mda1d  22070    0m    304K    43s mda1d
             mda2d  22072    0m    576K    43s mda2d
             mda3d  22074    0m   1232K    43s mda3d
           mdaprod  22076    0m    300K    43s mdaprod
           mdattnn  22078    0m   1436K    43s mdattnn
              mlda  22080    0m   1204K    42s mlda
```

```
       mlprod  22082    0m    300K    42s mlprod -T mlprod
    mngdskerr  22084    0m    100K    42s mngdskerr
         mrpg  21891  770m    712K    60s mrpg -p -v startup
   nexradAmda  22086    0m   2200K    42s nexradAmda
  nexradMigfa  22089    0m   2736K    42s nexradMigfa -N
     ntda_alg  22092    0m    328K    42s ntda_alg
      ntda_fp  22094    0m    300K    42s ntda_fp
   owr_server  22096    0m    136K    42s owr_server -l
          pbd  22098    0m    368K    42s pbd -v -l 5000
     pcipdalg  22100    0m    312K    42s pcipdalg
     prcpadju  22102    0m    436K    42s prcpadju
     prcpprod  22104    0m    596K    41s prcpprod
     prcprtac  22106    0m    912K    41s prcprtac
     prcpuspt  22108    0m    448K    41s prcpuspt
      prfbmap  22110    0m    308K    41s prfbmap
     prfselect 22112    0m    312K    41s prfselect
    ps_onetime 22114    0m    220K    41s ps_onetime -v 3
    ps_routine 22116    0m    412K    41s ps_routine -v 3 -l500
          qia  22118    0m    284K    41s qia
      qperate  22120    0m   2252K    41s qperate
     radcdmsg  22122    0m    460K    41s radcdmsg -T radcdmsg
      recclalg 22124    0m    300K    41s recclalg
    recclprods 22126    0m    308K    40s recclprods
       recomb  22128    0m    288K    40s recomb -T recomb -l 1000
 replay_basrflct 22130  0m    304K    40s basrflct -T replay_basr.
 replay_basspect 22132  0m    300K    40s basspect -T replay_bass.
 replay_basvlcty 22134  0m    304K    40s basvlcty -T replay_basv.
  replay_bref8bit 22136 0m    308K    40s bref8bit -T replay_bref.
  replay_bvel8bit 22138 0m    304K    40s bvel8bit -T replay_bvel.
  replay_cmprfcg 22140  0m    476K    40s cmprfcg -T replay_cmprf.
 replay_dp_dua_accum 22142 0m 284K   40s dp_dua_accum -T replay_.
 replay_radcdmsg 22144  0m    460K    40s radcdmsg -T replay_radc.
  replay_srmrmrv 22146  0m    312K    39s srmrmrv -T replay_srmrm.
 replay_user_sel_LRM 22148 0m 1924K  39s user_sel_LRM -T replay_.
     replay_vad 22150   0m    444K    39s vad -T replay_vad
  replay_vertxsct 22152 0m    468K    39s vertxsct -T replay_vert.
        rpgdbm  22156   0m    360K    39s rpgdbm -v
       saaprods 22158   0m    328K    39s saaprods
       saausers 22160   0m    324K    39s saausers
       segmtalg 22162   0m    436K    39s segmtalg
      snowaccum 22164   0m    636K    39s snowaccum
    sr_elev_prod 22166  0m    280K    38s elev_prod -T sr_elev_pr.
        srmrmrv 22168   0m    324K    38s srmrmrv -T srmrmrv
    status_prod 22170   0m    300K    38s status_prod
        stmtrprd 22172  0m    400K    38s stmtrprd
        strucprod 22174 0m    400K    38s strucprod
       superes8bit 22176 0m   304K    38s superes8bit
       superob_vel 22178 0m   300K    38s superob_vel
          tda1d  22180   0m    432K    38s tda1d
        tda2d3d  22182   0m    416K    38s tda2d3d
       tda2d3dru  22184  0m    416K    38s tda2d3dru
        tdaruprod  22186 0m    336K    38s tdaruprod
        trfrcalg   22188 0m    372K    37s trfrcalg
         tvsprod   22190 0m    452K    37s tvsprod
  update_alg_data  22192 0m    232K    37s update_alg_data
     user_sel_LRM  22194 0m   1924K    37s user_sel_LRM -T user_se.
            vad   22196  0m    448K    37s vad -T vad
```

Vol 1 Appendix J. Outputs of rpg_ps

```
           veldeal  22198   0m   380K   37s veldeal -I
          vertxsct  22200   0m   460K   37s vertxsct -T vertxsct
          viletalg  22202   0m   424K   37s viletalg
           vilprod  22204   0m   444K   37s vilprod
          vwindpro  22206   0m   340K   37s vwindpro
    wideband_agent  22208   0m   404K   37s wideband_agent
RPG: Operating state - Active - In Operational Mode
code17_0r1_13:/home/code17_0r1_13:  30>
```

Appendix K. Software Removed for the Public Edition

Differences between the U.S. Government and Public Editions of CODE

The significant difference between the U.S. Government Edition and the Public Edition of CODE is the removal of certain proprietary software components in the Public release. The source code archive provided with the Public Edition has been modified to eliminate this software and the filename changed to include the term "pub" for public (e.g., `rpg_b##_#r#_##_pub_src.tgz`) in order to identify the correct archive.

Currently, 8 operational tasks have been removed from the NWS Edition. A summary of the software removed is contained in the following table.

Operational Processes Removed for the Public Edition					
Source Code Directory	Executable Task Name	Product Name	ID	Product Description	Source
`cpc010`	`nexradMigfa`	`MIGFA`	`140`	GFM Gust Front MIGFA	MIT/LL
`cpc022/tsk001`	`ntda_alg`	`NTDA_EDR_IP` `NTDA_CONF_IP`	`315` `316`	NTDA EDR Intermediate Prod NTDA CONF Intermediate Prod	NCAR
`cpc022/tsk002`	`ntda_fp`	`NTDA_EDR` `NTDA_CONF`	`156` `157`	NTDA EDR Final Product NTDA CONF Final Product	NCAR
`cpc022/tsk003`	`data_qual`	`DQA`	`297`	Edited Reflectivity Data	MIT/LL
`cpc022/tsk004`	`hiresvil`	`HRVIL`	`134`	High Resolution Digital VIL	MIT/LL
`cpc022/tsk005`	`hireseet`	`HREET`	`135`	Enhanced Echo Tops	MIT/LL
`cpc022/tsk007`	`icing_hazard`	`IHL`	`178`	IHL Icing Hazard Level	MIT/LL
`Cpc022/tsk008`	`hail_hazard`	`HHL`	`179`	HHL Hail Hazard Layer	MIT/LL