7/31/2013

A WIPS II

S ITE

M IGRATION

G UIDE

S ITE P REPARATION &

S YSTEM I NSTALLATION

Version 17

AWIPS II Site Migration Guide
Version 17

31 July 2013

Prepared By

Raytheon

Raytheon Technical Services Company LLC
8401 Colesville Road, Suite 800
Silver Spring, MD 20910

Under Contract DG133W-05-CQ-1067
Advanced Weather Interactive Processing System (AWIPS) Operations and Maintenance

Prepared For

U.S. Department of Commerce
NOAA NWS Office of Science and Technology
Programs and Plans Division, Program Management Branch
SSMC2, OST11, Room 15130
1325 East-West Highway
Silver Spring, MD 20910

TABLE OF CONTENTS

APPENDIX A. ADDITIONAL RESOURCES 1

LIST OF TABLES

Page

1. INTRODUCTION

Welcome to AWIPS II. This Site Migration Guide is a compilation of the work of several people in the National Weather Service and Raytheon. It is intended to be a comprehensive single source for migrating a site to AWIPS II, and it provides approaches, step-by-step procedures, tools, automation, and notes. It contains information from the NWS Migration Wiki, NWS documents, and Raytheon documentation. The ADAM SDC (AWIPS II Data & Applications Migration Site Data Configuration) Automation Tool has been extended by automating previous manual steps and including other automation tools from the NWS. The goal has been to minimize the effort needed to complete a site's migration by automating as much as possible; providing manual procedures that work; and providing a single source for related information.

There are many ways to accomplish similar tasks, but the time required to research, select, and test the various procedures and tools that are current for the release that you receive can be significant. This document provides a single good way to do it, but not the only way.

The Site Migration Guide is revised as needed with each new software version release. **This version was issued with the release of Operational Build 13.4.1.**

A. Additional Resources

Provided with this Migration Guide is a package of additional resources that may become useful resources as your site cuts over to AWIPS II. Appendix A provides a complete list of these resources.

B. Type Conventions Applied in This Guide

`user input`	information to be entered **(Courier New Bold)**
`/awips2/httpd_pypies/` `/awips2/httpd_pypies/`	Example (Courier New, shaded)
/awips2/edex/data/utility/edex_static/site/OAX/shapefiles/OAX_County/OAX_County.shx	File name **(Time New Roman Bold)**

2. GENERAL GUIDELINES, INCLUDING LIMITATIONS AND MITIGATIONS

The most effective approach for migrating to AWIPS II is to perform the following steps **in the order shown**.

1. Install the version of ADAM delivered with the release to be cut over to operations. There is a field mod note for this action.

2. Set up all appropriate users on ADAM so that migration of AvnFPS transmit configuration succeeds.

3. Use the ADAM SDC Automation Tool to establish the base configuration.

4. Test the localization just created with the SDC Automation Tool. Find and fix any localization errors before moving on. At this stage you can assume there are no AWIPS II software problems. If you do encounter a problem that you cannot fix, call the Network Control Facility (NCF) (301.713.9344).

5. Set up Subversion (SVN) on ADAM to facilitate converting local apps and Smart Tools. It is much easier to retrieve and install these items using SVN than without it, as you will see later on.

6. Complete the steps in this document. Test as you go to limit the source of errors as much as possible; this will help you avoid losing time troubleshooting a problem. Correct the errors before moving on unless the problem requires a lot of time to resolve (e.g., a smart tool, or a local app). For a time-consuming problem, record where you are and the ticket number provided by the NCF (see step 3).

7. When the localization, configuration, and customizations are completed, test the AWIPS II applications on ADAM. Some testing will already have been done in checking the localizations. At this point, specific site product generation can be tested (e.g., GFE, WarnGen). Test other job functions. Let forecasters test drive it. Learn the variances, etc. Do as much as you can on ADAM (see "ADAM Testing Limitations and Mitigations), including local application testing and migration. Try to keep local app testing and AWIPS II testing separate in order to control the source of error (i.e., is it an AWIPS II problem or a smart tool problem?).

   The more you complete with ADAM, the quicker and smoother the task of going live will be.

8. Coordinate with your Raytheon installation point of contact a week prior to your installation date regarding pre-installation steps.

9. Follow the steps in Section 4, "System Installation" unless Raytheon will be installing the software.

A. ADAM Testing Limitations and Mitigations

1. Rehosted Applications

   Rehosted applications are the same apps used in AWIPS I. They include Climate, HWR, FSI, AsyncScheduler, MHS, NWRWAVES, Archive GUI, internal LDAD processes, etc. The apps are same across the sites and run with little or no adjustments to their environment variables. They have been tested in Raytheon labs and in FITs with no issues.

   Test during pre-cutover testing, after AWIPS II has been installed.

2. Service Backup

   Tested with other sites, no issues currently.

   However, upload and download to central service can be tested after AWIPS II is installed, before exiting Service Backup.

3. ISC

   Testing to date has shown no issues.

   Test after AWIPS II has been installed, before exiting Service Backup if desired.

4. ORPG

   ORPG (Open Radar Products Generator) cannot be connected to ADAM; thus dedicated radar operations cannot be tested where the following are concerned:

   – RPS list creation and send from CAVE
   – Data ingest from RPG
   – AlertViz notifications of RPG events (i.e., need maintenance or system errors)
   – Automatic RPS list send on VCP mode change
   – One-Time Request functionality (OTR) and radar multiple request functionality (RMR)
   – Sending bias and environmental data to the RPG
   – National reporting of radar files to the NWS Telecommunications Gateway (NWSTG) (of course, this **also** requires Message Handling Service (MHS), which is not on ADAM).

This should be tested prior to exiting Service Backup at cutover.

B. Pre-Installation Checklists

The checklists provided here as **Tables 2-1** and **2-2** may be useful aids in your efforts to ensure that maximum migration has been completed and tested prior to installing AWIPS II on the AWIPS system. **Table 2-1** relates to overall migration; **Table 2-2** is specific to the testing of various WarnGen templates. [**Note:** These checklists can be expanded to meet the needs of each forecast office. To allow for easy expansion, copies of both checklists are included in Excel format in the package of additional resources identified in Appendix A (see "Organization Aids" folder).]

Table 2-1. Overall Migration Checklist

	Test Loc	OPR	Comments
Training			
User			Distance learning can do after start
Application Focal Points			Distance learning can do after start
ITO / ESA			12 Working Days residence course, do before start
Local Systems Prep		Site	
ADAM Installed			
ADAM SDC Auto Tool			
SVN			
WarnGen Templates			
D2D Procedures			
GFE SmartTools, etc.			
GFE Formatters			
Local Apps			
LDAD			
AWIPS II on ADAM Testing		Site	
Functional			
Mission			
AWIPS II Installation		RTN	RTN considers it ready for cutover
Pre-Cutover User Test		Site	**Some testing can be done concurrently**
D2D, radar, GFE, etc.			
Other			
Enter Service		MIC	Exit Service Backup

Table 2-2. WarnGen Template Checklist

Template	WHO	Date	Shift	P/F	Comments
Tornado Warning					
Severe Thunderstorm Warning					
Severe Weather Statement, follow-up to TOR or SVR					
Extreme Wind Warning					
Severe Weather Statement, follow-up to EWW					
Special Marine Warning					
Marine Weather Statement, follow-up to SMW					
Standalone Marine Weather Statement					
Standalone Marine Weather Statement for Ashfall					
Flash Flood Warning, convective					
Flash Flood Statement (follow-up to convective FFW)					
Dam Break (nonconvective flash flood warning)					
Dam Break follow-up					
Areal Flood Warning					
Areal Flood Statement (follow-up to FLW)					
Areal Flood Advisory (events below severe criteria)					
Areal Flood Advisory Statement (follow-up to areal flood advisory)					
Significant Weather Advisory					
Short Term Forecast					
Special Weather Statement					
ER Special Weather Statement					

3. SITE PREPARATION

A. Run AWIPS II SDC Automation Tool on the ADAM Platform

System *root* access is required on ADAM and AWIPS to perform some of the steps in this section. Prior to performing these steps, ensure that ADAM and the SDC Automation Tool are at the current release level.

1. Log into DX1 as user *root*.

2. Download the latest baseline shapefiles to /data/fxa/nationalData. See AWIPS1 SMM Chapter 15 and section 5.C.4.b in this document for details. Copy the files you want to use to adam1:/data/fxa/nationalData using the following as an example.

   ```
   # scp /data/fxa/nationalData/c11-zone*
   adam1:/data/fxa/nationalData
   ```

   Repeat for each shapefile that was downloaded.

3. Change directories into the ADAM installation directory and untar the collect_files.sh script.

   ```
   # cd /data/fxa/INSTALL/ADAM
   # tar –xzf AWIPSII_AUTOMATION_TOOL.tgz
   collect_files.sh vmgfe/GFE_unmangler.py
   ```

4. Run the collect_files.sh script.

   ```
   # ./collect_files.sh LLL
   ```

   Where LLL is the AWIPS ID of the site to be localized by the script.

5. Copy the file created by collect_files.sh and the automation tool bundle to the ADAM platform, and then secure shell into the adam1 platform.

   ```
   # scp /data/fxa/TEMP/awipsIloc_lll.tar.gz
   adam1:/var/tmp/
   # scp
   /data/fxa/INSTALL/ADAM/AWIPSII_AUTOMATION_TOOL.tgz
   adam1:/var/tmp/
   # ssh adam1
   ```

   Where lll is the lower-case AWIPS ID of the site to be localized by running the script.

6. Unpack the tar bundle created with collect_files.sh, and install the automation tool into the proper locations.

```
# mkdir –p /var/tmp/sdc
# tar –C / -xzf /var/tmp/awipsIloc_lll.tar.gz
# tar –C /var/tmp/sdc –xzf
/var/tmp/AWIPSII_AUTOMATION_TOOL.tgz
```

Where lll is the lower-case AWIPS ID of the site to be localized by running the script.

7. Stop the EDEX JVMs, and ensure that the database, qpidd, and http-pypies are still running.

```
# service edex_camel stop
# ps –fu awips
```

EXAMPLE. This is an example of what you might see after the previous command. It is important to make sure postgres, qpidd, and httpd-pypies are running.

```
/awips2/httpd_pypies/usr/sbin/httpd –f
/awips2/httpd_pypies/etc/httpd/conf/httpd.conf
/awips2/httpd_pypies/usr/sbin/httpd –f
/awips2/httpd_pypies/etc/httpd/conf/httpd.conf
/awips2/httpd_pypies/usr/sbin/httpd –f
/awips2/httpd_pypies/etc/httpd/conf/httpd.conf
/awips2/httpd_pypies/usr/sbin/httpd –f
/awips2/httpd_pypies/etc/httpd/conf/httpd.conf
/awips2/httpd_pypies/usr/sbin/httpd –f
/awips2/httpd_pypies/etc/httpd/conf/httpd.conf
/awips2/httpd_pypies/usr/sbin/httpd –f
/awips2/httpd_pypies/etc/httpd/conf/httpd.conf
/awips2/httpd_pypies/usr/sbin/httpd –f
/awips2/httpd_pypies/etc/httpd/conf/httpd.conf
/awips2/httpd_pypies/usr/sbin/httpd –f
/awips2/httpd_pypies/etc/httpd/conf/httpd.conf

/awips2/python/bin/python /awips2/python/lib/python2.7/site-
packages/pypies/logging/logProcess.py

/awips2/qpid/usr/sbin/qpidd --daemon --auth=no --worker-threads 20 --pid-
dir /awips2/qpid/var/run/qpidd

/awips2/postgresql/bin/postmaster –D /awips2/data –p 5432
postgres: logger process
postgres: writer process
postgres: wal writer process
postgres: autovacuum launcher process
postgres: stats collector process
```

If the above <u>commands are not running, issue the following</u>. Then retry the ps command:

```
# service adam_servers start
# service edex_camel stop
```

8. To migrate local shape files, stage them to the directory
/awips2/edex/data/utility/edex_static/site/LLL/shapefiles

The shapefiles should be added in a manner similar to the following:

submenu/shape_desc/shapefile.(dbf|shp|shx)

It is important to note a few things about this structure.

– First, the name of the shape file is determined by its parent directory and NOT by the name of the shapefile. Also, all underscore characters "_" will be translated into spaces in the CAVE menu.

– Second, the directory name of *SUBMENU* is optional, but if desired this will create a submenu on your Maps menu in CAVE. This means that you can have a shapefile staged in this manner:

/awips2/edex/data/utility/edex_static/site/LLL/shapefiles/shape_desc/shapefile.shp

And it will create a menu entry in CAVE under **Maps** named **shape_desc**.

Any shapefiles staged within more than one directory will be listed under a submenu in the Maps pulldown in CAVE that is named the same as the directory itself. To clarify, whatever text you put in for *submenu* above will be the text shown as the submenu on CAVE.

For example, the following staging:

/awips2/edex/data/utility/edex_static/site/LLL/shapefiles/local_roads/dirt_roads/dirtroad.shp

/awips2/edex/data/utility/edex_static/site/LLL/shapefiles/local_roads/grass_roads/grassroad.shp

will create the following structure in the menu of CAVE:

```
Maps → local roads → dirt roads

                   → grass roads
```

Finally, it is important to note that only ONE **shp, dbf**, and **shx** file can reside in the final directory for staging shapefiles. This is **NOT** valid staging:

/awips2/edex/data/utility/edex_static/site/LLL/shapefiles/city_parks/playgrounds.shp

/awips2/edex/data/utility/edex_static/site/LLL/shapefiles/city_parks/bigfieldsa.shp

because there are two **.shp** files under **local_roads**, which would be used to create the "city parks" menu. In this case you probably want to create two subdirectories under **city_parks** – one named **playgrounds/** and another **bigfields/**.

The last important thing to note is that the directory name of *shape_desc* will determine the table name into which the shapefiles will be imported. It also determines the name of the map bundle created. The name of the map bundle is case sensitive, while the table name will always be all lower case. All underscore characters ('_') in *shape_desc* will be converted to spaces when creating the Maps menu name.

For example, the following shapefiles will create **mapdata.oax_county** schema in the maps database and create a new entry *OAX County* on the main Maps menu in CAVE :

```
/awips2/edex/data/utility/edex_static/site/OAX/shapefiles/O
AX_County/OAX_County.dbf
```
```
/awips2/edex/data/utility/edex_static/site/OAX/shapefiles/O
AX_County/OAX_County.shp
```
```
/awips2/edex/data/utility/edex_static/site/OAX/shapefiles/O
AX_County/OAX_County.shx
```

In this example, the shapefiles will create **mapdata.grr_cwa_roads** schema in the maps database, and will create a *Roads* submenu in the Maps pulldown within CAVE. Under the *Roads* submenu will be a selection *GRR CWA Roads*.

```
/awips2/edex/data/utility/edex_static/site/GRR/shapefiles/R
oads/GRR_CWA_Roads/grr_cwa_roads.dbf
```
```
/awips2/edex/data/utility/edex_static/site/GRR/shapefiles/R
oads/GRR_CWA_Roads/grr_cwa_roads.shp
```
```
/awips2/edex/data/utility/edex_static/site/GRR/shapefiles/R
oads/GRR_CWA_Roads/grr_cwa_roads.shx
```

Do this for each local shapefile you wish to migrate into the AWIPS II database.

Note: If you change the structure of edex_static/site/LLL/shapefiles, you may have to remove old map bundles manually. You can do this through the localization perspective.

9. Migrate the patterns that are in your **acqPatternsAddOns.txt** file to this file:

/usr/local/ldm/etc/pqact.conf.lll

Where lll in the file name is your AWIPS site identifier.

Two files – **LDM Ingest Checklist.docx** and **Ingest Filter Configuration.docx** – have been included with the additional resources listed in Appendix A to help with this stage of migration. [See the "LDM Ingest" folder.]

Note for RFCs: To archive shef data to the AX, you will need to migrate entries that will continue to write raw data to /data/fxa/ispan/hydro_adbs. However, you want to omit the –edex flag in those ldm entries to prevent that data from being ingested again by EDEX.

10. Run the automation tool:

```
# cd /var/tmp/sdc
```

Note: The initial site data configuration should be run with the following syntax, and you should answer Y to all prompts:

```
./config_awips2.sh db LLL
./config_awips2.sh shp LLL
./config_awips2.sh ldm LLL
./config_awips2.sh edex LLL
./config_awips2.sh cave LLL
```

Note: If you already have a localization that you are trying to override, pass the -f force option to the script as shown below:

```
./config_awips2.sh -f db LLL
./config_awips2.sh shp LLL
./config_awips2.sh -f ldm LLL
./config_awips2.sh -f edex LLL
./config_awips2.sh -f cave LLL
```

Where LLL is the AWIPS ID of the site to be localized by the script.

11. Modify the site-level shef distribution file. Without this, ldm will capture the data and send it to EDEX, but EDEX will not recognize it as shef data and will drop it. Edit **/awips2/edex/data/utility/edex_static/site/LLL/distribution/shef.xml**, adding regex rows for each site-specific shef data added to your **pqact.conf.lll** file in step 8. For example, if you want to ingest SXUS44 KWOH data, then add a line below the last regex line:

```
<regex>^SXUS44 KWOH</regex>
```

12. Once the script is complete, make the changes to the localConfig.py as shown in **Table 3-1**, and then copy the file into place.

Table 3-1. Necessary localConfig.py Changes

AWIPS I Token	AWIPS II Equivalent	Description

serverConfig. D2DDIRS	serverConfig. D2DMODELS	Change all instances of D2DDIRS to D2DMODELS and change the arguments passed from the AWIPS I netCDF directory to the AWIPS II dataURI model name. Example:
		
		CHANGE
		`serverConfig.D2DDIRS.append(('/data/fxa/Grid/netCDF/CONUS212/SREF/','SREF'));`
		
		TO
		`serverConfig.D2DMODELS.append(('SREF212', 'SREF'));`
		
		You can retrieve the model name from the grib_models table in the metadata database with the following SQL statement:
		
		`select distinct(modelname) from grib_models;`
		
		or from the AWIPS II files with:
		
		`/awips2/edex/data/utility/common_static/base/grid/*.xml`

Any local model coming into AWIPS I	Model name may change based on addition of product into AWIPS II	If any local model is sent into GFE in AWIPS I, this model will first have to be added into AWIPS II in order to utilize GFE's localConfig.py. Immediately after copying this file, comment out any entries pertaining to local model data until these data sets are added into the system. See the table identified as Table 5.3-1 in the Additional Resources provided with this Guide for baseline model name reference (see Appendix A).

> IMPORTANT NOTE: Do not add declarations to localConfig.py that override system variables. Now that GFE processing has been incorporated into EDEX, these declarations could negatively affect the behavior of multiple systems. For example, do not set TZ inside localConfig.py, as it will have an adverse effect of the behavior of WarnGen.

```
# cp /awips/GFESuite/primary/etc/SITE/localConfig.py
/tmp/
# vi /tmp/localConfig.py
```

Make the necessary changes here. When complete, exit the vi editor:

```
# :wq!
# cd
/awips2/edex/data/utility/edex_static/site/LLL/config/gfe
```

Where LLL is the AWIPS ID of the site to be localized by the script.

```
# cp /tmp/localConfig.py .
# chown awips:fxalpha localConfig.py
# chmod 775 localConfig.py
```

EXAMPLE.

The following is an example of converting a line from AWIPS I to AWIPS II syntax.

```
1.  AWIPS I Entry:

##############################
###  Add RUC40 database   ###
serverConfig.D2DDIRS.append(('/data/fxa/Grid/SBN/netCDF/Grid236/RUC2/',
'RUC40'))

serverConfig.INITMODULES["myRUC40"] = ["RUC40"]

RUC40 = ('RUC40', GRID, '', NO, NO, 2, 0)

RUC40_Parms = [([QPF, SnowAmt], LT6NG),
               ([Haines], LT3NG),
               ([Temp, Td, Wind, MixHgt, TransWind, FzLevel, RH, Sky], TC1),
               ([MaxT], MaxTTC), ([MinT], MinTTC),
               ([MaxRH],MaxRHTC),([MinRH],MinRHTC)]
```

```
2.  Information Gathering For AWIPS-II:

# grep -B1 RUC236 $EDEX_CONFIG/common_static/base/grid/models/*.xml

              <title>RUC40</title>
              <name>RUC236</name>
```

```
3.  AWIPS II Entry:

##############################
###  Add RUC40 database    ###
serverConfig.D2DMODELS.append(('RUC236','RUC40'))

serverConfig.INITMODULES["myRUC40"] = ["RUC40"]

RUC40 = ('RUC40', GRID, '', NO, NO, 2, 0)

RUC40_Parms = [([QPF, SnowAmt], LT6NG),
               ([Haines], LT3NG),
               ([Temp, Td, Wind, MixHgt, TransWind, FzLevel, RH, Sky], TC1),
               ([MaxT], MaxTTC), ([MinT], MinTTC),
               ([MaxRH],MaxRHTC),([MinRH],MinRHTC)]
```

13. Start EDEX.

   # service edex_camel start

14. Tail the log file to make sure that EDEX starts successfully:

   **# tail –f /awips2/edex/logs/edex-ingest-$(date
   +%Y%m%d).log**

Note: Site-specific map scale bundles (e.g., PR Mercator scale) that are not standard (e.g., Regional, State, WFO for CONUS) need to added manually to AWIPS II. See Appendix C of the *AWIPS II Site Data Configuration & Localization Step-by-Step Guide* for information on how to create and add these map bundles. For a copy of the *Step-by-Step Guide*, see Appendix A of this document ("Localization Doc References" folder).

B. Test Base Localization

If you are following this document step by step, then you are ready to test the base localization.[1]

The following items should be configured during the base localization run:

- *Plugin filters*, which filter which products are sent to the shef decoder, the obs decoder, and the metar decoder respectively based on geographic latitude/longitude location.
- *GOES and POES spi files*, which should be copied into place for decoding and displaying of GOES and POES sounding and sounding availability.
- *Hydrology site-level Apps_Defaults file*.
- *Hydrology site-level ascii and binary geo_data*.

---

[1] When starting ADAM to test localizations, the EDEX server needs to be running in order to start AlertViz or Cave (click on the EDEX desktop icons to stop and start EDEX). AlertViz needs to be started prior to starting CAVE (click on the AlertViz desktop icons to stop and start EDEX). The first time you do this, you will have to add your SITE ID, e.g., BOU. When done, click Validate. All red should disappear.

- *MPE msbin, prism, and utiltriangles files.*
- *XDAT parameter groups* (RFC Only).
- *Hydro timeseries group definitions files.*
- *Shef decoder metar.cfg file.*
- *GFE siteConfig.py file.*
- *FFMP runtime configuration file.*
- *All national data files,* which should be copied into place.
- *Distribution XML files* (these files configure which products are addressed by which decoding plugins within EDEX).
- *Grid geospatial information files for MPE and DHR-MOSAIC grids,* which allow display through Volume Browser in CAVE.
- *Radar RMR definition file migration.*
- *Spotters.dat migration into spotters.xml.*
- *WarnGen* (backup site pulldown, basic site location text for generated products).
- *Dedicated radar menus,* which should appear as they do in AWIPS I D2D.
- *Dial radar menus,* which should appear as they do in AWIPS I D2D.
- *Mosaic radar menus,* which should appear as they do in AWIPS I D2D.
- *Upper Air menus.*
- *AvnFPS.*
- *Default scales* shown when D2D perspective launches.
- *Scale selector pulldown* in D2D perspective.
- *Local warnings display* in D2D perspective.
- *Satellite display* for OCONUS localizations.
- *Products menu* in Text Workstation application.
- *cities.lpi, ldad15.spi, ldad15prcp.spi files,* which should be copied into place.
- *Text alarm/alert configurations* per workstation migrated.
- *Volume Browser fields menu.*
- *PostgreSQL configuration and database import.*
- *FFMP shapefiles import.*
- *Local shapefiles import.*
- *Triggers import.*

Before you begin, check to be sure the EDEX server is running. The EDEX server needs to be running in order to start AlertViz or CAVE.

With EDEX running, start CAVE and test the localization. Then use the following checklist to verify migration.

Perspective	Migration Area To Verify	Migration Verified? (✔)
D2D Perspective	Default Maps Loaded on launch	
D2D Perspective	Scale selector pull-down correct.	
D2D Perspective	Dedicated radar menus correct	
D2D Perspective	Dial radar, Mosaic, SCAN and FFMP menus correct	
D2D Perspective	Upper Air menus correct	
D2D Perspective	Families menu correct	
D2D Perspective	Volume Browser fields menu correct	
D2D Perspective	WarnGen launches	
D2D Perspective	WarnGen templates verified.	
D2D Perspective	Verify that Spotters map is correct	
Hydro Perspective	Perspective launches with proper gauges	
Hydro Perspective	Point Data Control launches	
MPE Perspective	Perspective launches	
GFE Perspective	Perspective launches with active site.	
AvnFPS	AvnFPS launches with correct configuration	
AvnFPS	Transmit button active for appropriate users.	

Note: The **config_awips2.sh** automation tool will attempt to look at the AWIPS I forecasters list for AvnFPS and match each of the forecaster names in that file to a forecaster's Linux system login name in the **/etc/passwd** file. Because AvnFPS in AWIPS II is tied into this user name, in order for the Send button to be activated an entry must be in the file with the correct user name.

During the run of the **config_awips2.sh** script, the individual running the tool will be notified if the tool could not migrate one of the users in the AWIPS I AvnFPS forecasters list to the AWIPS II AvnFPS forecasters list. These forecasters can be added manually through the AvnFPS program itself or through the localization perspective in CAVE.

Note About Dual Polarization Radar: If your site has done the dual pole migration on your dedicated radar, an additional step is necessary to see the proper menu files. If you have completed the dual pole modifications to the radar, then issue these steps on your ADAM:

```
su - awips
cd /awips2/edex/data/utility/common_static/base/radar
./dualPolTurnKey.sh new
```

Next, test to determine whether the AWIPS I **afos2awips.txt** file has been successfully integrated into the AWIPS II database with the following commands:

```
su – awips
psql –d fxatext –c "select * from afos_to_awips;"
```

This looks for site-specific products that would normally be issued from your office. Ensure that all products are present.

C. Set Up SVN (Subversion) on ADAM (Recommended)

1. Subversion How To[2]

The first place to go for help should be the "svn-book"[3] in this package (see Appendix A, "Subversion" folder). By far, the most important single step you can take to learn subversion is to set aside some time to read https://collaborate.werh.noaa.gov/wiki/index.php/SubversionHelp.[4] It provides an exceptional introduction. Nothing that follows in this section is a substitute for reading and performing the tutorials on that page. HowToSubversion (also in the "Subversion" folder) and includes the following sections:

1. The Repository
2. First Contact
3. Setting up Subversion Environment
4. Where Should I Put My Files?
5. Initial Import of an Application
6. Updating an AWIPS II Local Application
7. Downloading an Application for Operational Use

---

[2] Copied from the NCLADT Wiki.
[3] There is an HTML version and a PDF version for SVN 1.7.
[4] Included in Subversion folder of additional resources cited in Appendix A as three files: SubversionTutorialPart1 - Erh-itproject.pdf; SubversionTutorialPart2 - Erh-itproject.pdf; and SubversionTutorialPart3 - Erh-itproject.pdf.

NOTE: From this point on, there is no particular order. The steps in the remaining sections can be completed in any order.

D. WarnGen Template Customization

Start by testing the AWIPS II Baseline templates and use the localization perspective to adjust as needed. The localization perspective makes this easy to do. The site AFP for WarnGen should get the current status and training on updating these templates.

Use the checklists in this document, and those found among the resources identified in Appendix A, to help track the testing of all templates.

You are strongly encouraged to migrate your site's Mile Markers and Site Specific Dam Break information *prior to installing AWIPS II*. These files can be created on the ADAM machine and ported over to AWIPS II after the install. Instructions for completing these tasks can be found in the Additional Resources attached to this Guide (see Appendix A).

Sites with marine warning responsibilities can combine their marine zones using the documentation identified in Appendix A and the WarnGen wiki page. Combining marine zones prior to installing AWIPS II is not as critical as migrating Mile Markers and Site Specific Dam Break information. This step can be completed on the ADAM machine prior to installation and then ported over to AWIPS II, or it can be done after the install on the AWIPS II system.

WFOs that issue Zone-based WarnGen products can use the templates that use Zone instead of County-based warnings, denoted by the baseline templates with _Zones at the end of the file name.

For more information on how to customize WarnGen, see the following files, all of which are included in the resources identified in Appendix A.

1. AWIPS 2 WarnGen Documentation 1.3.1.doc (see "WarnGen" folder)

2. AWIPS 2 WarnGen Localization Tutorial.docx (see "WarnGen" folder)

3. A2 WarnGen Localization_Combine Marine Zones.docx (see "WarnGen" folder)
   - https://collaborate.nws.noaa.gov/trac/siteconfig/wiki/CombiningMarineZones

4. A2WarnGen Localization_Create Site Specific Dams.docx (see "WarnGen" folder)
   - https://collaborate.nws.noaa.gov/trac/siteconfig/wiki/SiteSpecificDams

5. Adding Mile Markers in AWIPS 2.docx (see "WarnGen" folder)

- https://collaborate.nws.noaa.gov/trac/siteconfig/wiki/AddingMileMarkers

6. AWIPS 2 Site Mig Org Aids V8.xlsm (see "Organization Aids" folder).

Note: These instructions and documents can also be found on the AWIPS II WarnGen wiki page, located at

https://collaborate.nws.noaa.gov/trac/siteconfig/wiki/WarnGen

Bookmark this link! Documentation on this wiki page will be updated regularly with additional information and resources not included in this Site Migration Guide.

E. D2D Procedure Conversion

In order to run Jim Ramer's D2D procedure conversion tool successfully, the following environmental variables need to be true:

- The fxa user must exist as it exists on an AWIPS system.

- The fxa environment must have the following variables properly set (this will be set by running **setup_fxa_user** below):

  $FXA_HOME
  $FXA_DATA
  $FXA_NATL_CONFIG_DATA
  $FXA_LOCALIZATION_ROOT
  $FXA_LOCALIZATION_SCRIPTS
  $FXA_LOCALIZATION_LOG
  $FXA_LOCAL_SITE

- AWIPS I D2D procedures should reside in their normal directory structure under **/data/fxa/userPrefs/**

- AWIPS I subset of localization files are present. See the following for details on installing on an ADAM platform.

On an ADAM platform, these should be set up properly. If they are not, please contact the NCF (301.713.9344) with the exception of $FXA_LOCAL_SITE, which is set when running **config_awips2.sh** with the *procs* option. If you are running the convertBundle executable manually, you will have to set this before executing.

To set up the fxa user on ADAM, complete the following:

On DX1 as user *root*:

```
# cd /data/fxa/INSTALL/ADAM

# scp setup_fxa_user.sh adam1:~
```

On ADAM as *user root:*

```
# cd ~root

# ./setup_fxa_user.sh
```

The actual conversion executable is delivered into a subdirectory of the SDC Tool install directory, which is named bundleConverter. On AWIPS platforms, the install directory is normally /data/fxa/sdc, so the executables are under /data/fxa/sdc/bundleConverter. On an ADAM platform, it is /var/tmp/sdc so the executables are under /var/tmp/sdc/bundleConverter.

The SDC Tool has made it easy to convert all users' procedures, or to convert procedures one user at a time. To run the D2D procedure conversion tool through the SDC Tool, issue the following command as user **root**:

```
./config_awips2.sh procs LLL
```

Where LLL is the AWIPS Site ID for the system's localization. This is important because it sets the $FXA_LOCAL_SITE variable to this ID.

a. **Running Procedure Conversion on an ADAM Platform**

To run the procedure conversion on an ADAM platform, the following must be done ahead of time:

1. Log into DX1 via *ssh* and untar the grabFromD-2D.csh script from the AWIPSII_AUTOMATION_TOOL.tgz tar bundle:

   ```
   # cd /data/fxa/INSTALL/ADAM
   # tar –xzvf AWIPSII_AUTOMATION_TOOL.tgz
   bundleConverter/grabFromD-2D.csh
   ```

2. Log into an LX workstation using *ssh* and run the script

   ```
   # ssh lx1 (you may choose any lx)

   # su - fxa

   # cd /data/fxa/INSTALL/ADAM/bundleConverter

   # ./grabFromD-2D.csh

   # exit (this exit's from the fxa shell)
   ```

3. Copy the resulting file into the /data/fxa/INSTALL/ADAM/ directory.

```
# scp /awips/fxa/convBundStage.tgz
/data/fxa/INSTALL/ADAM/
```

4. Exit from your ssh session into the LX workstation (see above) and copy the file onto the ADAM platform.

```
# exit (this exit's from the ssh session into the LX
workstation)
# scp /data/fxa/INSTALL/ADAM/convBundStage.tgz
adam1:/awips/fxa/
```

5. Log into the ADAM platform using *ssh* and untar the file.

```
# ssh adam1

# cd /awips/fxa

# tar -xzvf convBundStage.tgz

# chown -R fxa:fxalpha /awips/fxa/
```

After following the above steps, the environment should be set up to run ./config_awips2.sh procs LLL as described above.

It is recommended that you convert one user's procedures at a time. After migration, take an inventory of the procedures and have the user test each one for usefulness and accuracy compared to AWIPS I. Check the following:

- Is it a procedure that is still useful, or is it outdated and unused? If the latter, consider deleting the procedure.
- Does it display accurately? If not, contact Jim Ramer for conversion help.

> **SITE ACTION:** Create a list of all procedures, organized by user, and have the users verify that each procedure is still necessary and valid after conversion.

F. GFE SmartInits, SmartTools, Text Formatters

Keep in mind as you read this section that your knowledge of what is necessary for your backup site to perform service backup **once it has migrated to AWIPS II** is key in determining when to upload your AWIPS II configuration using Step 11 of this section. Once you are comfortable, perform Step 11 to send your GFE configuration from ADAM to the central server to allow your backup site, running AWIPS II, to perform service backup.

1. GFE SmartTools References

These references were copied from the NCLADT Wiki Site and are current with this software release. They are included among the additional resources cited in Appendix A (see "GFE Migration" folder). The first one, HowTo GFE Porting Notes, provides global comments that are useful to read before starting. Note that the Wiki link name does not always match the actual file name. In this document, we always attempt to use the actual file name.

1. HowtoGFEPorting Notes.docx
2. AppGFEafpsProxy.doc
3. GfeModuleInstaller.sh.docx, GfeModuleInstaller doc.docx
4. AppCheckForPointTools.docx, checkForPointTools doc.docx
5. gfePorter.docx
6. GFErename modules.docx
7. NCLADT Repository.docx
8. NumericToNumpy.docx
9. AppConvertEditAreas.docx.

2. Inventory SmartInits, SmartTools, and Procedures

1. Inventory your GFE SmartInits, SmartTools, and procedures.

2. Determine what to move to AWIPS II. [A spreadsheet titled "Site Mig Org Aids V8.xlsm" has been included with the additional resources to this guide; you may find it useful in completing this step.]

3. Determine which have already been converted and which have not. Look in the repository on the wiki.

Note: This information is not included in the additional resources provided with this guide because it changes frequently. Here is the repository URL: **https://collaborate.nws.noaa.gov/trac/ncladt/browser/ncladt/ladroot/gfe**

Note: The SDC ADAM Automation Tool collected the AWIPS SmartInits, SmartTools, Procedures, and Formatters and placed them in the identical directory as on an AWIPS I system when it was executed.

4. Replace the AWIPS I Tools, etc. with the converted AWIPS II Tools, etc.

3. **Tools to Run First**

The following utilities should be run before porting tools. Check to ensure you have the latest version of each tool installed before running them. Ensure that you have updated the ADAM version of the AWIPSII_AUTOMATION_TOOL, and that you have checked the SVN repository and downloaded the most current scripts for the other tools.

1. Import the AWIPS I GFE Fcst database into ADAM. This will help with testing of all converted procedures. Start on DX1 as user *root*. Run the export_gfe_fcst.sh script:

   ```
   cd /data/fxa/INSTALL/ADAM
   ./export_gfe_fcst.sh
   ```

   Once that is finished, SSH into ADAM and import the GFE Fcst database. Note that the EDEX Servers must be running for this to succeed.

   ```
   ssh root@adam1
   ./import_gfe_fcst.sh
   ```

   Run this sequence of scripts as often as necessary to test all tools properly.

2. Run Virgil's GFE Migration scripts:

   ```
   ./config_awips2.sh gfem LLL
   ```

   Where LLL is your AWIPS Site ID. This function will run the following migration scripts:

   - GFE EditArea Conversion
   - GFE Color Table Conversion
   - GFE Weather Element Group Conversion
   - GFE Config Conversion
   - GFE Samples Migration
   - GFE Timerange Migration
   - GFE Combination Files

   You may need to run this module multiple times to do all user's and site-level files.

   After running this tool, check to see that any necessary manual changes to the GFE Config files have been made. Reference the files **gfeConfigFileChanges.docx** and **GfeConfigMaps.docx** (see Appendix A, "GFE Migration" folder) for additional information.

3. **GFE_module_installer.** This is a wrapper script for ifpServerText that simplifies installing GFE modules.

4. **checkForPointTools.** All sites must convert, delete, or replace all point-based SmartTools before beginning AWIPS II migration. Run this script to list SmartTools that are affected.

5. **findAdamGfeFile.sh.** This is a simple utility script used to search for GFE files on ADAM. It searches for BASE files in the CAVE install directory, for SITE/user files in the EDEX utility directory, and for the working copy of the ladroot/gfe repository directory. This is mainly an example, so edit as needed to work on your system.

6. **GFE_rename_modules.** This is a utility used to export GFE modules such as GFE SmartTool and Procedure from AWIPS I to files suitable for use in AWIPS II. **The AWIPS 1 ifpServer** program maintains files in the /awips/GFESuite/primary/data/database with mangled names. .Tool, Procedure, etc. – names as seen in the GFE GUI – actually represent Python modules, but the GUI allows the names to contain non-alphanumeric characters. The ifpServer translates certain characters to code strings in the filename used to store the Python code on disk. For instance, an underscore in the GFE name gets translated to "_FP" in the actual filename. These mangled filenames must be converted for use with AWIPS II. The AWIPS I modules also have non-standard Python extensions such as .Procedure, .Tool, etc. GFE_rename_modules converts all modules to have the standard Python extension of .py. This program is intended to be run **on AWIPS I** as a precursor GFE migration step prior to actually porting code.

7. **gfe_porter**. This program does **simple** translation for some of the known items that have changed in AWIPS II, like Numeric to numpy. What the program converts is based on changes found when doing diffs of AWIPS I and II Python files (based on R1G1-3).

   The program does not do a full porting of the code; it only automates some of the simpler string substitutions. It also provides a detailed log file that will indicate if other possible changes are needed.

   The program should not be considered to be either complete or static. As more users port GFE modules, it is likely that more items that could have been changed by this program will be discovered. Please update the program or contact the maintainer if you discover any additional items. It will greatly help

all that need to port modules if you take notes during the porting and post your information on the Wiki page "HowToGfePortingNotes."

4. Review Differences Between AWIPS I and AWIPS II Python

CommonSmartToolChanges (see Appendix A, "GFE Migration" folder) lists common changes that focal points should reference when converting smart tools and procedures from AWIPS I to AWIPS II. This list is based on information from Tom Lefebvre.

Note: This list likely does not include all potential changes to smart tools or procedures needed at your site!

HowToGFEPortingNotes highlights python codes changes that you will need to be aware of when migrating from AWIPS I to AWIPS II.

5. Review Smart Script Methods Changes

Review these documents (see Appendix A, "GFE Migration" folder) to determine impact on your tools.

- **SmartScriptDifferences.docx.** A summary of differences.

- **SmartScriptIssues.docx.** Provides a detailed status of the methods contained in **SmartScriptDifferences.docx.**

- **SmartScriptUsageAnalysis.docx.** Provides interesting statistics on methods, and is provided "FYI."

6. Review Local Model Ingest Used in GFE

Review this document (see Appendix A, "GFE Migration" folder) to determine impact on your tools.

- **D2DinGFE.docx.** A summary of the changes to GFE parameter files required to allow D2d GriB model datasets to be utilized in the GFE perspective for either display in WxElementBrowser or for subsequent smartInit processing.

7. Convert AWIPS I SmartInits, SmartTools, Procedures Formatters

Using the inventory built earlier (see section 3.F.2, "Inventory SmartInits, SmartTools, and Procedures"), **test** the current tools one by one. Those that need additional work should be manually migrated now.

a. **Using the Localization Perspective**

From the localization perspective, you will be able to add, edit, copy, and delete various configurations.

b. **How To Install a New Tool Manually Using the Localization Perspective**

1. Launch CAVE and open the localization perspective.

2. Expand GFE, then expand SmartTools using the left click menu.

3. To edit a tool, left click to expand the tool name, then right click on BASE and copy to SITE level. Then double click the site level.

4. To create a new tool, right click on either Smarttool or procedure and select New.

5. Enter the name of the tool. (Use the same spelling as AWIPS I.)

   a. If the tool has been ported to AWIPS II, simply replace the contents in the template with the contents of the ported tool. In the new window that appears, delete the current contents.

   b. If the tool has NOT been ported, then you will have to copy and paste code from your AWIPS I tool. Then save and test.

6. To save, click the Save Icon in the upper left corner.

7. If any yellow banner messages appear, you most likely have incorrect syntax (i.e., bad indentation or spelling).

   a. Make the adjustments and then click Save again.

8. Switch to the GFE (Cave Perspective) and test your work.

Continue this process. Note that the current process is not perfect. Additional adjustments may need to be made. Please be patient.

8. SmartInits

a. **Porting Smart Inits**

- Most AWIPS I SmartInits are compatible. The biggest issue is Numpy/Numeric differences. See: NumericToNumpy.docx (see Appendix, "GFE Migration" folder).

- Before starting, check with your Regional Focal Point to see whether smartInits may be baselined for your region. If possible, begin with regionally baselined versions for easier migration to a site-specific script.

– Hard copies uncontrolled. Verify effective date before use. –

- The method of using multiple models in a single smart init has changed. See: MultiModelSmartInits.docx (see Appendix, "GFE Migration" folder).

- SITE SmartInits go into directory:
  edex/data/utility/edex_static/site/XXX/smartinit

  Where XXX is the site identifier.

Note: You will likely have to create this directory yourself.

- The class defined in your SmartInit code must be named the same as the SmartInit filename + Forecaster.

  For example: If your filename is MyGFS40.py, then the class **MUST** be named MyGFS40Forecaster in the code.

- To override a baseline SmartInit, use the same entries in localConfig as used in AWIPS I:

  ```
  del serverConfig.INITMODULES["GFS40"]

  serverConfig.INITMODULES["MyGFS40"]=["GFS40""]
  ```

  The del command tells it NOT to do the default GFS40.py SmartInit, and the new INITMODULES statement tells it to run the MyGFS40.py SmartInit when the GFS40 model arrives.

- In AWIPS I, you could have a "dummy" SmartInit that would not have any calcXXXX routines, and when it was run, it would at least create a "blank" GFE database. That does not appear to be true in AWIPS II. You must return at least one grid from a calcXXXX routine in your SmartInit for a database to be created. In addition, it appears that it must be a "real" grid (meaning returning "None" does not work).

b. Loading Smart Init Changes

- After updating SmartInit code in localConfig.py, you must restart GFE server processes via the GFE Site Activation GUI, which can be accessed from the GFE Perspective.

- Changes to smart init files are automatically loaded into EDEX the next time the smart init is called. There is no need to restart GFE processes.

9. Formatters

Most Formatters should run without problems; the method for migrating is to try them and fix those that do not work using the Python differences referenced in this kit.

10. Review and Migrate (if necessary) ifpImage Changes

Certain changes need to be made for migration of ifpImage functionality. Reference **gfeConfigFileChanges.docx** (see Appendix A, "GFE Migration" folder) for additional information.

11. Inventory, Review and Verify Migrated GFE Items

At this point, all GFE portions should be migrated. Create an itemized inventory of all migrated pieces. Run through each piece and verify proper functionality of all items. Address any problems before AWIPS II is installed on the AWIPS system.

12. Upload GFE Configuration For Service Backup

If one of your backup sites has already migrated to AWIPS II, the configuration created on ADAM must be uploaded to the central server so that the site can perform service backup. Begin by contacting the NCF to have the **export_adam_gfe.sh** script pushed to /data/fxa/INSTALL/ADAM/. Then, as root on dx1 run the script:

```
# cd /data/fxa/INSTALL/ADAM
# ./export_adam_gfe.sh
```

G. Local Apps

1. Local App Migration HowTo Guide

This section reproduces the HowToLocalAppMigration page from the NCLADT Wiki.

Local Applications Repository

The software for migrated local applications and GFE tools and formatters will be maintained in a centralized Subversion repository. The url to use with the Subversion svn command is:

- https://collaborate.nws.noaa.gov/svn/ncladt/ladroot/apps for applications
- https://collaborate.nws.noaa.gov/svn/ncladt/ladroot/gfe for GFE specific files.

These locations are effectively the replacements for the LAD and STR websites respectively. The repository layout is defined on the RepoLayout page. How to

access/use the NCLADT repository with Subversion is described at HowToSubversion. Even if you are familiar with Subversion, please review the HowToSubversion page for specifics on how local applications should be organized and maintained in the repository.

Migration Process

You need to remember four points about the local applications migration process.

First, *local applications migration should be done on the ADAM platform.*

Second, *you need to set up a local app development environment.*

1. See the Local Applications Guide for the directory structure that will need to be created and the environment variables that will need to be defined. In short, /localapps is now the home on AWIPS II for all local applications and files. Do development in /localapps/dev; programs that are run operationally go in /localapps/runtime. The directory structure and environment variable should have been set up as part of your ADAM installation. If they are not present, please contact the NCF.

2. Copy your application's files to somewhere in /localapps/dev (recommend /localapps/dev/a1/<appname> for the initial transfer from AWIPS I to AWIPS II (ADAM)).

3. Put all local applications including GFE tools/procedures, etc., into the national Subversion repository.

   - See HowToSubversion for instructions for initial contact with the repository, setting up your Subversion environment settings and initial import of application files into the repository.
   - See RepoLayout for guidance on where to put the files in the repository.

Third, *know your app.* What parts of the infrastructure does it use? Look for Wiki pages that will help.

   ✓ textdb - a drop-in replacement for the awips1 version of textdb.
   ✓ Triggers in awips1 must be re-implemented as subscriptions in awips2.
   ✓ gfe
   ✓ how to query EDEX
      ▪ uEngine scripting
         – grib data access routines
      ▪ thrift
         – intro

✓ LDAD - Information on migrating LDAD applications
✓ Hydro - Information on migrating applications using the Hydro database
✓ Crons - Information on where to run cron jobs.

For topics that are not listed, see the Finding stuff section of the HowToWiki to see if a page exists within the Wiki. If not, ask on **awips2dev** for help. *Add what you learn to the Wiki so others can benefit*.

Fourth, *after completing migration for an application, perform the following steps:*

1. Commit application changes to NCLADT repository.

2. Tag the application in the repository (Instructions TBD) .

3. Update tickets for the application on the AWIPS Migration OT&E site.

4. If the application is used at other sites, consider creating a Wiki page for the application. Start the page name with "App" (see HowToWiki).

Rehosted Functions

The following AWIPS I functions have been Rehosted, which means that no significant code changes were made when moving to AWIPS II:

✓ Message handling. This includes distributeProduct and handleOUP.pl.
✓ LDAD external - no changes on the external LDAD side (ls2/3).

Additional Information
✓ "Local Apps working session - migration.docx"
✓ "Local Applications Guide.docx"

2. Triggers

1. Take an inventory of all triggers in your siteTrigger.template file, which is in **/data/fxa/siteConfig/textApps** on AWIPS I.

2. Make an AWIPS II-specific copy of the siteTrigger.template so your changes are not overwritten in a future **collect_files.sh** run:

   cp –a /data/fxa/siteConfig/textApps/siteTrigger.template

   /data/fxa/siteConfig/textApps/AWIPS2-siteTrigger.template

3. Move the appropriate scripts to the correct place in /localapps and change the entries in the AWIPS2-**siteTrigger.template** on ADAM to reflect this change.

– Hard copies uncontrolled. Verify effective date before use. –

4. Identify and manually migrate any portions of the script so they will work with the AWIPS II environment. Some things to consider:

   ✓ Product output is now put into **/awips2/edex/data/fxa/trigger instead of /data/fxa/trigger**.

   ✓ Scripts are kicked off by an EDEX process, which runs as user **awips** instead of user **fxa**.

   ✓ Scripts are run from either DX3 or DX4.

   ✓ Environment variables for PGUSER and PGHOST will still be set by the environment.

4. Reload the triggers into the database with the AWIPS II SDC Automation Tool with the following command:

   ```
   scp /data/fxa/siteConfig/textApps/AWIPS2-
   siteTrigger.template
   /data/fxa/siteConfig/textApps/siteTrigger.template
   ./config_awips2.sh -f triggers LLL
   ```

5. Ensure the scripts are launching and performing their correct operation.

3. Crons

1. Identify SITE level crons on DX1/DX2/DX3/DX4/PX1/PX2.

2. Adjust crons for AWIPS II.

3. Make adjustments for directory changes (e.g., /localapps/runtime).

4. Plan to move off DX3/DX4, as heartbeat no longer runs so SITEdx3cron and SITEdx4cron will no longer be loaded. Move these crons to PX1/PX2. This will be done following the AWIPS II software installation.

4. rsyncGridsToCWF

NWS offices use rsyncGridsToCWF to export the GFE database to a netcdf file, which is then uploaded to the regional web farm via LDAD for use by NWS point-and-click websites.

This application is now baselined in AWIPS II under /awips2/GFESuite/bin.

H. Shapefiles

Shapefiles should have been set up before running the initial migration using the AWIPS II Site Data Configuration automation tool (**config_awips2.sh**). If they need to be updated, staging should take place as described in this chapter (see 3.A, step 7).

Once staging is complete, the following function of the **config_awips2.sh** script can be used:

```
./config_awips2.sh shp LLL
```

If there are errors reported while importing FFMP shapefiles or local shapefiles, or if FFMP fails to start and there are FFMP errors in the ingestDat logs, the shapefiles should be tested for errors.

1. Shapefile Testing

The OpenJump program can be found at http://www.openjump.org/.

After installation, start the program.

Go to **File>>Open File** and navigate to where the shapefile exists and add it.

Make sure the layer/shapefile to be tested is highlighted and then go to **Tools>>QA>>Validate Selected Layers.** Using the values below, fill in the form and click OK

- Check basic Topology.
- Disallow repeated consecutive points.
- Check Polygon orientation.
- Check minimum segment length (Use 0.000001).
- Check minimum polygon area (Use 0.00000001).
- Check that LineStrings are simple.

After processing, if there are errors, a number of orange circles will appear on the map. Press the blinking output window to see what kinds of errors were found or to confirm no errors found.

2. Fixing Shapefiles

Any GIS capable of editing shapefiles can be used. However, every WFO has access to the commercial GIS called ArcGIS via a shared license pool; therefore, it recommended that this software be used to edit the shapefiles if required. A small number of errors can be edited by hand. Nodes/vertices can be deleted for short line segments. Small polygons can be deleted or merged with a neighboring large

polygon. Single-point intersections can be changed to two polygons. Large numbers of node type errors are best fixed using advanced ArcInfo techniques.

I. LDAD

Here are three tests you can perform for LDAD with the ADAM platform.

IMPORTANT NOTE: When setting up Local Model Ingest in AWIPS II, be sure <u>not</u> to use an underscore ("_") in your model name or in any parameter names if you are setting up parameter info for that model for GFE.

1. Identify All Data Incoming Into LDAD

Note the following:

- Most point data works the same as AWIPS I.

- Grid data may need to be defined in AWIPS II in order to ingest properly.

- Grid preprocessors will need to be changed. They should put raw grib files into **/awips2/edex/data/manual/.** You might also consider prepending all grid files with a string so only one change will need to be made to the grib distribution xml in AWIPS II.

2. Test a Sample of Grid Data Ingest on ADAM

- Copy raw grid file from LDAD to DX1, and then over to ADAM.

- Ensure that grid distribution xml has been copied to the site-level directory and that a change has been made for this file name.

- Suggestion: Mimic what will be used by the AWIPS II preprocessor after migration.

- Copy the file into **/awips2/edex/data/manual** with a file name that matches the entry in the grid distribution xml.

- Follow the steps in "AddLocalGrid.pdf" (see Appendix A, "Local Applications" folder) to see if the product has been identified and create configurations as necessary. Repeat for all grids.

3. Test Mesonet LDAD Ingest on ADAM

The **LDADinfo.txt** file defines mesonet data that is transferred from LDAD into AWIPS (see the *AWIPS System Manager's Manual*, Chapter 9, for more information).

To perform the steps in this section you should be familiar with your site's LDAD preprocessing scripts and aware of where the data files will reside once they are copied over to the AWIPS side.

The AWIPS II rehosted process routerStoreEDEX sends mesonet data directly to EDEX for processing. Because this rehosted process cannot be run on ADAM, data files need to be manually copied to ADAM for testing.

To run this process in standalone mode, you must be user LDAD on the px2f server (the host running the px2apps package), and ensure that there is an ldadmesonet.xml distribution xml created in the **/awips2/edex/data/utility/edex_static/site/LLL/distribution** directory on ADAM. You should have an entry that allows file names starting with "ldad" inside. The entry should look like this:

```
<regex>^ldad</regex>
```

Now create an XML file to copy over to ADAM by running the routerStoreEDEX in standalone mode, and redirect the output to a new file

```
su – ldad
source /etc/profile.d/awips2Notification.csh*
routerStoreEDEX -s -p /data/fxa/LDAD/decoded/<FILE_NAME>
> /tmp/ldadIngestFile
```

Copy this file to the manual endpoint in ADAM – this is done as user root

```
scp /tmp/ldadIngestFile adam1:/awips2/edex/data/manual
```

The mesonet data should now be ready for ingest into the AWIPS II on ADAM.

If changes to LDADinfo.txt for AWIPS II are necessary, create an AWIPS II version of the file and make your modifications in the AWIPS2-LDADinfo.txt file:

```
cp –a /data/fxa/LDAD/data/LDADinfo.txt
/data/fxa/LDAD/data/AWIPS2-LDADinfo.txt
```

On install day, the LDADinfo.txt file will be preserved as AWIPS1-LDADinfo.txt, and a link from LDADinfo.txt to AWIPS2-LDADinfo.txt will be created. This will preserve your AWIPS I configuration, and facilitate swapping modes.

NOTE: Until further notice, Raytheon will perform the steps in Section 4, "System Installation."

4. SYSTEM INSTALLATION[5]

This section details the procedures for installation of the AWIPS II software build on a pre-existing AWIPS I platform. These procedures are tailored to an AWIPS I system that has an OB9.2 or upstream ("newer") software build. The assumption is that the system has a "baseline" hardware and software configuration; the procedures do not account for non-baseline hardware or software. If the system being installed has non-baseline hardware or software (i.e., anything considered outside the national AWIPS baseline, including anything specific to a region), you must ensure that steps are taken to preserve the functionality of this hardware or software prior to executing the installation. Otherwise, these procedures could affect changes that cause non-baseline hardware or software to become non-functional or to malfunction.

This section details the system-level configuration and changes required to install the AWIPS II software. These steps will need to be **completed and verified before any AWIPS II software is installed on the system.** You will have a Raytheon point of contact who will be responsible for the installation of the AWIPS II software, and for coordination of the following system-level changes. The information contained in these steps may be useful in building your knowledge of what changes occur for AWIPS II architecturally. In addition, the Raytheon point of contact who is responsible for collaborating with you on most of the system-level changes will need on-site assistance to complete the setup, so a review of the steps is suggested prior to coordination.

Note: Verification tables are provided to ensure key points are not missed.

A. Architectural Overview

1. Preliminary Architecture

The preliminary architecture is described in Table 4-1.

---

[5] Until further notice, Raytheon will perform the steps in Section 4, "System Installation."

Table 4-1. AWIPS II Preliminary Architecture

Server	DX1	DX2	DX3	DX4
AWIPS II Software Component	edex_postgres edex_rcm	httpd-pypies	edex_camel	edex_camel
Rehost	NTP MHS a2Ingest.dx1 dx1f			
Local Volumes	/awips2	/awips2	/awips2	/awips2
Mounts	das1:/aiidb das1:/aiihdf5 nas1:/aiidata/utility nas1:/GFESuite2 nas1:/localapps nas1:/data_store	nas1:/GFESuite2 nas1:/localapps nas1:/data_store	nas1:/aiidata nas1:/GFESuite2 nas1:/localapps nas1:/data_store	nas1:/aiidata nas1:/GFESuite2 nas1:/localapps nas1:/data_store
Server	PX1	PX2	CPSBN1	CPSBN2
AWIPS II Software Component			IPVS LDM qpidd	future: edex_camel (registry)
Rehost	a2Ingest.px1 xyplex SBN EbNo CUPS Async MUX px1f	a2Ingest.px2 startLDAD.csh mcMON px2f		
Local Volumes	/awips2	/awips2	/awips2	/awips2
Mounts	nas1:/aiidata nas1:/verify nas1:/GFESuite2 nas1:/localapps nas1:/data_store	nas1:/aiidata nas1:/verify nas1:/GFESuite2 nas1:/localapps dx2f:/data_store	nas1:/qpid	

2. LX Workstations and XT Workstations

   /awips2

   nas1:/data_store mount

   nas1:/aiidata/share

   nas1:/verify

   nas1:/localapps

3. High-Availability Packages

The **High-Availability** packages contain the following functions in AWIPS II and are located in *etc/ha.d/resource.d* along with the AWIPS I packages:

a2dx1apps:

- NTPD
- FXA Ingest
- MHS (rehosted)
- notificationServer (rehosted)
- postgresSQL database engine (postmaster)
- RCM (Radar server)

a2dx2apps:

- pypies

a2px1apps:

- CUPS
- Xyplex Interface (csportd)
- FXA Ingest
- Async MUX ports
- httpd (web server)
- SBN Monitor

a2px2apps:

- FXA Ingest
- LDAD Ingest

a2cp1apps:

- qpidd
- IPVS (pulse)
- LDM
- CPSBN Comms (dvb reset, retrains rehost)

a2cp1apps:

- Future: edex_camel (registry)

4. FXA Ingest Packages

The **FXA Ingest** packages, which are controlled manually via [start|stop]A2ingest.${node} or via the HA cluster packages above, are located in */awips/fxa/bin* and contain the following functions:

[start|stop]A2Ingest.dx1:

- MhsServer
- MhsRequestServer
- NwwsProduct

[start|stop]A2Ingest.px1:

- asyncScheduler
- hmMonitorServer
- NwwsSchedule

[start|stop]A2Ingest.px2:

- LDAD Servers
- FSIprocessorEDEX

B. Pre-Installation Checkout

Prior to installation of the AWIPS II software, it is recommended that a complete system checkout be performed. Ensure that LAN connections are correct, and if necessary, reboot the switches. Coordinate with the Network Control Facility (NCF) if any reboot is being performed or desired.

It is also recommended (if not already done) that the LX and XT workstations be moved from the 100Mb switch to the GB switch for performance reasons. Communication between the CAVE client and EDEX occurs via HTTP, and maps within CAVE are dynamically queried from a GIS database; if running on the 100Mb LAN such functionality as map initialization and rendering will be slow.

Ensure that the NAS connections are correct and properly seated before continuing.

C. System Enhancements and Changes

Perform the following **prior to stopping any AWIPS I software or installing any AWIPS II software.** During the system enhancements and changes, the AWIPS I system will remain operational. It is recommended that these steps be performed a few days prior to the installation of the AWIPS II software.

Note: All devices should be booted and available on the network before proceeding. If you have a question as to whether a device is available, or should be available, coordinate with your Raytheon point of contact.

1. Stage the Software

   Insert the AWIPS II Software DVD1 into dx1, and then do the following to mount and copy the data in:

   mount /dev/cdrom /media/cdrecorder
   cd /media/cdrecorder
   ./deployAWIPS2.sh
   cd; eject

   Remove DVD1 and insert the AWIPS II Software DVD2 into dx1, and then do the following to mount and copy the data in:

   mount /dev/cdrom /media/cdrecorder
   cd /media/cdrecorder
   ./deployAWIPS2-2.sh
   cd; eject

2. System Enhancement/Changes

   Create New Users

   Run from dx1 as user *root*

   ```
   cd /data/fxa/INSTALL/a2install
   ./prepare_awips2.sh users
   ```

   Create EDEX HOME

   Create a logical volume on all DX and PX server within /dev/vg00 volume group. The AWIPSII software will be installed locally on each server within the file system created from a new logical volume.

   cd /data/fxa/INSTALL/a2install
   ./prepare_awips2.sh edex_home

   See **Table 4-2**, Verification of EDEX_HOME.

Table 4-2. Verification of EDEX_HOME

Task	Device:Filepath	Completed?
Create new logical volume	DX1	
	DX2	
	DX3	
	DX4	
	PX1	
	PX2	
	CPSBN1	
	CPSBN2	
Make the linux filesystem on the new logical volume	DX1	
	DX2	
	DX3	
	DX4	
	PX1	
	PX2	
	CPSBN1	
	CPSBN2	
Mount the filesystem	DX1	
	DX2	
	DX3	
	DX4	
	PX1	
	PX2	
	CPSBN1	
	CPSBN2	
Persist the mounting across reboots	DX1:/etc/fstab	
	DX2:/etc/fstab	
	DX3:/etc/fstab	
	DX4:/etc/fstab	
	PX1:/etc/fstab	
	PX2:/etc/fstab	
	CPSBN1:/etc/fstab	
	CPSBN2:/etc/fstab	

– Hard copies uncontrolled. Verify effective date before use. –

Create LDM, Database, and HDF5 Volumes

All of the following steps are executed as user **root** unless otherwise noted.

1. Log into **DX1** and then connect to the DAS:

   # telnet ats1 7041

2. Press **ENTER** until you see **dx2-das>** prompt. Shrink vol0:

   dx2-das> vol size vol0 -120G

   Press `ctrl-]` to exit the telnet session, and then enter "`quit`" to get back to the dx1 prompt

3. Log into **DX1** and then connect to the DAS:

   # telnet ats1 7040

4. Press **ENTER** until you see **dx1-das>** prompt. Create a volume for the database, ldm and hdf5:

   dx1-das> vol size vol0 -120G
   dx1-das> vol create /vol/awipsiidb aggr0 250g
   dx1-das> vol options /vol/awipsiidb nosnap on
   dx1-das> vol options /vol/awipsiidb nosnapdir on
   dx1-das> vol options /vol/awipsiidb fractional_reserve 0
   dx1-das> snap reserve awipsiidb 0
   dx1-das> vol create /vol/awipsiihdf5 aggr0 600g
   dx1-das> vol options /vol/awipsiihdf5 nosnap on
   dx1-das> vol options /vol/awipsiihdf5 nosnapdir on
   dx1-das> vol options /vol/awipsiihdf5 fractional_reserve 0
   dx1-das> snap reserve awipsiihdf5 0
   dx1-das> snap autodelete awipsiihdf5 target_free_space 5
   dx1-das> snap autodelete awipsiihdf5 defer_delete none
   dx1-das> snap autodelete awipsiihdf5 state on
   dx1-das> snap sched –V awipsiihdf5 0 0 4@1,5,9,13,17,22
   dx1-das> vol options awipsiihdf5 nosnap off
   dx1-das> vol create /vol/awipsiildm aggr0 170g
   dx1-das> vol options /vol/awipsiildm nosnap on
   dx1-das> vol options /vol/awipsiildm nosnapdir on
   dx1-das> vol options /vol/awipsiildm fractional_reserve 0
   dx1-das> snap reserve awipsiildm 0

5. Create the LUNs for direct attachment to DX1/2:

   dx1-das> lun setup

   Press **ENTER** to create a LUN.

   Press **ENTER** to select the linux protocol.

Enter **LUN path: /vol/awipsiidb/base-lun**

Press **ENTER** to reserve space in the LUN.

Enter **LUN size: 249g**

Enter comment string: **awipsii database flat files**

Press **ENTER** for the default initiator group.

Press **ENTER** for default LUN id of 1.

Press **ENTER** to accept the configuration.

Press **y** to create a second LUN.

Press **ENTER** to select the linux protocol.

Enter **LUN path: /vol/awipsiihdf5/base-lun**

Press **ENTER** to reserve space in the LUN.

Enter **LUN size: 399g**

Enter comment string: **awipsii hdf5 files**

Press **ENTER** for the default initiator group.

Press **ENTER** for default LUN id of 2.

Press **ENTER** to accept the configuration.

Press **y** to create another LUN.

Press **ENTER** to select the linux protocol.

Enter **LUN path: /vol/awipsiildm/base-lun**

Press **ENTER** to reserve space in the LUN.

Enter **LUN size: 169g**

Enter comment string: **awipsii ldm files**

Press **ENTER** for the default initiator group.

Press **ENTER** for default LUN id of 3.

Press **ENTER** to accept the configuration.

Press **ENTER** to end creation of LUNs.

Press **ctrl-]** to exit the telnet session, and then enter "**quit**" to get back to the dx1 prompt.

On DX1 as root:

```
cd /data/fxa/INSTALL/a2install
./prepare_awips2.sh das_volumes
```

See **Table 4-3,** Verification of the DAS Mounts.

Table 4-3. Verification of the DAS Mounts

Task	Device:Filepath	Completed?
Create DAS awipsiildm, awipsiidb and awipsiihdf5 volumes	DX1-DAS	
Create DAS LUNs for direct attachment	DX1-DAS	
Create gv_aiildm, vg_aiidb, vg_aiihdf5 and logical volumes awipsiildm, awipsiidb and awipsiihdf5	DX1	
Create ext2 filesystem on awipsiidb and xt3 filesystems on awipsiildm and awipsiihdf5	DX1	
Create mount points and mount awipsiidb and awipsiihdf5	DX1	
Create /awips2/data and /awips2/edex/data/hdf5 mount points;	DX2	
NFSD started and persisted across reboots with chkconfig	DX2	

Create New NAS Shares for AWIPSII

Note: If there is a localapps partition on the NAS from the old SCAT project, it **will be removed** at this point. Coordinate with the Raytheon installation point of contact if you wish to save anything on this partition.

As root on DX1:

```
cd /data/fxa/INSTALL/a2install
```

```
./prepare_awips2.sh nas_shares
```   (takes about 20 minutes)

Note: To verify the changes have taken effect login to the NAS and at the prompt (nas1-xxx>) enter menu and hit [**ENTER**]. This will take you into the TUI of the ST5320. Enter option 'L. Volume Access' by entering the character letter 'L'. The [**SPACEBAR**] is used to scroll the screen, search for the newly created /aiidata, /verify, /GFESuite2 and /qpid volumes and ensure the General Access is listed as Read/Write. It is normal for the *.chkpnt volume to have No Access. If the newly created volumes are not Read/Write, contact your install support personnel.

Mount and Persist the Newly Created LUNs on the AWIPSII Servers

As root on DX1:

```
cd /data/fxa/INSTALL/a2install
```

```
./prepare_awips2.sh mount_luns
```

See **Table 4-4,** Verification of NAS Shares.

Table 4-4. Verification of NAS Shares

Task	Device:Filepath	Completed?
Create new volumes	nas1:/GFESuite2 nas1:/aiidata nas1:/verify nas1:/qpid nas1:dataSTORE nas1:/localapps	
Edit nas1 approve list	DX1:/mnt/nas	
Persist the mounting across reboots	DX_SERVERS:/etc/fstab	
	PX1:/etc/fstab	
	PX2:/etc/fstab	
	LX_WORKSTATIONS:	
	/etc/fstab	
	XT_WORKSTATIONS:	
	/etc/fstab	
Mount the filesystems	DX_SERVERS	
	PX1	
	PX2	
	LX_WORKSTATIONS	
	XT_WORKSTATIONS	

D. IPVS Installation and Configuration

IPVS will be used to manage the edex (DX3/4) clusters. To install and configure:

Verify that /data/fxa is mounted on the CPSBN servers. On **CPSBN1** and **CPSBN2** as user root:

```
# mount -l | grep /data/fxa
```

As user root on **DX1**:

```
# cd /data/fxa/INSTALL/<A2_REL_ID>/REHOST_CODE/ipvs
# ./install_ipvs.sh
```

E. LDM Installation

Note: Legacy AWIPS SBN procs and the AWIPSII-LDM <u>cannot</u> run concurrently with a feed from a single DVB-S receiver. Only a single process can read from the receiver threads at a single time.

Note: Most of the LDM installation procedures are now included in the awips2-ldm rpm that will be installed when the AWIPS II software is installed.

a. Upstream LDM Host Installation

1. Edit required system configuration files. Complete the following steps on both cpsbn1 and cpsbn2, one at a time:

2. Add the following lines:

```
local0.debug       -/data/ldm/logs/ldmd.log
local3.debug       -/data/ldm/logs/nwstg.log
local4.debug       -/data/ldm/logs/goes.log
local5.debug       -/data/ldm/logs/nwstg2.log
local6.debug       -/data/ldm/logs/oconus.log
local7.debug       -/data/ldm/logs/polarsat.log
```

3. Also add the following after "cron.none" in the line that ends with "/var/log/messages":

```
;local0,local3,local4,local5,local6,local7.none
```

Other Changes

Ramdisk creation for ldm pq (all one command):

```
# sed -i -e "/vmlinuz-2.6.18-
194.17.1.el5PAE/s/rhgb/ramdisk_size=1500000 rhgb/1"
/boot/grub/grub.conf
```

Verify the sed command worked (rerun if output it not correct):

```
# grep rhgb /boot/grub/grub.conf
```

b. CP Disk Settings

Fail over channels to cpsbn1 and setup init scripts.

On **CPSBN1** as user root:

 # config_dvb -a -c GOES -c NOAAPORT_OPT –c NMC3

Fix the CP BIOS settings and perform disk optimization.

First, prepare for disk optimization. You will need a Red Hat 5.5 bootable CD. If you do not have one, create one using the archive server (AX):

1. Put a blank CD-R (not DVD) into the Archive Server (AX) CD-R Drive.

2. Secure shell login to the ax box as user root from any machine:

    Enter: **ssh root@ax**

3. Ensure the /data/fxa mount is available, and mount it if it isn't.

    Enter: **mount -l | grep dataFXA**

    If nothing returns, then mount the volume:

    Enter: **mount nas1:dataFXA /data/fxa**

4. Burn the CD. The following command is all on one line:

    Enter: **cdrecord
    /data/fxa/install_root/redHatFiles/32bit/images/boot.i
    so**

5. Eject and remove the CD. Label it "RHEL5u5 32bit Rescue CD."

6. Enter: `eject /dev/cdrom`

Next, create kick-start files:

1. Change into the /data/fxa/CP_PARTITION directory and run the
    create_cp_ks.sh script:

    Enter: **ssh root@dx1**

    Enter: **cd /data/fxa/CP_PARTITION**

    Enter: **./create_cp_ks.sh cpsbn1**

    Note: There should be no output to this script.

    Enter: **./create_cp_ks.sh cpsbn2**

    Note: There should be no output to this script.

2. Insert a USB thumb drive into the dx1 server, and copy the files over to the
    usb:

    After inserting the drive, issue the following:

    Enter: **dmesg|tail -20**

    You will see output similar to the following:

    ```
    [24151.364387] sd 8:0:0:0: Attached scsi generic sg2
      type 0
    [24151.368796] sd 8:0:0:0: [sdb] 7831552 512-byte
      logical blocks: (4.00 GB/3.73 GiB)
    [24151.369664] sd 8:0:0:0: [sdb] Write Protect is off
    ```

```
[24151.369668] sd 8:0:0:0: [sdb] Mode Sense: 43 00 00
    00
[24151.369671] sd 8:0:0:0: [sdb] Assuming drive cache:
  write through
[24151.373051] sd 8:0:0:0: [sdb] Assuming drive cache:
  write through
[24151.374182] sdb: sdb1
[24151.375654] sd 8:0:0:0: [sdb] Assuming drive cache:
  write through
[24151.375657] sd 8:0:0:0: [sdb] Attached SCSI
  removable disk
```

Replace DEVICENAME with the output from the above highlighted device name:

Enter: **mkdir -p /media/usbdisk**

Enter: **mount /dev/DEVICENAME /media/usbdisk**

3. Copy the kickstart files to the USB and eject the USB

Enter: **cp ks.cfg.cpsbn1 /media/usbdisk**

Enter: **cp ks.cfg.cpsbn2 /media/usbdisk**

Enter: **eject /media/usbdisk**

Note: If your CPs are remote, you should perform the rest of this section where the CPs are physically located. Don't forget to bring the flash drives with the ks files.

Back up the cpsbn2 device.

1. You will need a monitor and a keyboard hooked up to the front of the CPSBN2 device to proceed. Hook a monitor to the VGA port on the front, and a keyboard to the USB port in the front.

2. Ensure that /data/fxa is mounted, and mount if it is not:

Enter: mount -l | grep dataFXA

If nothing returns, then mount the volume:

Enter: mount nas1:dataFXA /data/fxa

3. Copy the backup script to / and run it:

Enter: cp /data/fxa/CP_PARTITION/BackupCP.sh /

Enter: cd /data/fxa/TEMP/cpsbn

Enter: mkdir archive

Enter: mv cpsbn2* archive/

Enter: cd /

Enter: `./BackupCP.sh`

Reboot cpsbn2 and change BIOSsettings.

1. Put the RHEL5u5 32bit Rescue CD into the CD drive.

2. Log into cpsbn2 as user root and issue the reboot command:

   Enter: `reboot -f`

3. The HP ProLiant screen will show, and the buttons in the right corner will light up. Then text will begin scrolling. Get ready to hit F9 at this point, and then press **F9** when the option appears in the bottom left corner. You should see "F9 Pressed" if it took the keypress.

4. Once booted into BIOS, select Power Management Options --> HP Power Regulator. At this point it will give a note; just press the ENTER key to move forward.

   Select HP Static High Performance Mode and hit **ENTER.**
   Hit **Esc** twice and then **F10** to exit.

5. Once out of BIOS, put the USB drive into the server. You should be presented with a RedHat install prompt. [**Note:** Depending on how the flash drive was formatted, you may need to type "sda" instead of "sda1" when you type the line below.]

   At the prompt type:

   `linux ks=hd:sda1:/ks.cfg.cpsbn2`

6. The install takes about 15 minutes. It will run a restore on the CP from your backup automatically. When it reboots, you should be able to log in and perform these cleanup steps:

   Enter: **`rm -rf /data/*`**

   Enter: **`mkdir -p /data/ldm/logs`**

   Enter: **`mkdir -p /data/ldm/data`**

   Enter: **`chown -R ldm:fxalpha /data/ldm`**

   Enter: **`service syslog restart`**

   Enter: **`config_dvb -a -c NMC -c NMC2`**

Back up the cpsbn1 device.

1. You will need a monitor and a keyboard hooked up to the front of the CPSBN1 device to proceed. Hook a monitor to the VGA port on the front, and a keyboard to the USB port in the front.

– Hard copies uncontrolled. Verify effective date before use. –

2. Ensure that /data/fxa is mounted, and mount if it is not:

> Enter: `mount -l | grep dataFXA`

If nothing returns, then mount the volume:

> Enter: `mount nas1:dataFXA /data/fxa`

3. Copy the backup script to / and run it:

> Enter: `cp /data/fxa/CP_PARTITION/BackupCP.sh /`

> Enter: `cd /data/fxa/TEMP/cpsbn`

> Enter: `mkdir archive`

> Enter: `mv cpsbn1* archive/`

> Enter: `cd /`

> Enter: `./BackupCP.sh`

Reboot cpsbn1 and change BIOSsettings.

1. Put the RHEL5u5 32bit Rescue CD into the CD drive.

2. Log into cpsbn1 as user root and issue the reboot command:

> Enter: `reboot -f`

The HP ProLiant screen will show, and the buttons in the right corner will light up. Then text will begin scrolling. Get ready to hit F9 at this point, and press **F9** when the option appears in the bottom left corner. You should see "F9 Pressed" if it took the keypress.

3. Once booted into BIOS, select Power Management Options --> HP Power Regulator. At this point it will give a note; just press the ENTER key to move forward.

> Select HP Static High Performance Mode and hit ENTER

> Hit **Esc** twice and then **F10** to exit

4. Once out of BIOS, put the USB drive into the server. You should be presented with a RedHat install prompt. [**Note:** Depending on how the flash drive was formatted, you may need to type "sda" instead of "sda1" when you type the line below.]

> At the prompt type:

> `linux ks=hd:sda1:/ks.cfg.cpsbn1`

5. The install takes about 15 minutes. It will run a restore on the CP from your backup automatically. When it reboots, you should be able to log in and perform these cleanup steps:

> Enter: `rm -rf /data/*`
>
> Enter: `mkdir -p /data/ldm/logs`
>
> Enter: `mkdir -p /data/ldm/data`
>
> Enter: `chown -R ldm:fxalpha /data/ldm`
>
> Enter: `service syslog restart`
>
> Enter: `config_dvb –r –c GOES –c NOAAPORT_OPT –c NMC3`
>
> Enter: `ssh cpsbn2 "config_dvb –r –c NMC –c NMC2"`

3. Downstream LDM Client Pre-Installation

Complete the following pre-installation steps. Once these steps are completed, the AWIPS II LDM client software can be installed on the system. [**Note:** This will be repeated for DX2.]

Note: Uninstall any pre-existing AWIPS LDM software on DX1, as user root:

 # rpm –e ldm-6.6.5-4.AWIPS.OB9211
 # rpm -qa | grep -i ldm

If anything is returned other than a command line prompt, remove those packages (rpm -e package_name) from the system before installing the AWIPSII LDM client software.

---

1. Begin by setting up logging for LDM.

    ```
    # mkdir /data/logs/ldm
    # chown ldm:fxalpha /data/logs/ldm
    #  vim /etc/syslog.conf
    ```

2. Add the following line:

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

3. Also add the following after "cron.none" in the line that ends with "/var/log/messages":

    ```
    ;local0.none
    # service syslog restart
    ```

Increase Size of Logging Partition. The /data/logs partition on both DX1 and DX2 needs to be resized from its current 2GB to10GB in order to accommodate the needs of the downstream LDM. No processes need to be stopped.

 # lvs | grep lvol09

Note: You will see the current size as 2GB. If this says 10GB already, skip the next three steps.

> **# lvextend -L10G /dev/vg00/lvol09**
> **# resize2fs -fp /dev/vg00/lvol09**
> **# lvs | grep lvol09**

Note: This should show the new size as 10GB.

When you have finished, **REPEAT the "Downstream Client Installation" section for dx2.**

F. AWIPS II Software Installation

1. Pre-Install

At this time sites should download any updated baseline shapefiles to /data/fxa/nationalData to be picked up later by the config_awips2.sh script. Sites should look at the AWIPS1 SMM Chapter 15 and section 5.C.4.b in this document for details. If sites do not want to use these updated shapefiles in AWIPS1, then do not run mainScript as described in the AWIPS I SMM. The AWIPS2 config_awips2.sh tool can use them as they are downloaded from noaa1, but will rename them as appropriate to the general AWIPS names. It is recommended that the files be updated in AWIPS I as well, for consistency purposes in the event of a roll-back.

Run a backup script on ADAM to ensure all local configurations are backed up:

```
# ssh root@adam1

# ./backup_localapps.sh

# exit
```

Run the following steps as user root on DX1.

Download the latest SDC tool and extract it:

```
# scp <username>@as3-
ancf:~KevinJ/AWIPSII_AUTOMATION_TOOL.tgz
/data/fxa/INSTALL/ADAM

# mkdir /data/fxa/sdc

# tar –xvf /data/fxa/INSTALL/ADAM/
AWIPSII_AUTOMATION_TOOL.tgz –C /data/fxa/sdc
```

Make sure you have a current backup of the AWIPS I databases by running as root on DX1:

```
# /awips/ops/bin/backup_pgdb -a
```

Make sure you have a current backup of the AWIPS 1 GFE FCST and ISC DBs:

```
# ssh dx4f
# su - ifps
# cd /awips/GFESuite/primary/bin
# ./ifpnetCDF -t -k -o /data/fxa/TEMP/LLL_Fcst_BU -d
LLL_GRID__Fcst_00000000_0000
# ./ifpnetCDF -t -k -o /data/fxa/TEMP/LLL_ISC_BU -d
LLL_GRID__ISC_00000000_0000
```

Where LLL is your AWIPS Site ID

WARNING: From this point on, there will be an operational impact to the site. The site should be in service backup before continuing.

WARNING: VerifySshKeys.sh will be run. Be prepared to restore local modifications after the install.

Mount /data/fxa on cpsbn1 and cpsbn2:

```
# ssh cpsbn1 "mkdir -p /data/fxa; mount nas1:/dataFXA
/data/fxa"
# ssh cpsbn2 "mkdir -p /data/fxa; mount nas1:/dataFXA
/data/fxa"
```

Set up heartbeat on the CPSBN servers.

```
# cd /data/fxa/INSTALL/HA4CP
# ./config_ha_cpsbn.sh cpsbn1
# ./config_ha_cpsbn.sh cpsbn2
```

Run VerifySshKeys.sh to set up awips and cp1f keys.

```
ssh px1
cd /data/fxa/INSTALL/<A2_REL_ID>/REHOST_CODE/desktop
./VerifySSHKeys.sh
```

exit (exits back to DX1)

```
cd /home/awipsadm/ssh
```

Note: The **installCRSssh.sh** script will prompt you for the root password of the CRS system, so have the information on hand.

```
./installCRSssh.sh
```

Shut down all the AWIPS I packages:

```
# for host in $DX_SERVERS $PX_SERVERS $RP_SERVERS
 > do
 > machine=`echo $host | cut -d- -f1`
 > ssh -q $host "hb_halt ${machine}apps"
 > done
```

Stop LDM on LDAD:

```
# ssh ls1
# su – ldm
# ldmadmin stop
# exit
# exit
```

Bring up the <u>dx1f</u> floater IP address:

```
# ypcat hosts | grep dx1f
```

Plug in the following IP address:

```
# ifconfig bond0:0 165.92.XX.XX up
```

Bring up the dx2f floater IP address:

```
# ypcat hosts | grep dx2f
```

Plug in the following IP address:

```
# ssh dx2 "ifconfig bond0:0 165.92.XX.XX up"
```

2. Install

   a. **Verification of AWIPS II Repository**

   The AWIPS II repository was set up in a previous step. Verify that the repository was set up correctly. Issue the following command on DX1 as user *root*.

   ```
   # yum grouplist | grep 'AWIPS II'
   ```

   The following should be present:

   AWIPS II Backup Database Server
   AWIPS II Database Server
   AWIPS II LDM Server
   AWIPS II Message Broker Server
   AWIPS II Processing Server

> AWIPS II Rehost Server
> AWIPS II Standalone
> AWIPS II Visualize

If any are not present, the rpm repository needs to be verified.

b. Install Visualization Applications (CAVE/GFEClient/AlertViz/etc)

Log into DX1 as user root through a terminal server connection. Complete the following step to install the necessary software on all the workstations.

1. Prepare the workstations for 64bit.

```
#   for host in $LX_WORKSTATIONS $XT_WORKSTATIONS
 > do
 > echo $host;
 > ssh $host "cd
/data/fxa/INSTALL/security_patches/scripts;
./ready64bit.sh > /dev/null 2>&1" &
 > done
```

2. The script will take about 5 minutes to run. Use the following to confirm that it is complete [do NOT go on to Step 3 before it is complete]:

```
#   for host in $LX_WORKSTATIONS $XT_WORKSTATIONS
 > do
 > echo $host;
 > ssh $host "ps -ef | grep ready64 | grep -v grep"
 > done
```

3. Check for errors in the log output for the script [do NOT go on to Step 4 before this step is complete].

```
#   for host in $LX_WORKSTATIONS $XT_WORKSTATIONS
 > do
 > echo $host;
 > ssh $host "grep -iP 'error|fail'
/var/tmp/ready64bit.logfile"
 > done
```

4. Reboot each workstation. Workstations will boot up into 64bit (orange splash screen). Once all workstations have booted into 64bit mode, run the following to update the 64bit instances:

```
#   for host in $LX_WORKSTATIONS $XT_WORKSTATIONS
 > do
 > echo $host;
 > ssh $host "cd
```

– Hard copies uncontrolled. Verify effective date before use. –

```
/data/fxa/INSTALL/security_patches/scripts;
./update64bit.sh > /dev/null 2>&1" &
 > done
```

5. The update takes about 5-10 minutes. Use the following to verify that the script is done [do NOT go on to Step 6 before the update is complete]:

```
#   for host in $LX_WORKSTATIONS $XT_WORKSTATIONS
 > do
 > echo $host;
 > ssh $host "ps -ef | grep update64 | grep -v grep"
 > done
```

6. Check for errors in the log output for the script [do NOT go on to Step 7 before this step is complete].

```
#   for host in $LX_WORKSTATIONS $XT_WORKSTATIONS
 > do
 > echo $host;
 > ssh $host "grep -iP 'error|fail'
/var/tmp/update64bit.log"
 > done
```

7. Run the Visualization Applications deployment script.

```
#   cd /data/fxa/INSTALL/awips2/scripts
#   ./caveInstall.sh install
#   ./rehostInstall.sh install
```

c. Install the Qpid Server

Log into DX1 as user root through a terminal server connection. Follow these steps to install the necessary software on the CPSBN1 and CPSBN2:

1. If your site has remote cpsbn servers (VRH, PBP, HFO, GUM), issue the following command:

```
#   ssh root@px1
```

Otherwise, issue the following command:

```
#   ssh root@cpsbn1.
```

2. Install the AWIPS II software

```
# mount nas1:dataFXA /data/fxa
# cd /data/fxa/INSTALL/awips2/scripts
# ./qpidInstall.sh install
```

```
# service qpidd stop
```

3. Start up the upstream LDM server

   ```
   # chkconfig sbncp off
   # chkconfig ldmcp on
   # service sbncp stop
   # ps -wef|grep sbn|grep -v grep
   ```

   <u>Expected Results</u>

   Nothing should return from this command.

   ```
   # service ldmcp start
   ```

4. SSH into the secondary server. If your site has a remote CPSBN server (VRH, PBP, HFO, GUM) then issue the following:

   ```
   # ssh root@px2
   ```

   Otherwise issue this command:

   ```
   # ssh root@cpsbn2
   ```

5. Install the AWIPS II software.

   ```
   # mount nas1:dataFXA /data/fxa
   ```

   ```
   # cd /data/fxa/INSTALL/awips2/scripts
   ```

   ```
   # ./qpidInstall.sh install
   ```

6. Start up the upstream LDM server.

   ```
   # chkconfig sbncp off
   # chkconfig ldmcp on
   # service sbncp stop
   # ps -wef|grep sbn|grep -v grep
   ```

   <u>Expected Results</u>

   Nothing should return from this command.

   ```
   # service ldmcp start
   ```

7. Exit from the SSH connections.

   ```
   # exit
   ```
   (exits from the ssh connection into the secondary server)
   ```
   # exit
   ```
   (exits from the ssh connection into the primary server)

d. Install the EDEX Servers (DX3 & DX4)

1. Log into DX3 as user *root* using SSH.

   ```
   # ssh root@dx3
   ```

2. Ensure that the /awips2/edex/data partition is mounted from the NAS share.

   ```
   # mount –l | grep aiidata
   ```

   Expected Results

   ```
   nas1:/aiidata on /awips2/edex/data type nfs
   (rw,tcp,timeo=600,actimeo=0,addr=165.92.xxx.xxx)
   ```

3. Verify that the /awips2/GFESuite partition is mounted from the NAS share.

   ```
   # mount –l | grep GFESuite2
   ```

   Expected Results

   ```
   nas1:/GFESuite2 on /awips2/GFESuite type nfs
   (rw,tcp,timeo=600,noac,addr=165.92.xxx.xxx)
   ```

4. Install the AWIPS II EDEX Server.

   ```
   # cd /data/fxa/INSTALL/awips2/scripts
   # ./edexInstall.sh install
   ```

5. Log into DX4 as user *root* using SSH

   ```
   # ssh root@dx4
   ```

6. Ensure that the /awips2/edex/data partition is mounted from the NAS share.

   ```
   # mount –l | grep aiidata
   ```

   Expected Results

   ```
   nas1:/aiidata on /awips2/edex/data type nfs
   (rw,tcp,timeo=600,actimeo=0,addr=165.92.xxx.xxx)
   ```

7. Verify that the /awips2/GFESuite partition is mounted from the NAS share.

   ```
   # mount –l | grep GFESuite2
   ```

   Expected Results

   ```
   nas1:/GFESuite2 on /awips2/GFESuite type nfs
   (rw,tcp,timeo=600,noac,addr=165.92.xxx.xxx)
   ```

8. Install the AWIPS II EDEX Server

```
# cd /data/fxa/INSTALL/awips2/scripts
# ./edexInstall.sh install
# exit (exits from ssh session to DX4)
```

e. Install the EDEX Database Servers (DX1 & DX2)

Log into DX1 as user root through a terminal server connection. Follow these steps to install the necessary software on the DX1 and DX2 servers:

1. Ensure you are currently on the DX1 server:

```
#  hostname
```

Expected Results

```
 dx1-ccc
```
Where ccc is your platform identifier.

2. Verify that the proper mounts are in place before proceeding:

```
#  mount -l | egrep 'aiidb'
```

Expected Results

```
/dev/mapper/vg_aiidb-awipsiidb on /awips2/data type ext2
(rw)
```

3. Install AWIPS II software

```
#  cd /data/fxa/INSTALL/awips2/scripts
#  ./dbInstall.sh install
#  ./ldmInstall.sh install lll
```

Where lll is the localization ID and only run at GUM or HFO

4. Verify PostgreSQL authentication

```
#  mySubnet=$( ip addr show bond0 | grep -w inet |
head -1 | awk '{print $2}' | cut -f1-3 -d. )
#  echo $mySubnet
```

Expected Results

```
165.92.XXX
```

Where XXX is the proper third octet to your site's local area network.

This also applies for the next expected results section (below).

```
#  grep $mySubnet /awips2/data/pg_hba.conf
```

Expected Results

host	fxatext	all	165.92.XXX.0/24	trust
host	hd_ob83oax	all	165.92.XXX.0/24	trust
host	dc_ob7oax	all	165.92.XXX.0/24	trust
host	hmdb	all	165.92.XXX.0/24	trust
host	metadata	all	165.92.XXX.0/24	md5
host	maps	All	165.92.XXX.0/24	md5
host	postgres	All	165.92.XXX.0/24	md5

5. Configure the AWIPS II Radar Server (RCM) by migrating the AWIPS I configuration.

```
# scp
dx3:/awips2/edex/lib/plugins/com.raytheon.uf.common.ut
il.jar /awips2/rcm/lib/
# chown awips:fxalpha /awips2/rcm/lib/
com.raytheon.uf.common.util.jar
#  /awips2/rcm/bin/importAwips1
#  cd /awips2/rcm/data/config/persist
# sed -i -e "/archiveRoot/s/\/tmp\/sbn/\/data_store/g"
config.xml
# scp /data/fxa/radar/lists/rps-[RS]PGOP-tcp.storm
/awips2/rcm/data/config/drop-ins
# scp /data/fxa/radar/lists/rps-RPGOP-tcp.clear-air
/awips2/rcm/data/config/drop-ins
# scp /awips/fxa/data/prodList.txt
/awips2/rcm/data/config/drop-ins
```

Check to see if the site is a national reporter by issuing the following:

```
# ssh dx2 "grep ^[KPT]XXX
/awips/fxa/data/wmoSiteInfo.txt" | cut -c31- |
grep -E '[A-Z]..Y'
```

Where [KPT]XXX is the proper 4-letter AWIPS ID (e.g., KLWX or TJSJ) of the platform being installed.

If you get a line returned, issue the following (if you get NOTHING returned, SKIP this step):

```
# sed -i -e "6s/false/true/1" config.xml
```

If you have remote CPSBN servers (VRH, PBP, HFO, GUM), then issue the following command:

```
# sed -i -e "/connectionURL/s/localhost/px1f/g"
config.xml
```

All other sites issue this command:

```
# sed -i -e "/connectionURL/s/localhost/cp1f/g"
config.xml
```

Fix the max rps list sizes for AWIPS2 configuration

```
# sed -i "s/maxRpsListSize>150/maxRpsListSize>300/"
config.xml
```

6. Log into DX2 for installation activities.

```
#  ssh dx2
```

7. Verify that the proper mounts are in place before proceeding:

```
#  mount -l | egrep 'aiihdf5'
```

Expected Results

```
/dev/mapper/vg_aiihdf5-awipsiihdf5 on /awips2/edex/data/hdf5
type ext3 (rw)
```

8. Install AWIPS II software.

```
#  cd /data/fxa/INSTALL/awips2/scripts
#  ./dbInstall.sh install
#  ./ldmInstall.sh install lll
```
Where lll is the localization ID and only run at GUM and HFO
```
#  exit  (exits from script)
```

9. Exit SSH connection to DX2.

```
#  exit
```

f. Check on CAVE and Rehosted Server installation

1. Log into DX1 as user root using SSH.

```
# ssh root@dx1
```

2. Run the installation monitor scripts.

```
#  cd /data/fxa/INSTALL/awips2/scripts
#  ./caveMonitor.sh
#  ./rehostMonitor.sh
```

3. Make sure /awips2/edex/data/share is mounted on all workstations.

```
#  for host in $LX_WORKSTATIONS $XT_WORKSTATIONS; do echo
$host; ssh $host "mount /awips2/edex/data/share"
```

g. Migrate ADAM Configuration to AWIPS System

1. Ensure that you are on DX3. If not, ssh into DX3

```
# hostname
```

Expected Results

   dx3-ccc (where CCC is the platform identification where install is
   occurring)

If dx3 is not returned, ssh into dx3:

```
# ssh root@dx3
```

2. Extract the ADAM configuration from the backup archive.

```
# cd /awips2/edex/data/utility
# tar -xzpvf /data/fxa/adam_backup/adam-utility.$(
date +%A ).tar.gz
# cd /
# tar -xzpvf /data/fxa/adam_backup/adam-localapps.$(
date +%A ).tgz
# cd /awips2/edex/data/share
# tar -xzpvf /data/fxa/adam_backup/adam-share.$( date
+%A ).tgz
# cd /awips2/edex/data/utility
# rm -f edex_static/site/LLL/config/gfe/siteConfig.py
```

Where LLL = localization ID of installed site.

3. Check the SEND_ISC_ON_SAVE token in the AWIPS II localConfig.py. If it
is different from the AWIPS I localConfig.py, then change it to match.

```
#grep SEND_ISC_ON_SAVE
/awips/GFESuite/primary/etc/SITE/localConfig.py
# grep SEND_ISC_ON_SAVE
edex_static/site/LLL/config/gfe/localConfig.py
```

4. Remove references to pghost from site-level **Apps_defaults** file.

```
# sed -i '/pghost/d'
common_static/site/LLL/hydro/Apps_defaults
```

5. Ensure proper ownership

```
# chown –R awips:fxalpha /awips2/edex/data/utility
```

6. Ask the site if it has migrated trigger scripts on its ADAM. If so, copy the **siteTrigger.template** down from ADAM.

```
# cd /data/fxa/siteConfig/textApps
# cp –a siteTrigger.template AWIPS2-
siteTrigger.template
# scp adam1:$(pwd)/siteTrigger.template ./AWIPS2-
siteTrigger.template
# mv siteTrigger.template AWIPS1-siteTrigger.template
# ln –s AWIPS2-siteTrigger.template
siteTrigger.template
```

h. Run the Rehost Script and Import Databases

Due to video driver/X restarts, you must use KVM to external Console to log into dx1 as root to perform these steps:

```
# cd /data/fxa/INSTALL/<RelID>/REHOST_CODE
# ./rehost_awips.sh setup LLL
# exit (exits script)
# ssh dx2
# hb_halt a2dx2apps
# hb_run a2dx2apps
```

Verify that a2dx2apps comes up properly before proceeding.

```
# exit (to go back to dx1)
# cd /data/fxa/sdc/
# scp adam1:/usr/local/ldm/etc/pqact.conf.lll
/usr/local/ldm/etc
# ./config_awips2.sh db LLL
```

Note: Check the output of the script and look for "ERROR: out of memory" messages when importing the FFMP shapefiles. Or run "`grep memory /data/fxa/sdc/logs*`". If this ERROR occurs, call the NCF for support.

Where lll and LLL = localization ID of installed site. This will restore the AWIPS I database backups and import the FFMP shapefiles.

```
# ./config_awips2.sh ldm LLL
# ssh dx2
# scp adam1:/usr/local/ldm/etc/pqact.conf.lll
/usr/local/ldm/etc
# cd /data/fxa/sdc
# ./config_awips2.sh ldm LLL
# exit (exits back to dx1)
```

i. Start EDEX on DX3 and DX4

1. Set edex start configuration and import aviation climate

```
# ssh dx3
# cd /data/fxa/sdc
# ./config_awips2.sh edex LLL
    Press Y and enter when prompted to create setup.env
# ./config_awips2.sh cave LLL
# cd /data/fxa/A2_SysMonitor/scripts
# ./detectLocalization.sh
# exit (exits back to dx1)
```

2. Start EDEX on both DX3 and DX4.

```
# ssh root@dx3 "service edex_camel start"
# ssh root@dx4 "service edex_camel start"
```

3. Verify that EDEX comes up on DX3.

```
# ssh dx3
# tail –f /awips2/edex/logs/edex-ingest-$( date
+%Y%m%d ).log
# exit
```

4. Run config_awips2

```
# cd /data/fxa/sdc
# ./config_awips2.sh triggers LLL
```

Where LLL is the AWIPS ID of the localization running on the system.

5. Import the textproduct info table from AWIPS I. Due to long run time, put the process in the background.

```
# cd /data/fxa/INSTALL/<A2_REL_ID>/REHOST_CODE
# ./importTPI.sh &
```

6. Mount /awips2/edex/data/manual on px1 and px2.

```
# ssh px1 "mount -a && ls -ld
/awips2/edex/data/manual"
# ssh px2 "mount -a && ls -ld
/awips2/edex/data/manual"
```

j. Install LAPS/MSAS

1. SSH into PX1 and run the install script.

```
# ssh root@px1
# cd /data/fxa/INSTALL/<A2_REL_ID>/REHOST_CODE/gsdA2PX
# script -a -f backup.out
# ./backup.sh
# exit (exits script)
# cd /data/fxa/INSTALL/awips2/scripts
# ./gsdRun
```
  - **TYPE** install and press **ENTER** when prompted
```
# exit (exits ssh)
```

2. SSH into PX2 and run install scripts.

```
# ssh root@px2
# cd /data/fxa/INSTALL/awips2/scripts
# ./gsdRun
```
  - **TYPE** install and press **ENTER** when prompted
```
# exit (exits ssh)
```

3. SSH into DX3 and run the install script.

```
# ssh root@dx3
# cd /data/fxa/INSTALL/awips2/scripts
# ./gsdRun
```
  - **TYPE** install and press **ENTER** when prompted
```
# chmod 775 /awips2/edex/data/manual
# service edex_camel restart ingest ingestGrib
# exit
```

4. SSH into DX4 and restart ingest and ingestGrib

```
# ssh root@dx3
# service edex_camel restart ingest ingestGrib
# exit
```

3. Post-Install

1. Start LDM on LDAD:

```
# ssh ls1
# su – ldm
# ldmadmin start
# exit
# exit
```

2. Import the A1 Active Table into A2:

```
# cp /awips/GFESuite/primary/data/vtec/active.tbl /tmp

# gzip /tmp/active.tbl

# /awips2/GFESuite/bin/ingestAT -s XXX -h ec -a active
-n -f /tmp/active.tbl.gz
```

3. Import the A1 GFE databases into A2.

```
# ssh dx3
# su – awips
# cd /awips2/GFESuite/bin
# ./iscMosaic –n -f /data/fxa/TEMP/LLL_Fcst_BU –d
LLL_GRID__Fcst_00000000_0000
# ./iscMosaic –n -f /data/fxa/TEMP/LLL_ISC_BU
LLL_GRID__ISC_00000000_0000
```

4. Import spotters.dat file

```
# ssh dx1
# cd /data/fxa/sdc/
# ./config_awips2.sh spotters LLL
```

G. Security Patches

1. Launch the security patch install.

```
# cd /data/fxa/INSTALL/security_patches/scripts
# ./kickoff_patch_install.sh
# watch ./monitor_sec_patches.sh
```

2. Once the install is complete for all machines, hit ctrl-c to kill the watch process. If any errors are detected, call the NCF for support. [**Note**: **./monitor_sec_patches.sh** can be run by itself to get a report of the current status of the patch install. Also once a run is reported as finished,

./**kickoff_patch_install.sh** can be rerun and the script will only attempt to install missing patches.]

H. Setup

1. Set Up the AWIPS1 Rehosted Applications

Note on Rehosted Applications: Rehosted applications are managed via heartbeat, much the same way as AWIPS1 heartbeat operated.

There are only five packages for AWIPSII: a2cp1apps, a2dx1apps, a2dx2apps, a2px1apps and a2px2apps. Use the familiar commands to manage these packages: hb_run, hb_halt, hb_swap and hb_stat. AWIPSII rehost crons are stored in a2cp1cron, a2cp2cron, a2dx1cron, a2dx2cron, a2px1cron and a2px2cron. Site-level crons can be added to the a2SITEcp1cron, a2SITEcp2cron, a2SITEdx1cron, a2SITEdx2cron, a2SITEpx1cron and a2SITEpx2cron. By default the SITE level crons are empty, so be sure to add in any crons that are still necessary for site operations, such as for run_hg_genXML, nrldb and climate. Also be sure to modify or shut down crons that run from any local hardware you may have on AWIPS.

2. Set Up BOIVerify

The BOIVerify crons will run as part of a2px2apps. A base cron file has been installed in /etc/ha.d/cron.d/a2SITEpx2cron on px1 and px2. Modify this cron for your site's BOIVerify needs (checking against your AWIPS1 cron entries), and then run "hb_halt a2px2apps; hb_run a2px2apps" on px2 to activate your new cron.

The BOIVerify scripts have been rehosted to work with AWIPS2. The configuration is identical though, and can be copied over from your A1 BOIVerify directory. The following items should be copied over to /data/verify on px1:

Grids/
Stats/
EditAreas.dat
FcstrNum.dat

Copy the BiasCorrModels.txt and place it in /data/verify/scripts on px2.

3. Install NMAP (OPTIONAL)

For those sites that use NMAP, do the following to install an AWIPS II-compatible version.

```
# cd /data/fxa/INSTALL/<A2_REL_ID>/REHOST_CODE/nmap
# ./installNMAP_A2
```

5. AUTOMATION TOOL (CONFIG_AWIPS2)

The Site Data Configuration Automation Tool (**config_awips2.sh**) was developed to aid in the manual migration task of creating AWIPS II configuration files from AWIPS I configuration files. These files are predominately used by EDEX to localize the software to the normal configuration of a specific WFO or RFC. Any functions of the tool that perform this action are referred to as migration actions in this document.

Through the evolution of the tool, areas were identified where it could be used for ongoing maintenance support of an AWIPS II installation. Any functions of the tool that perform this type of action are referred to as maintenance actions in this document.

A. Assumptions

The assumptions and requirements identified in **Table 5-1** are assumed to be true for the system on which the tool is being run. If any are not true, contact your support person.

Table 5-1. Site Data Configuration Automation Tool: Assumptions and Requirements

Assumption/Requirement	How to Check …	
	Baseline AWIPS System	Standalone (e.g., ADAM)
AWIPS II Database Engine (PostgreSQL) running on specified database server.	# ssh dx1f "ps –wef|grep postmaster"	# ps –wef|grep postmaster
AWIPS II QPID running on JMS host	# ssh cp1f "ps -wef|grep qpidd"	# ps -wef|grep qpidd
AWIPS II Pypies running on the database server.	# ssh dx2f "ps -wef|grep pypies"	# ps -wef|grep pypies
AWIPS II EDEX servers **NOT** running if the following areas are to be configured: • FFMP Shapefiles • Importing of databases • EDEX setup.env file.	# ssh dx3 service edex_camel status # ssh dx4 service edex_camel status	# service edex_camel status
User *root* access on the device from which the tool will be executed.		
Necessary AWIPS I files are available on device from which script is being executed.	See running collect_files in this document.	

B. Running the Tool

Running the Site Data Configuration Automation tool is done as user root on a specific device depending on what function you are trying to achieve. The general syntax is as follows:

./config_awips2.sh *action* **LLL**

Where *action* above is a configured set of configuration functions within the tool and **LLL** is your AWIPS localization ID.

The script is dependent on a few environmental variables. These can be set before running. If they are not, and config_awips2.sh cannot determine proper values, it will suggest values to the user. **Table 5-2** lists the environmental variables and their normal settings.

Table 5-2. Automation Tool Environmental Variables

Variable	Normal AWIPS System Value	Normal ADAM Value
EDEX_HOME	/awips2/edex	/awips2/edex
EDEXSVR	dx3	adam1
EDEXDBSVR	dx1f	adam1

Table 5-3 lists the actions that can be passed and where config_awips2.sh should be run for those specific functions in order to configure.

Table 5-3. Automation Tool Function and Execution Server

Function	Description	Execution Server
All	When passed to config_awips2.sh this function will run through almost all the created migration operations. It begins with running the functions designated as EDEX functions, then runs those designated as CAVE functions, followed lastly by those listed as other functions. These actions are **NOT** performed by passing all to config_awips2: • GFE Migration tasks • Procedure conversion tasks • Trigger loading	✓ ADAM only (not designed for an AWIPS system)
edex	When passed to config_awips2.sh this function will run through all EDEX designated migration tasks. Tasks include: • Configuring plugin-filters which filter specific data into, or excluded from, other plugins. An example is a metarToShef plugin filter, that sends metars which are inside a certain geographic area into the shef plugin as	✓ ADAM ✓ DX3 or DX4 (not both)

Function	Description	Execution Server
	well as the metar plugin. • Copies spi files necessary for data storage into the AWIPS II configuration tree. • Copies hydro specific files from /awips/hydroapps into the AWIPS II specific area. This includes the Apps_defaults_site. • Configures GFE siteConfig.py file. • Configures EDEX setup.env configuration file. • Configures AWIPS II hydro set_hydro_env file. • Configures baseline FFMP runtime configuration file. See **ffmp** below. • Copies supported files into the ndm endpoint for updating in the AWIPS II system. See **ndm** below. • Configures shef distribution patterns to allow the proper data into the shef decoding plugin. • Configures grid definition file for the MPE and DHR-MOSAIC grids based on site specific files. • Migrates Radar Multiple Request configuration into AWIPS II configuration location.	
cave	When passed to config_awips2.sh this function will run through all CAVE designated migration tasks. Tasks include: • Migration of spotters from spotters.dat • Baseline configuration of Warngen GUI • Creation of input file for radar menu creation • Creation of input file for upper air menu creation • Migration of AvnFPS configuration from AWIPS I files. • Configuration of Local and Regional Warnings menu • Configuration of Text Workstation products menu • Configuration of scale selector in D2D • Delivery of local map scale bundles. • Copy of ldad15.spi and ldad15prcp.spi into basemaps • Migration of text alarm/alert settings from AWIPS I • Migration of the Fields menu in Volume Browser from AWIPS I	✓ ADAM ✓ DX3 or DX4 (not both)
avnfps	AvnFPS migration function. When passed to config_awips2.sh this function performs the following tasks: • Migrates AWIPS I AvnFPS configuration files to AWIPS II • Creates DEFAULT configuration file, if possible, or alerts user that one needs to be created. • Creates AWIPS II AvnFPS forecaster list from AWIPS I forecasters file. Dependent on system user accounts in /etc/passwd • Migrates AWIPS I AvnFPS climate files to AWIPS II	✓ ADAM ✓ DX3 or DX4 (not both)

Function	Description	Execution Server
db	Database configuration function. When passed to config_awips2.sh this function performs the following tasks: • Searches for database backups in the standard location (/data/fxa/DAILY_BACKUP/postgres) which were from the last 24 hours, and asks the user if loading is desired. Configures pg_hba.conf appropriately based on the answers. • Imports FFMP shapefiles into the ffmp_streams and ffmp_basins tables of the maps database. • Imports Warngen LocalCitiesInfo.txt into database	✓ ADAM ✓ DX1 (floater)
ffmp	FFMP migration function. When passed to config_awips2.sh this function creates a base localized FFMPRunConfig.xml file.	✓ ADAM ✓ DX3 or DX4 (not both)
gfem	**Not run automatically by edex, cave or all options.** GFE migration scripts. Wrapped in migration python scripts. Dependent on collect_files.sh run, and can only be run on ADAM. Migrates the following GFE configurations: • GFE Edit Areas • GFE Color Tables • GFE Weather Element Groups • GFE Configs • GFE Sample Sets • GFE Timeranges • GFE Combinations • Create GFE Parameter Info Files from CDLs	✓ ADAM
grids	Grid definition maintenance function. When passed to config_awips2.sh this function creates grid definition file for MPE and DHR-MOSAIC grids.	✓ ADAM ✓ DX3 or DX4 (not both)
hydro	Hydro migration function. When passed to the config_awips2.sh this function performs the following: • Migration of AWIPS I .Apps_defaults_site to AWIPS II Apps_Defaults • Migration of hydro geo_data to AWIPS II • Migration of hydro mpe files to AWIPS II • Migration of xdat configuration to AWIPS II (RFC Only) • Migration of group_definitions.cfg to AWIPS II • Migration of shef metar.cfg to AWIPS II • Modification of AWIPS II set_hydro_env	✓ ADAM ✓ DX3 or DX4 (not both)

Function	Description	Execution Server
ingest	When passed to config_awips2.sh this function will perform the following actions: • Configuring plugin-filters which filter specific data into, or excluded from, other plugins. An example is a metarToShef plugin filter, that sends metars which are inside a certain geographic area into the shef plugin as well as the metar plugin. • Configures shef distribution patterns to allow the proper data into the shef decoding plugin.	✓ ADAM ✓ DX3 or DX4 (not both)
ldm	LDM maintenance function. When passed to config_awips2.sh this function will create the active pqact.conf file by doing the following: • Starts with the pqact.conf.template baseline • Appends localized radar and hydro patterns based on same logic as AWIPS I. • Appends site manually migrated pqact.conf.lll (where lll = site ID), which should be migrated form of acqPatternAddOns.txt	✓ ADAM ✓ DX1 and DX2
maps	When passed to config_awips2.sh this function will perform the following actions: • Copies goesBufr.spi and poesBufr.spi from AWIPS I into AWIPS II basemaps • Copies cities.lpi, ldad15.spi and ldad15prcp.lpi into AWIPS II basemaps. • Extracts any delivered map bundles from cave_static_site.tar.gz (e.g., WFO.xml) • Creates scale selector pull-down from AWIPS I scaleInfo.txt file	✓ ADAM ✓ DX3 or DX4 (not both)
mpe	MPE activation maintenance function. When passed to config_awips2.sh this function runs the following executables to initialize MPE server side configuration files. **Note: EDEX must be running for this action** • run_create_mpe_gage_file • run_create_mpe_station_lists • run_create_freezing_station_list • run_create_mpe_climo_lists • run_create_triangles • run_create_mpe_beam_height_file • run_mpe_fieldgen 3 • run_dqc_preprocessor	✓ ADAM ✓ DX3 or DX4 (not both)

Function	Description	Execution Server
ndm	National Dataset Management (NDM) maintenance function. Copies the following files into the ndm endpoint for EDEX ingest into AWIPS II system: • afos2awips.txt • metarStationInfo.txt • modelBufrStationInfo.txt • MTR.primary • MTR.goodness • goesBufrStationInfo.txt • poesBufrStationInfo.txt • maritimeStationInfo.txt • raobStationInfo.txt • raob.goodness • raob.primary • ispan_table.dat (if exists)	✓ ADAM ✓ DX3 or DX4 (not both)
procs	**Not run automatically by edex, cave or all options.** D2D Procedure and Color Curve migration function. Wraps in bundle converter executable.	✓ ADAM
shp	Shape file maintenance function. When passed to config_awips2.sh performs the following tasks: • Loads in FFMP shapefiles into maps database. **Only if forced (-f).** • Looks for local shapefiles staged in /awips2/edex/data/utility/edex_static/site/LLL/shapefiles and imports into maps database. Also creates map bundle xml so it can be selected from CAVE. • Looks for updates to known national shapefiles downloaded in accordance with SMM.	✓ ADAM ✓ DX1 (floater)
spotters	When passed to config_awips2.sh this function will load in the spotters from spotters.dat	✓ ADAM ✓ DX1 (floater)
triggers	When passed to config_awips2.sh this function will load triggers (subscriptions) into the AWIPS II system utilizing the same files as AWIPS I.	✓ ADAM ✓ DX1 (floater)
wg	Warngen configuration option. When passed to config_awips2.sh this function will perform the following tasks: • Creates a baseline Warngen GUI configuration pulling from AWIPS I wwaConfig.txt file.	✓ ADAM ✓ DX3 or DX4 (not both)

The tool can also be run with optional arguments. **Table 5-4** describes the different arguments that can be passed to the script.

Table 5-4. Optional Arguments for config_awips2.sh

Argument	Description
-f	This is the force option. It will force the migration of specific files from AWIPS I configuration files.
-d	Sets the DEBUG flag. Will output more information to the user.
-e	Sets EDEX_HOME variable. Example: -e /awips2/edex

C. Identified Maintenance Tasks

Through the course of developing the migration tool, certain tasks were identified as being useful for maintaining an AWIPS II installation beyond an initial migration. These tasks are identified and explained in this section.

1. Creating the Active pqact.conf LDM Ingest File

AWIPS II utilizes the UniData LDM software for ingest of SBN data. The downstream LDM server will decide what products to store in raw format using an active pqact.conf file. The locations and file name of the files are as follows:

AWIPS II System: /usr/local/ldm/etc/pqact.conf

ADAM: /usr/local/ldm/etc/adam-pqact.conf

The active pqact.conf file should not be manually edited, as this file is created using config_awips2.sh script based on input files. These are the files that should be modified to effect changes in the active pqact.conf file. **Table 5-5** identifies and describes the input files

Table 5-5. Input Files to Active pqact.conf Creation

File Name	Description
/usr/local/ldm/etc/pqact.conf.template	Delivered baseline template. This file is delivered with the awips2-ldm RPM and risks being overwritten on a software installation. This file should not be edited.
/usr/ocal/ldm/etc/pqact.conf.xxx	Where xxx = AWIPS Site ID for ingest localization of the AWIPS system. This file should be a migration of the AWIPS I file acqPatternAddOns.txt input file. Site customizable file to allow addition of new patterns into the SBN ingest stream.
/usr/local/ldm/etc/pqact.conf.local	This file is created new on running of ./config_awips2.sh ldm LLL Included contents are NNEXRAD (SBN radar data), hydro shef patterns built by localization in AWIPS I.

Once the edits are made to the correct input files, the active pqact.conf file can be created with the following options being passed to the automation tool:

```
./config_awips2.sh ldm LLL
```

Where LLL is the AWIPS site ID of the ingest site being configured on the AWIPS II system

2. Triggers

The AWIPS II SDC uses the same input files to populate the trigger database as the AWIPS I mainScript.csh program. These files are identified in **Table 5-6**.

Table 5-6. Input Files to Trigger Loading in AWIPS II

Input File	Description
ldadTrigger.template	LDAD actions file.
	Uses LLL-ldadConfig.txt to substitute wildcards in the trigger template
ldadSiteBackupTrigger.template	LDAD backup site trigger file.
	Uses LLL-ldadSiteConfig.txt to substitute wildcards in the trigger template.
adaptTrigger.template	ADAPT action file.
	Uses LLL-adaptSiteConfig.txt to substitute wildcards in the trigger template
LLL-wsoTrigger.template	Provides access to non-AWIPS NWS Sites' (Alaska Region) products in the WWA monitor
LLL-faxTrigger.template	Created by using Configure Auto Fax interface on the text workstation.
siteTrigger.template	Site customizable trigger template.
hazCollectTrigger.template	HazCollect triggers
or	Uses LLL-hazCollectSiteConfig.txt to configure triggers.
LLL-hazCollectTrigger.template	

To load triggers into an AWIPS II system, these files must be present and properly edited. The following command should be run:

```
./config_awips2.sh triggers LLL
```

If a trigger is being deleted, then the configuration must be *forced* (-f option). This causes the script to first delete all products from the subscription.subscriptions database schema before loading all triggers new.

Remember this when adding or modifying triggers in siteTrigger.template: The file needs to be in /data/fxa/siteConfig/textApps. This file will be transferred to the AWIPS II system during installation.

3. National Data Management (NDM) Files

The site data configuration automation tool will copy those national data management files, already maintained on the AWIPS system, into the endpoint for EDEX, which allows the software to ingest and localize properly based on the file.

The files that are supported are identified in **Table 5-7**.

Table 5-7. NDM Files Supported by AWIPS II ndm Endpoint

Input File Name	Staging Location	Output
afos2awips.txt	/awips/fxa/data	afos_to_awips table in the fxatext database
afos_lookup_table.dat	/awips/fxa/data	/awips2/edex/data/utility/common_static/base/textdb/afos_lookup_table.dat
modelBufrStationInfo.txt	/awips/fxa/data	modelBufr.spi in /awips2/edex/data/utility/common_static/site/LLL/basemaps
MTR.primary MTR.goodness	/awips/fxa/data/	MTR.spi, MTR.primary, MTR.goodness in /awips2/edex/data/utility/common_static/site/LLL/basemaps
goesBufrStationInfo.txt	/awips/fxa/data	goesBufr.spi in /awips2/edex/data/utility/common_static/site/LLL/basemaps
poesBufrStationInfo.txt	/awips/fxa/data	poesBufr.spi in /awips2/edex/data/utility/common_static/site/LLL/basemaps
maritimeStationInfo.txt	/awips/fxa/data	BUOY.spi in /awips2/edex/data/utility/common_static/site/LLL/basemaps
metarStationInfo.txt	/awips/fxa/data	common_obs_spatial table in metadata database
textCCChelp.txt textNNNhelp.txt textOriginTable.txt textCategoryClass.txt	/awips/fxa/data	/awips2/edex/data/utility/common_static/base/textdb/
synopticStationTable.txt	/awips/fxa/data	edex_static/base/synopticStationTable.txt and common_obs_spatial table in metadata database
raob.goodness raob.primary	/data/fxa/nationalData	raob.spi in /awips2/edex/data/utility/common_static/site/LLL/basemaps
raobStationInfo.txt	/awips/fxa/data	common_obs_spatial table in metadata database
ispan_table.dat	/awips/fxa/data/ localizationDataSets/LLL	ispan_table in /awips2/edex/data/utility/edex_static tree

– Hard copies uncontrolled. Verify effective date before use. –

Input File Name	Staging Location	Output
fsl-w88d.dbf fsl-w88d.shp fsl-w88d.shx	/data/fxa/nationalData	radar_spatial table in metadata database.
national_category_table .dat	/data/fxa/nationalData	/awips2/edex/data/utility/common_static/base/textdb/national_category_table.template
afosMasterPIL.txt	/data/fxa/nationalData	/awips2/edex/data/utility/common_static/base/textdb/afosMasterPIL.txt

By running the following command, the user will be presented with a select menu allowing files to be copied into the ndm endpoint for EDEX to ingest and create the proper output file:

```
./config_awips2.sh ndm LLL
```

This task will need to be run whenever there are NDM file updates.

4. Shapefiles

a. Local Shapefiles

Local shapefiles can be imported into the maps database using the automation tool. Files should be staged in the following location:

```
/awips2/edex/data/utility/edex_static/site/LLL/shapefiles
```

The shapefiles should be added in a manner similar to the following:

```
submenu/shape_desc/shapefile.(dbf|shp|shx)
```

It is important to note a few things about this structure. First, the name of the shape file is determined by its parent directory and NOT by the name of the shapefile. Also, all underscore characters "_" will be translated to spaces in the CAVE menu.

Second, the directory name of *SUBMENU* is optional, but if desired this will create a submenu on your Maps menu in CAVE. This means that you can have a shapefile staged in this manner:

/awips2/edex/data/utility/edex_static/site/LLL/shapefiles/shape_desc/shapefile.shp

It will also create a menu entry in CAVE under **Maps** named **shape_desc**.

Any shapefiles staged within more than one directory will be listed under a submenu in the Maps pulldown in CAVE, which is named the same as the directory itself. To clarify, whatever text you put in for *submenu* above will be the text shown as the submenu on CAVE.

For example, the following staging:

> **/awips2/edex/data/utility/edex_static/site/LLL/shapefiles/local_roads/dirt_roads/dirtroad.shp**

> **/awips2/edex/data/utility/edex_static/site/LLL/shapefiles/local_roads/grass_roads/grassroad.shp**

will create the following structure in the menu of CAVE:

```
Maps → local roads → dirt roads

                   → grass roads
```

Finally, it is important to note that only ONE **shp**, **dbf**, and **shx** file can reside in the final directory for staging shapefiles. This is **NOT** valid staging:

> **/awips2/edex/data/utility/edex_static/site/LLL/shapefiles/city_parks/playgrounds.shp**

> **/awips2/edex/data/utility/edex_static/site/LLL/shapefiles/city_parks/bigfieldsa.shp**

Because there are two .shp files under local_roads, which would be used to create the "city parks" menu. In this case you probably want to create two subdirectories under city_parks – one named *playgrounds/* and another *bigfields/.*

The last important thing to note is that the directory name of *shape_desc* will determine the table name into which the shapefiles will be imported. It also determines the name of the map bundle created. The name of the map bundle is case sensitive, while the table name will always be all lower case. All underscore characters ('_') in *shape_desc* will be converted to spaces when creating the Maps menu name.

For example, the following shapefiles will create **mapdata.oax_county** schema in the maps database and create a new entry OAX County directly on the main Maps menu in CAVE:

```
/awips2/edex/data/utility/edex_static/siote/OAX/shapef
iles/OAX_County/OAX_County.dbf

/awips2/edex/data/utility/edex_static/siote/OAX/shapef
iles/OAX_County/OAX_County.shp

/awips2/edex/data/utility/edex_static/siote/OAX/shapef
iles/OAX_County/OAX_County.shx
```

Note: The staged directory name does not have to match the file name of the shape file.

In this example, the shapefiles will create **mapdata.grr_cwa_roads** schema in the maps database, and will create a **Roads** submenu in the Maps pulldown within CAVE. Under the **Roads** submenu will be a selection **GRR CWA Roads**.

```
/awips2/edex/data/utility/edex_static/site/GRR/shapefi
les/Roads/GRR_CWA_Roads/grr_cwa_roads.dbf

/awips2/edex/data/utility/edex_static/site/GRR/shapefi
les/Roads/GRR_CWA_Roads/grr_cwa_roads.shp

/awips2/edex/data/utility/edex_static/site/GRR/shapefi
les/Roads/GRR_CWA_Roads/grr_cwa_roads.shx
```

The name of the staging directory is used to create the database table name, as well as the map bundle XML file name. The bundle file name will be created in /awips2/edex/data/utility/cave_static/site/LLL/bundles/maps. For example, the first example above would create OAX_County.xml in this directory.

The label on the Maps menu in CAVE will match the name of the directory, except that every '_' in the directory name will be turned into a space.

The second example above would create **GRR_CWA_Roads.xml** in **/awips2/edex/data/utility/cave_static/site/LLL/bundles/maps/Roads.**

Note: The bundle will only be written by **config_awips2.sh** if the following is true: 1. It does not exist for 2. It exists and was created by **config_awips2.sh.** This means that if a bundle has been edited by the localization perspective since its original creation by config_awips2, it will ***not*** be overwritten.

Once everything is staged correctly, to import the shapefiles staged into the database, run the following:

```
./config_awips2.sh shp LLL
```

When the force option (-f) is added to this command, it will also call the config_ffmp_shapefiles script to load the FFMP shapefiles.

To remove a shapefile from the system, follow these steps. Note that, for these steps, the example is removing a shape file named My_Shape_File:

1. Remove the shapefile from the system. Identify the shapefile staging directory, and remove the **directory**

```
cd /awips2/edex/data/utility/edex_static/site/LLL/shapefiles
rm -rf My_Shape_File
```

2. Remove the map bundle XML file, which is also created, and which also has the same name as the staging directory.

```
cd /awips2/edex/data/utility/cave_static/site/LLL/bundles/maps
rm My_Shape_File.xml
```

3. Connect to the database, and drop the table.

```
su - awips
psql maps
DROP TABLE mapdata.my_shape_file;
DELETE FROM mapdata.map_version where
table_name='my_shape_file';
\q
```

Note : If you are uncomfortable with any of these steps. feel free to call the NCF for assistance at 301.713.9344.

b. Baseline Shapefiles

Baseline shapefiles will also be updated using the *shp* option to **config_awips2.sh**. They can be maintained in the same manner as AWIPS I. This means that you are to download the shapefile from the NOAA1 server and stage in **/data/fxa/nationalData.**

You have the option of renaming the files yourself as you would with AWIPS I. However, the **config_awips2.sh** is also aware of the AGMG naming convention, so the files **do not have to be renamed for AWIPS II purposes**. The script itself will rename the files to their proper AWIPS I names in **/data/fxa/nationalData.**

As an example, the public forecast zones can be named **c11-zone.shp** OR **z_*mmddyy*.shp**. Both will be recognized by **config_awips2.sh.** If the shapefile is

left as **z_mmddyy.shp,** the script will rename it to **c11-zone.shp** when it imports the file.

The tool works in this way: When the script is run with the *shp* argument, it will check for updated files that are newer than the import time of the corresponding table in the maps database. If any are found, it will copy them into the AWIPS II directory in **/awips2/edex/data/utility/edex_static/base/shapefiles**, and import them into the database automatically. It also makes a backup of the old shapefiles in the following form:

> **YYYYMMDD-*filename*.archive**
>
> Where filename is the name of the shapefile, for example, **usa_lake.shp.**

IMPORTANT NOTE: AWIPS II relies on accurate shapefiles. If you notice a problem with an application using a shapefile that has been updated, do *not* try and reimport the same shapefile. Call the NCF, which has procedures for importing the archived shapefile. Every time you rerun config_awips2.sh with the shp option, it will remove the archived shape files, so you could lose the "good" ones.

5. Collect Files Script

The collect_files.sh script is a tool developed to help provide the necessary AWIPS I files that config_awips2.sh uses to create the input configurations for AWIPS II software.

Usage:

```
./collect_files.sh LLL
```

Where LLL is a valid AWIPS ID.

Please note that if given an invalid AWIPS ID, it will merely not find the files necessary for site data configuration.

This script requires user interaction as it will ask if you would like to bundle the AWIPS I D2D procedures into the archive. It is *highly recommended* that this option only be used when attempting to transfer these procedures to ADAM for one-time conversion using the conversion tool.

Output:

The script will create the following file:

```
/data/fxa/TEMP/awipsIloc_lll.tar.gz
```

Where lll is the lower-case AWIPS ID passed to the collect_files.sh script.

Note: The file's name is awips followed by a capital letter i (I) (not the number one), and then loc for localization.

The tar archive can then be copied via network, or physical media, to the AWIPS II platform and unpacked following these steps:

1. Log into the AWIPS I platform as user *root.*

2. Secure copy the file across the network to the AWIPS II machine:
   ```
   # scp /data/fxa/TEMP/awipsIloc_lll.tgz hostname:/tmp/
   ```
   Where hostname is the hostname of the device onto which you are copying.

3. Ensure there are no NFS mounts currently active on the AWIPS II platform.
   ```
   # umount –tnfs –a
   ```
   If you are an RFC, issue these commands as well:
   ```
   # mount ns1:adamHYDRO /awips/hydroapps
   # mount ns1:adamREP /awips/rep
   ```

4. Unpack the archive on the platform.
   ```
   # tar –C / -xzf /tmp/awipsIloc_lll.tgz
   ```
   At this point, the AWIPS II SDC Automation Tool can be executed.

– Hard copies uncontrolled. Verify effective date before use. –

APPENDIX A. ADDITIONAL RESOURCES

The documents and tools identified in this appendix are provided additional resources with the Site Migration Guide.

Organization Aids

1. AWIPS II Site Mig Org Aids V8.xlsm

Wiki

2. AWIPS Migration Site Configuration INDEX.docx
3. NCLADT Wiki INDEX.docx

LDM Ingest

4. Ingest Filter Configuration.docx
5. LDM Ingest Checklist.docx

Subversion

6. SubversionTutorialPart1 - Erh-itproject.pdf
7. SubversionTutorialPart2 - Erh-itproject.pdf
8. SubversionTutorialPart3 - Erh-itproject.pdf
9. Apn Svn Tag Release.docx
10. HowToSubversion.docx
11. HowToSubversionNewApp.docx
12. RepoLayout.docx
13. Subversion Tips.docx
14. 1.7 svn-book-html-chunk
15. 1.7 svn-book-html-chunk.tar.bzw
16. 1.7 svn-book.pdf

D2D Procedure Conversion

17. D2D Bundle Conversion.docx

GFE Migration

18. AppCheckForPointTools.docx
19. AppConvertEditAreas.docx
20. AppGfeAfpsProxy.docx
21. BaseSiteUserDirs.docx
22. checkForPointTools doc.docx

23. CommonSmartToolChanges.docx

24. findAdamGfeFile.docx

25. GfeActivateSite.docx

26. gfeConfigFileChanges.docx

27. GfeConfigMaps.docx

28. GfeModuleInstaller.sh.docx

29. gfePorter.docx

30. GFErename modules.docx

31. HowToGfe - GFE Migration.docx

32. HowToGfePortingNotes.docx

33. HowToImportShapefile.docx

34. MultiModelSmartInits.docx

35. NCLADT Repository.docx

36. NumericToNumpy.docx

37. SmartScriptDifferences.docx

38. SmartScriptIssues.docx

39. SmartScriptUsageAnalysis.docx

40. gfeClient.docx

41. gfeClient PngWriter.docx

42. gfeClient runProcedure.docx

43. gfeClient TextProductTest.docx

44. PublishingGfeGrids.docx

45. D2DGridsinGFE.docx

Local Applications

46. Local Applications Guide.docx

47. Local Apps working session - migration.docx

48. AddLocalGrid.pdf

WarnGen

49. Adding Mile Markers in AWIPS 2.docx

50. AWIPS 2 WarnGen Documentation 1.3.1.doc

51. A2 Warngen Localization_Combine Marine Zones.docx

52. A2 Warngen Localization_Create Site Specific Dams.docx

53. AWIPS II WarnGen Localization Tutorial.docx

54. Main Page for AWIPS Migration WarnGen Info.docx

– Hard copies uncontrolled. Verify effective date before use. –

55. WarngenDoc.docx

56. WarngenTemplateTeam.doc

Localization Doc References

57. ADAM SDC AutoTool Functions.docx

58. ADAM SDC AutoTool Log File.docx

59. Table 5.3-1.docx

60. AWIPS II Site Data Configuration & Localization: Step-by-Step Guide (A2_SDC_Localization_Guide.pdf)

System Installation

61. AWIPS II Installation Status Check.doc

62. Service Backup.pptx

63. Rollback/Forward Procedures (A2_Rollback_doc.a2.rlbk-02.pdf)

– Hard copies uncontrolled. Verify effective date before use. –