A-4. Getting Started

ob8.2

No changes were made in build ob8.2 that impact this section. This section remains valid for build ob8.1

ob7.2

This section has been written specifically for build ob7.2. The following steps assume that the RAX has been incorporated into your AWIPS system, the OS and Postgres RDBMS are installed, Postgres is running, the RFC Archive DB has been created, the flatfiles archive directory structure has been defined, oper's cron is running, and the data feed to the shef_decode_raw incoming directory has been set-up. If one or more of these assumptions is not true, contact the RFC Support Group.

Step 1:   Getting familiar with the RAX

Review the information in sections A-1 thru D-2.

Step 2:   Setting up the proper user environment

Add the following lines to each user's .profile file:

```
DEVn=`hostname | cut -c1-2`
echo $DEVn
if [ "$DEVn" = ax ]
then
. /rfc_arc/lib/rax.profile
stty erase ^?
fi
```

Adding these lines to the user's .profile sets up the proper user environment each time the user connects to the RAX (ax-xxx where xxx is the office's 3-char identifier). The rax.profile script is new in build ob7.2, see section L-3 for more information about it.

Step 3:   Log on to the RAX

Open a terminal window, and at the prompt, type:

```
ssh ax
```
(press the [Enter] key)

This connects the user to the RAX.

Step 4:   Check RFC Archive DB Pre-defined Tables

These tables should have been defined in build ob1 as part of the original install process. However, it has been discovered that at least one RFC somehow missed this step in ob1, so it is recommended each RFC double check that the tables were populated. The tables are:

Reference Data Tables

country	state	counties	huc2	huc4	huc6
huc8	wfo_hsa	rfc	shefdur	shefex	shefpetrans
shefpe	shefpe1	shefprob	shefqc	shefts	

Quality Control Table   NWSRFS Table
Datalimits   modctrl

Appendix B provides detailed information about most of the data used to define these tables. The easiest way to see if these tables have data is to use the database utility adbpg.pl, (see section J-3 for instructions). Use the adbpg.pl application to query each of the tables listed above. If zero rows are returned for any of these tables, contact the RFC Support Group.

Step 5:   Defining the RFC Archive DB's location and ingestfilter tables

Assuming the RFC has been actively using their RAX, the *location* and *ingestfilter* tables in the RFC Archive DB should already be defined. These tables need to be defined prior to turning on the shefdecoders (shef_decode_raw and shef_decode_pro). If these tables do not contain any data, the user must run the adbinit suite to populate the tables. See section E-1 for instructions.

Step 6:   Updating the RFC Archive DB's location and ingestfilter tables

Currently there is no user-friendly way to keep the RFC Archive DB's *location* and *ingestfilter* tables in-sync with the IHFS DB's *location* and *ingestfilter* tables. A HOSIP project is in development, and is tentatively designated for the build ob8 series. This project, "Synchronize Archive (RAX) DB with IHFS DB Metadata", will provide robust tools for syncing not only the *location* and *ingestfilter* information, but other meta-data as well. In the meantime, the user has two avenues, use the adbinit suite to pick up new entries only, and/or use the database utility adbpg.pl to make additions to these

tables. See sections E-1 and J-3, respectively, for more information on these applications.

Note: If using the adbinit suite, it is suggested that the shefdecoders be shut down while the tables are being updated. This will speed up the inserting of new rows into these tables. The shefdecoders will need to be restarted after the update is complete. See the next step and sections F-1 and F-2 (sub-section 3.0) for further information.

Step 7: Turning on/off the shefdecoders

In order to ingest data into the RFC Archive DB's SHEF Data Value tables, the shefdecoders must be running. At a minimum the shef_decode_raw application needs to be turned on and running in background. Running shef_decode_pro is optional and depends on whether the user is running the level 1 processors in oper's cron. See section G2 for more information about the level 1 processors.

Default values of several apps_defaults tokens have been defined in the national .Apps_defaults file. If the user wishes to use a different value than the default, changes should be specified in the file .Apps_defaults_site. The .Apps_defaults and .Apps_defaults_site files are located in the directory /awips/hydroapps.

For detailed information about both shefdecoders, including instructions for starting and stopping the shefdecoders, see sections F-1 and F-2.

Step 8: Cron Jobs

Section B-5 provides a listing of oper's cron file, which resides in the /rfc_arc/crons directory in the file adb_oper_crons. Processes listed in section `#1# DB and System admin scripts` **must** **not** be removed or commented out. However, processes listed in section `#2# db data processes` are optional. If your office does not wish to run the level 1 processors as a cron job, then comment the entry out. In order for any changes to take effect, the user (as user oper) must stop and restart the cron.

Note: If the level 1 processors are not run routinely then the shef_decode_pro application does not need to be running in background.

Step 9: Accessing the Applications

Most applications and scripts can be run through arcmenu (see section I-1) or individually on the command line, these applications/scripts are:

DatView   ff_oper_view.tcl   display_rc

arnav	process_temp	run_vacuumdb
ff_oper_viewX.tcl	process_sw	run_PGbkups
adbpg.pl	process_flow	status_decoders
dcextract	transfer_txn	start_raw_decoder
dcparse	transfer_precip	start_processed_decoder
nrcsdlyparse	slope_to_stage	stop_raw_decoder
usgsdlyparse	find_bigfiles	stop_processed_decoder
process_precip	rax_status_df	
process_stage	purge_files	

Some applications and scripts are not available via arcmenu and can only be run on the command line, these are:

arcmenu	get_params	run_level1_process
rax.profile	loadmods	run_radarlist
check_raw_decoder	fetchmods	cron_PGbkups
check_processed_decoder	group_parse, del_parse	dump_script
ofsshef	pgload	restore_script
get_states	log_stats.tcl	run_bounce_postgres

For instructions on individual applications and scripts, see the appropriate section.

Step 10:   Flatfiles Archive

The NWS Directive 10-911 section 4.8 outlines the archiving responsibilities of the RFCs (see section A-3). A separate effort under the OHD RDM's direction provided for the conversion of the scripts that OHD used to run at each RFC and the sharing of local scripts that move these and other files to the RAX. There are currently no RAX baseline scripts to archive, i.e. move, the required flat files, the RFCs are responsible for putting the needed script(s) in place to ensure the required archiving requirements of NWS Directive 10-911 are being fulfilled.

A script for monitoring the flatfiles portion of the archive is under development and will be shared with the RFCs once it becomes available.

Section D-2 provides a chart of the current flatfiles directory structure. Most files stored in the flatfiles archive can be viewed with either the flatfile viewer or arcnav. See sections J-1 and J-5, respectively.

Step 11:   Backups

It is the responsibility of **each** RFC to make and store routine, complete backups of their RAX so that recovery of files and the database will not be compromised.

_____

Sections M-1 thru M-3 cover the scripts provided for backing up and restoring the underlined database. Note that the database backup scripts dump the database to a file and this file is written to the RAX. To complete the database backups process, this file must be saved off to tape or another system to ensure that the data will not be lost. This can be done separately, or as part of the file system backups.

Section M-7 provides information on one method for performing backup and recovery for the file system. An RFC can choose to use an alternative backup and restore procedure instead of the scripts that have been provided. Just make sure that your RFC is doing file system backups, and that you can successfully restore with whatever procedure is utilized.

It is extremely important that each RFC perform routine backups of their RAX database and files system.

_____