



AWIPS Flow Tag Instructions: ADE Setup

Document No. AWPR.FT.ADE.OB21.4.1-01.00
29 Jun 2023

Prepared Under
Contract EA133W-17-CQ-0082
Advanced Weather Interactive Processing System (AWIPS)
Operations and Maintenance

Submitted to:

Mr. Beraq Azeem
Contracting Officer's Technical Representative
U.S. Department of Commerce
NOAA NWS Office of Central Processing
1325 East-West Highway, Room 15130
Silver Spring, MD 20910

Prepared by:

Raytheon Intelligence & Space (RIS)
8401 Colesville Road, Suite 800
Silver Spring, MD 20910

Change History

| Remarks | Pages Affected | Date | Rev |
|---|----------------|----------------|---------------------|
| Update release with Build 17.2.1 | 9-12 | Dec. 7, 2017 | Build 17.2.1 |
| Update release with Build 17.3.1 | N/A | Apr. 26, 2018 | Build 17.3.1 |
| Update release with Build 18.1.1 | N/A | Sept. 27, 2018 | Build 18.1.1 |
| Update release with Build 18.2.1/18.2.2 | N/A | Mar. 19, 2019 | Build 18.2.1/18.2.2 |
| Update release with Build 19.1.1/19.2.1 | N/A | Aug 23, 2019 | Build 19.1.1/19.2.1 |
| Update release with Build 19.3.1 | N/A | Nov 25, 2019 | Build 19.3.1 |
| Update release with Build 20.2.3 | N/A | May 10, 2021 | Build 20.2.3 |
| Update release with Build 20.3.2 | N/A | May 17, 2022 | Build 20.3.2 |
| Update release with Build 21.4.1 | N/A | June 29, 2023 | Build 21.4.1 |

Table of Contents

| | <i>Page</i> |
|--|-------------|
| 1. Introduction | 5 |
| 2. Document Conventions | 5 |
| 3. AWIPS Integration Deployment Instructions | 5 |
| 3.1 Deploy the ADE..... | 6 |
| 3.1.1 Test Your Graphics Card (Required for CAVE Development) | 6 |
| 3.1.2 Stop and Uninstall EDEX, CAVE, and/or the ADE (if already installed) for Releases 1-7.3 and Later | 6 |
| 3.1.3 Install the ADE RPMs and the Standalone Development Environment | 8 |
| 3.1.4 Install the AWIPS II ADE source..... | 9 |
| 3.1.7 Set Up the AWIPS II ADE | 9 |
| 3.1.8 How to Run/Debug EDEX in the ADE | 14 |
| 3.1.9 How to Run/Debug CAVE in the ADE | 18 |

Acronyms and Abbreviations Used in This Document

| | |
|--|------------|
| AWIPS Development Environment | ADE |
| Advance Weather Interactive Processing System | AWIPS |
| Common AWIPS Visualization Environment | CAVE |
| Configuration Management | CM |
| Database | DB |
| Deficiency Report/Change Request | DR |
| Enterprise Data EXchange | EDEX |
| Enterprise Service Bus | ESB |
| Forecaster Integration Testing | FIT |
| Free & Open Source Software | FOSS |
| Hierarchical Data Format 5-multi-object file format for the transfer of graphical and numerical data between computers | HDF5 |
| Hypertext Transfer Protocol | HTTP |
| Yellowdog Updater, Modified | YUM |
| Java Development Kit | JDK |
| Java Runtime Environment | JRE |
| Local Data Manager-a collection of cooperating programs that select, capture, manage, and distribute arbitrary data products | LDM |
| Process ID | PID |
| Postgres Structured Query Language-FOSS database software | PostgreSQL |
| Open Source AMQP Messaging | Qpid |
| Round 1 Group 1 | R1G1 |
| Red Hat Package Manager | RPM |
| Snap Up (Delivery after TO11 Slice 6) | SU |
| Task Order | TO |
| Topographical directory/files | TOPO |
| Visualization | Viz |

1. Introduction

In order to perform many actions in this document, it is necessary to have root access or sudo privileges to access root. This Flow Tag document provides instructions for installing and setting up the AWIPS II AWIPS Development Environment (ADE). The procedures are comprehensive, enabling an engineer to conduct the setup with very little background.

2. Document Conventions

The flow tag steps are presented in outline format. All steps utilize the following typographic conventions:

A square box indicates a step to be accomplished. *Example:*

- Stop All Services:

A circle indicates a checkpoint for the step. *Example:*

- Stop Camel:

Bold font indicates keystrokes, button clicks, notes of interest, etc. *Example:*

- **./edex_camel stop**

Bold italic blue font indicates something that needs to be updated to the current value (machine name, date, etc.). *Example:*

- ***rm -rf /home/user/caveData***

Commands in parentheses are example commands based on the Omaha AWIPS II Testbed Configuration.

- **mount *edex-machine*:/awips2/edex/opt/data/hdf5 /test-data** (mount awips-dev1:/awips2/edex/opt/data/hdf5 /test-data)

3. AWIPS Integration Deployment Instructions

To make things faster and more accessible for multiple deployments, you may want to copy the files from the delivery media to a network location.

For the purposes of this document, we will use *[path to installers]* in place of this location.

For example, usually the path to the delivery files for the OMAHA AWIPS Integration deployment is **/awipscm/AWIPSDVDs/AWIPS_XXX_Installers** (where the **XXX** is the Release/Build name/number).

3.1 Deploy the ADE

3.1.1 Test Your Graphics Card (Required for CAVE Development)

QA ENG

To run and debug CAVE, you must have a CAVE-capable graphics card. This is not needed if you will only be running and debugging EDEX.

- Test your graphics card (commands may vary based on your graphics card and the driver installed).
 - **Save everything and close all applications.** If your graphics card or driver is not correct, your XWindows may crash.
 - Open a terminal if you do not already have one.
 - **glxinfo | grep OpenGL**
 - *Make sure there are no errors and that your XWindows does not crash.*
 - *Make sure that the **OpenGL version string** is 2.0 or higher.*
 - **glxgears**
 - You should see an animation of a set of gears.
 - If neither glxinfo nor glxgears is available, you can try to open CAVE via Eclipse. (**Note:** Section 3.1.8 provides instructions on **How to Run/Debug CAVE**, but you will need to run through all the sections before the debug CAVE section.)
 - If you get any errors, if CAVE crashes, or if your XWindows crashes, try installing the latest driver for your graphics card.
 - Otherwise, you may need a CAVE-capable graphics card—one that supports OpenGL version 2.0 (or higher).

3.1.2 Stop and Uninstall EDEX, CAVE, and/or the ADE (if already installed) for Releases 1-7.3 and Later

QA ENG

Note: Root permissions are required to uninstall the ADE RPMs. Login as the root user or use sudo as in the examples.

- ☐ ____ Uninstall any previously installed AWIPS II Software.
 - First, stop any EDEX Services (ldm, camel, qpidd, postgres, and httpd-pypies), close CAVE, and/or close Eclipse.
 - Stop EDEX (all running instances)
 - Select the terminal where EDEX is running.
 - Press **Ctrl+C**.
 - If EDEX was started via service command, run **service edex_camel stop**

- as user root or via sudo to stop camel processes.
 - Ensure all camel processes are stopped by grepping for the wrappers & java.
 - **ps uax | grep wrapper** then **ps aux | grep java**
 - Use **kill** to kill any PID other than the grep process.
- Note:** If you kill java wait a moment and repeat the ps command as the wrapper may automatically restart.
- Stop Qpid.
 - As user root run **service qpid stop**
 - Ensure all Qpid processes are stopped by typing **service qpid status** or grepping for the qpid daemon
 - **ps aux | grep qpid**
 - Use **kill** to kill any pid other than the grep process
- Stop Httpd-Pypies.
 - Select the terminal where Httpd-Pypies is running and stop by running the service stop command
 - **/sbin/service httpd-pypies stop**
 - Ensure all httpd-pypies processes are stopped by grepping for them
 - **ps aux | grep httpd**
 - Use **kill** to kill any PID other than the grep process.
- Stop Postgres.
 - Select the terminal where Postgres is running.
 - Press **Ctrl+C**.
 - Ensure Postgres is stopped by grepping for postmaster
 - **ps aux | grep postmaster**
 - Use **kill** to kill any pid other than the grep process
- ____ Unpack the rpms:
 - **cd /home/user/**
 - **tar xvf <Path-to-ade-tar-file>/awips2-ade-*.tar**

Note: When you remove the ADE RPMs, other installed EDEX Services will be removed as well due to dependencies between components.

- ____ Uninstall the RPMs:
 - **sudo <Path-to-ade-tar-file>/ade_quick_uninstall.sh**
- ____ Verify that all AWIPS II components have been removed.
 - **rpm -qa | grep awips2**
- ____ Manually remove any AWIPS II components that remain (if any remain).
 - **sudo yum remove awips2-* -y**
- ____ Remove the root installation directory.
 - **sudo rm -rf /awips2/**
- ____ Remove old source code (replace *user* with your username)
 - **rm -rf /home/user/awips**

3.1.3 Install the ADE RPMs and the Standalone Development Environment

QA ENG

Note: Root permissions are required to install the ADE RPMs. Login as the root user or use sudo as in the examples. The sudo user should see an 'ADE_Eclipse' launch icon on the Desktop after the installation.

Note: EDEX requires approximately 6 GB of space.

- ____ Unpack the rpms (if not done already):
 - **cd /home/user/**
 - **tar xvf <Path-to-ade-tar-file>/awips2-ade-*.tar**
- ____ Create the awips user (if not done already):
 - **sudo useradd awips**
- ____ Create the fxalpha group (if not done already):
 - **sudo groupadd fxalpha**
- ____ Add the fxalpha group to the awips user (if not done already):
 - **sudo usermod -a -G fxalpha awips**
- ____ cd into the directory just untar'd and run the install script:
 - **cd awips2-ade-***
 - **sudo <Path-to-ade-directory>/ade_quick_install.sh**
 - **Note:** Ensure ade_quick_install.sh has executable permissions by typing **chmod +x <path-to-ade-directory>/ade_quick_install.sh**. This might need to be done as user root.

Note: At this point, you may want to re-install the Standalone Development environment. The following commands assume the user has obtained the necessary rpms for the desired Standalone build and configured the YUM repository for said rpms. To re-install, run the following commands as user root or via sudo:

- **yum groupremove "AWIPS II Standalone"**
 - **yum groupinstall "AWIPS II Standalone"**
- **Note:** If errors are received during the groupinstall that refer to missing dependencies, you will probably need to ensure log4j.x86_64 and zlib.i686 are installed by issuing the following as root: **yum install log4j.x86_64 zlib.i686**
 - **Note:** Ensure that the AWIPS2 repository contains wxGTK-<version>.el6, and NOT wxGTK-<version>.el7.centos. The centos version of this package is flagged as an unacceptable dependency during installation.

- **Note:** If errors are received during groupinstall that refer to conflicts with the awips2-python-pillow package, an exclude statement will need to be added to the groupinstall command: `yum groupinstall "AWIPS II Standalone --exclude=awips2-python-pillow`

3.1.4 Install the AWIPS II ADE source

QA ENG

Note: Login as a development user

Note: The ADE requires approximately 4 GB of space.

- _____ Unpack the source jar
 - `cd /home/user/`
 - `/awips2/java/bin/jar -xvf <path-to-ade-directory>/awips2-ade-baseline-SOURCES.jar`

3.1.7 Set Up the AWIPS II ADE

QA ENG

- _____ Backup your site level localization files prior to uninstalling
 - Preserve your current environment by copying /awips2 directory to another directory
 - **Sudo cp -pr /awips2 /awips2_j8p2**
 - Uninstall using yum
 - **Sudo yum remove awips2***
 - **Sudo rm -rf /awips2/***
 - **Sudo yum group mark remove 'AWIPS II *'**
 - Uninstall using ADE: In the directory where you installed the ADE run the following command:
 - **Ade_quick_uninstall.sh**
- Install the 20.3.1 environment
 - Install using yum. Ensure your yum repository is configured to install from a 20.3.1 or later repository
 - Perform the following steps
 - **Sudo yum groupinstall 'AWIPS II Standalone'**
 - **Sudo yum install awips2-localization-OAX**
 - **sudo yum install awips2-eclipse awips2-ant**
 - Install using ADE: In the directory where you installed the ADE run the following command:
 - **Ade_quick_install.sh**
 - If you did not preserve your old environment above, you can skip to Change

ownership and permissions of the awips installation line

- Rename this new environment using the command below
 - **Sudo mv /awips2 /awips2_j11p3 /awips2**
- Create a bind mount to the new environment
 - **Sudo mount -rbind /awips2_j11p3 /awips2**
- Change ownership and permissions of the awips installation
 - **Sudo chown -R \${USER} :fxalpha /awips2**
 - **Sudo chmod -R 775 /awips2/edex/data**
- Edit the init.d scripts and pypies configuration to run the AWIPS processes under your username. In the following steps you can replace vim with the editor of your choice.
 - **Sudo vim /etc/init.d/edex camel**
 - **# locate EDEXUSER=awips and change awips to your username**
 - **Sudo vim /etc/init.d/qpid**
 - **# locate export QPIDUSER=awips and change awips to your username**
 - **Sudo vim /etc/init.d/edex_postgres**
 - **# locate export PGUSER=awips and change awips to your username**
 - **Sudo vim /etc/init.d/httpd-pypies**
 - **# locate nohup su awips and change awips to your username**
 - **Sudo vim /awips2/httpd_pypies/etc/httpd/conf/httpd.conf**
 - **# locate User awips and change awips to your username**
- Copy in your site level localization files from your previous installation as desired
- If you have sufficient disk space it is highly recommended to clone a second set of git repositories into a separate directory from your current set
- In each of your git repositories you will need to perform the following actions
 - **Git stash**
 - **# replace dev with swit_20.3.1 if prior to 20.3.1 merge to dev**
 - **Git checkout dev**
 - **Git pull -rebase**
 - **Git stash pop**
- If you have merge conflicts you will need to resolve them. (See the Java11_Changes.txt file for the kinds of changes you might need to make to be compatible with the new environment.
- For Python 3 updates see the README and accompanying scripts in the py3tools directory.
 - Open Eclipse.
 - Command line
 - **/awips2/eclipse/eclipse -vmargs -xmx3072m -xms512m**
- It is highly recommended that you create a separate Eclipse workspace for the new environment
- Configure Eclipse for AWIPS development

- Select **Window->preferences**
 - Under Java->Installed JREs, **/awips2/java** is selected
- Select Java->Compiler and set Compiler compliance level to 1.8
- Click Apply
- Expand Java->Code Style->Code Templates
- Click Import...
- Navigate to your AWIPS2_Baseline git repository and then to edexOsgi/build.edex/opt/eclipse/ Select codetemplates.xml and click Open
- Click Apply
- Expand Java->CodeStyle->Formatter
- Click Import...
- Navigate to your AWIPS2_Baseline git repository and then to edexOsgi/build.edex/opt/eclipse/ Select formatter.xml and click Open
- Click Apply
- Set your Ant version.
 - In the Preferences dialog, expand **Ant** and select **Runtime**
 - Click the **Ant Home...** button
 - Select **(filesystem) /awips2/ant**
 - Click **OK** (just to close the browse window; if you click the OK to save/close Preferences, you will need to re-open Preferences for the next step)
- Set your Pydev Interpreter.
 - In the Preferences dialog, expand **Pydev**, then expand **Interpreters** and select **Python Interpreter**
 - Click the **New...** button in the **Python interpreters** section (at the top of the dialog).
 - Browse to and select **(filesystem) /awips2/python/bin/python** and click **OK** (to close the Browse dialog).
 - On the **Selection Needed** popup, keep the default selections and click **OK**.
 - Click **OK** (save and close Preferences).
 - If you get a popup asking you if you want to build because your compiler settings have changed, click the **No** button. This will be done a few steps later.
- Disable Build Automatically (temporarily until everything is set up)
 - Select **Project**, and then deselect **Build Automatically**.
- Disable API Errors
 - Select **Window->Preferences**
 - Expand **Plug-in Development**
 - Select **API Baselines**
 - Look for **Missing API baseline:** within the **Options** box
 - Make sure this is set to **Ignore**
- Set up your Eclipse Projects.
 - Select **File->Import**.
 - In the **Import** dialog, select **General->Existing Projects into Workspace**.

- Click **Next**.
- Click the **Browse** button next to **Select root directory** edit box.
- Select **awips** (should be the default directory) and click **OK**.
It should find all the projects for each directory – they should all be selected.
- Click **Finish**.
- Set up the CAVE configuration folders in the eclipse directory.
 - Open a terminal window if not already open
 - **cd /awips2/eclipse** (if not already there)
 - **sudo ln -s /home/user/awips/build/static/common/cave/etc**
(Link the etc directory in your eclipse directory.)
- _____ Compile AWIPS II.
 - Open Eclipse if not already open (with the **/home/user/awips** workspace).
 - Select **Project->Clean**.
 - Make sure the **Clean all projects** radio button, the **Start a build immediately** checkbox, and the **Build the entire workspace** radio button are selected, and click **OK**.
 - This will clean and build all the projects. At the bottom of the eclipse window, the status bar will indicate how far the build is. Wait until it reaches 100% -- the percentage progress will disappear.
 - Select the **Problems** tab at the bottom of the Eclipse window and make sure there are no Errors. Warnings are okay.
 - **Note:** Errors such as:
Project 'com.raytheon.edex' is missing required library: 'client-plugin-airep.jar'

OR
Bundle 'com.facebook.thrift' cannot be resolved from com.raytheon.uf.common.serialization

may be due to a glitch in Eclipse. Try moving the **.classpath** file in **com.raytheon.edex** to a subfolder, rebuild, move it back, then build again.
 - Select **Window->Show View->Navigator**.
 - In the Navigator view, expand **com.raytheon.edex**.
 - Left click on **.classpath** and drag it into the **META-INF** folder (Any folder would work except for one that already contains a .classpath file.)
 - Select **Project->Clean**. Then click **OK** and wait for it to finish building. You will probably get a Build Failed message, but at times you will not get any errors when there should be – because the .classpath file is no longer in the right place.
 - Left click on **.classpath** in the **META-INF** folder and drag it back to the **com.raytheon.edex** folder.
 - Select **Project->Clean**. Then click **OK**, and wait for it to finish

- building.
 - You should no longer see errors in the **Problems** tab, only Warnings.
- **Note:** Errors such as:
 - “A cycle was detected in the build path...”
 - Close plug-in "ohd.hseb.common".
 - Select **Project->Clean**.
 - Open the “ohd.hseb.common” project.
 - Select **Project->Clean**.
- **Note:** Errors of type “C/C++ Problem” can usually be resolved by removing C/C++ code from the Workbench:
 - From the **Package Explorer** window to the left choose all packages with names beginning with “rary” as well as “com.raytheon.uf.nativelib”, right-click and select “Delete.”
 - Select **Project -> Clean** again to clean and rebuild.
 - Alternatively, when importing the project above, the rary and nativelib packages can be deselected prior to importing
- Re-enable Build Automatically.
 - Select **Project**. Then select **Build Automatically**.
- _____ Set up EDEX deploy.
 - In **Eclipse**, make sure the **Java - Navigator** perspective is selected.
 - If the Navigator tab is not available (Package Explorer and Hierarchy are available by default):
 - Make sure the **Java** Perspective is selected.
 - If Java is not selected, click the >> button in the upper right corner and select **Java**.
 - Select **Navigate->Show In->Navigator**.
 - May need to set ownerships and permissions:


```

          O  sudo chown user:fxalpha -R /awips2
            sudo chmod 775 -R /awips2
            sudo chmod 700 -R /awips2/data
          
```
- Expand **build.edex**.
 - Right click on **deploy-install.xml** and select **Run As->2 Ant Build...** (not 1 Ant Build).
- In the Edit configuration and launch dialog:
 - In the **Targets** tab make sure that **main [default]** is selected.
 - Click **Apply** to save any changes.
 - Click **Run** to deploy EDEX.
- If you get a message “/home/<Your Home Directory>/awips/build.edex/deploy-install.xml:556: Execute failed:

java.io.IOException: Cannot run program "chmod": java.io.IOException: error=7, Argument list too long"

- Edit the deploy-install.xml file as follows:
 - ```
<!-- Change permissions on the Linux start file -->
<chmod file="${esbDir}/bin/start.sh" perm="ugo+rx" />
<chmod file="${esbDir}/bin/linux-x86-32/wrapper"
perm="ugo+rx" />
<chmod dir="${esbDir}/data" perm="775"
includes="**/*" verbose="true" type="both"
parallel="false"/>
```
- If you get an error like this:

***Fatal Python error: Interpreter not initialized (version mismatch?)***

Then you need to, after right-clicking **deploy-install.xml** and selecting **Run as 2 Ant Build**, go to the **Main** tab and within the **Arguments** box near the middle of the **Edit Configuration** window, type the following:

***-Darchitecture=x86\_64***

Then stop all processes, clear your project, rebuild your workspace, and try again.

### 3.1.8 How to Run/Debug EDEX in the ADE

QA      ENG

---

- ☐ \_\_\_\_ Deploy EDEX (if you have not already done so, since the last code change/install).
  - Click on the **External Tools** button from the toolbar (the dropdown arrow next to a green play icon with a lock) and select **build.edex deploy-install.xml**.
    - **Note:** The External Tools button will use the last one selected going forward.
  - **Note:** Each time you Deploy EDEX you will need to shut down and restart EDEX; if you do not do this, you will not be able to connect to it for debugging. If you Deploy EDEX without restarting EDEX, you will get the following error:

Launching EDEX has encountered a problem. Failed to connect to remote VM. Connection refused.
- \_\_\_\_ Start the EDEX services.
  - **Note:** Currently, there is no way to execute EDEX from Eclipse; rather, you test run EDEX by starting EDEX in a terminal. The order of startup is critical, however! For the following instructions, you can either open multiple terminals or you can open a single terminal and open multiple tabs within the terminal.

- Start Postgres.
  - Open a terminal.
  - Change directory to the AWIPS II PostgreSQL directory.
  - **cd /awips2/postgresql/bin**
  - Start the development PostgreSQL.
  - **./start\_developer\_postgres.sh**
  - Leave this terminal open. Closing it can leave the process in a bad state. To shut down Postgres, make sure the terminal window is selected and use **Ctrl+C**.
- Start Httpd-Pypies.
  - Open a terminal.
  - Change directory to the AWIPS II Httpd-Pypies logs directory (this is not really a necessary step; however, it will make it easier to verify that httpd has started correctly).
  - **cd /awips2/httpd\_pypies/var/log/httpd**
  - Become root.
  - **sudo su**
  - Edit the httpd.conf file and httpd-pypies script so as to start httpd-pypies as your user
  - Change directory to /awips2/httpd\_pypies/etc/httpd/conf/
  - **cd /awips2/httpd\_pypies/etc/httpd/conf/**
  - Make the following change to **httpd.conf**: (substitute your username for **user**)
    - *User user*
    - *Group awips*
  - Change directory to /etc/init.d/
  - **cd /etc/init.d/**
  - Make the following change to **httpd-pypies**: (substitute your username for **user**)
    - *echo -n "\$Starting logging service:"*
    - *nohup su user -c "\$loggingCmd >*
    - */tmp/pypiesLoggingService.log 2>&1" > /dev/null &*
  - Start Httpd-Pypies as user root
  - **/sbin/service httpd-pypies start** (If /sbin/service does not work, you can also use: **/etc/init.d/httpd-pypies start**)
  - To verify that httpd-pypies started successfully, check the error\_log; if httpd-pypies started successfully, the last line of the error log will be similar to:  
**Apache/2.2.3 (CentOS) configured -- resuming normal operations**
  - Leave this terminal open. To stop Httpd-Pypies, use: **/sbin/service httpd-pypies stop** (or: **/etc/init.d/httpd-pypies stop**)
- Start Qpid.
  - Open a terminal.
  - Edit the qpidd script so as to start Qpid as your user
  - **su -** (switch to the root user)

- **cd /etc/init.d/**
  - Edit the qpidd script as follows: (substitute your username for **user**)
  - `# Who to run QPID as, usually "awips". (NOT "root")`  
`QPIDUSER=user`
  - Start Qpid
  - As root user or with sudo type **service qpidd start**
  - To shut down qpid, type **service qpidd stop** as root user or with sudo
  - Start EDEX.
    - Open a terminal.
    - Edit the edex\_camel start script so as to start EDEX as your user
    - **su -** (switch to the root user)
    - **cd /etc/init.d/**
    - Make the following change to edex\_camel: (substitute your username for **user**)
    - Exit from root user and back to your user by typing **Ctrl+D**
    - Change directory to the AWIPS II EDEX directory.
    - **cd /awips2/edex/bin**
    - Start EDEX ingest in debug (omit the **-b** or type **sudo service edex\_camel start ingest** if you want to run EDEX ingest without debug).
    - **./start.sh -b ingest**
    - It may take a few minutes for EDEX to finish loading. When you get the following text, EDEX is ready:

```

* *
* EDEX ESB is now operational *
* *

```

    - If you receive outOfMemory errors in the edex-ingest.log, you may need to set the value for non-root users within **/etc/security/limits.d/90-nproc.conf** to **200000**.  - Leave this terminal open. Closing it can leave the process in a bad state. To shut down EDEX, make sure the terminal window is selected and use **Ctrl+C**.
  - (Optional) Repeat for **Start EDEX Ingest Grib** and/or **Start EDEX Request**
- **Note:** If you accidentally close the terminal window for any of these processes, stop/kill all leftover processes and restart them in the proper order.
  - For EDEX
    - **ps uax | grep wrapper** then **ps aux | grep java**
    - Use **kill** to kill any PID other than the grep process.
    - **Note:** If you kill java wait a moment and repeat the ps command as the wrapper may automatically restart.
  - For Qpid
    - **ps aux | grep qpidd**
    - Use **kill** to kill any pid other than the grep process



- For Httpd-Pypies
  - **ps aux | grep httpd**
  - Use **kill** to kill any PID other than the grep process.
- For Postgres
  - **ps aux | grep postmaster**
  - Use **kill** to kill any PID other than the grep process.
- \_\_\_\_\_ Start the EDEX Eclipse Debugger.
  - If this IS the first time debugging EDEX:
    - Open Eclipse if not open already.
    - Click **Run->Debug Configurations...**
    - Select **Remote Java Application**, then select the **New launch.configuration** icon in the upper left corner of the window (it looks like a sheet of paper with a plus sign).
    - Set the **Name** to **EDEX**.
    - Make sure the **Project** selected is **com.raytheon.edex.common**.
    - Make sure the **Host** is set to **localhost**.
    - Set the **Port** to **5006** for ingest
      - This process should be repeated to debug other EDEX processes. Use port **5007** for ingestGrib and **5005** for request.
    - Click the **Apply** button.
    - Click the **Debug** button.
  - If this is NOT the first time Debugging EDEX:
    - Click the dropdown arrow next to the Debug button (bug icon) in the toolbar and select **EDEX**.
  - Set a breakpoint in the code, for example, to debug IndexSrv:
    - Set a breakpoint in.  
**com.raytheon.edex.ingestsrv/src/com/raytheon/edex/services/IndexSrv.java** in the **index ()** method (from the Java Perspective).
  - Switch to the Debug Perspective.
    - You may get a popup message asking if you want to switch to the Debug perspective (when starting a Debug session). If you get this message, just click **Yes**.
    - If this is the first time opening the Debug Perspective, click the **Open Perspective** icon (box with a plus sign) in the upper right corner and select **Debug** (with a bug icon) from the dropdown menu.
    - Otherwise, click **Debug** (icon with a bug) in the upper right corner.
  - Start Debugging
    - In order to hit the breakpoint, it needs a file to Ingest. Copy a data file into **/awips2/edex/data/manual**. As soon as the file starts to ingest, the breakpoint will show up in green.
    - **Note:** The data archive as of DR3 is only used by LDM. Therefore, if you do not have an LDM running locally, putting data files in the data archive location (/tmp/sbn, set by the EDEX installer) will not do anything. Also, deploying EDEX will



in the **EDEX Request** output, to show that EDEX is ready:

```

* *
* EDEX ESB is now operational *
* *

```

- Copy the qpid certs to your home directory:
  - `mkdir ~/.qpid`
  - `cp -p /awips2/edex/conf/jms/auth/guest.{key,crt} /awips2/edex/conf/jms/auth/root.crt ~/.qpid/`
  - `chown -R <user>:<group> ~/.qpid`
- \_\_\_\_ If you are using a Remote EDEX Server/standalone box, set up a mount point to the hdf5 directory.
  - *Note:* The EDEX instances, Qpid, Postgres, and Httpd-Pypies must be running on the Remote EDEX Server/Standalone machine you will connect to.
  - `su` – (switch to the root user so you can create a directory at /)
  - `mkdir /test-data` (make a directory to mount to)
  - `chmod user:group /test-data` (change the user and group of the directory to the debugging user)
  - `mount edex-machine:/awips2/edex/opt/data/hdf5 /test-data` (mount awips-dev1:/awips2/edex/opt/data/hdf5 /test-data)
  - *Note:* You will not have to do this every time. You will need to remount if the EDEX-machine is restarted, if EDEX is redeployed to it, or if your ADE machine is rebooted (or you log out).
  - *Note:* If you receive a **Project RPC Error** you may need to run **service nfs restart** as root and try again.
- \_\_\_\_ Start AlertViz in Eclipse.
  - *Note:* As of R1G1-3, by default you must start Standalone AlertViz before starting CAVE. CAVE will no longer start its own AlertViz, unless it is set up specifically to do so.
  - *Note:* Standalone AlertViz will not start unless it can connect to the **EDEX Request** service.
  - If this is the first time running AlertViz:
    - In Eclipse, in the **Java** Perspective, expand the project **com.raytheon.uf.viz.product.alertviz**
    - Right click on **alertviz.product** and select **Run As->1 Eclipse Application**. This will have errors, but will set up most of what is needed in a configuration.
    - As soon as you start to see error/warning/info messages in the **Console** tab or get the **Connectivity Preferences Dialog**, you can stop it by hitting the

- red box icon next to the **Console** tab.
  - Right click on **alertviz.product** again and select **Run As->Run Configurations...**
  - In the **Run Configurations Dialog**, **alertviz.product** should be selected in the right pane under **Eclipse Application**.
  - Select the **Configuration** tab.
  - Make sure the **Use an existing config.ini file as a template** radio button is selected.
  - Click the **Workspace...** button.
  - Expand **com.raytheon.viz.product.awips** and select **config.ini**. Then click the **OK** button.
  - Click the **Apply** button.
  - Click the **Close** button.
- \_\_\_\_\_ Start CAVE in Eclipse.
  - If this is the first time running CAVE:
    - In Eclipse, in the **Java** Perspective, expand the project **com.raytheon.viz.product.awips** (There may also be a **com.raytheon.viz.product.developer – developer.product**)
    - Right click on **awips.product** and select **Run As->1 Eclipse Application**. This will have errors, but will set up most of what is needed in a configuration.
      - If you receive an error like this:  
***Unrecognized option: -Xt***  
***Error: Could not create the Java Virtual Machine.***  
You should right click on **developer.product** instead of **awips.product** and select **Run As->1 Eclipse Application**
    - As soon as you start to see error/warning/info messages in the **Console** tab (after you see the CAVE splash screen) or get the Connectivity Preferences Dialog, you can stop it by hitting the red box icon next to the **Console** tab.
    - Right click on **awips.product** again and select **Debug As->Debug Configurations...**
    - In the **Debug Configurations Dialog**, **awips.product** should be selected in the right pane under **Eclipse Application**.
    - Select the **Configuration** tab.
    - Make sure the **Use an existing config.ini file as a template** radio button is selected.
    - Click the **Workspace...** button.
    - Expand **com.raytheon.viz.product.awips** and select **config.ini**. Then click the **OK** button.
    - Click the **Apply** button.
    - Click the **Close** button.
  - Start the **Debugger**.
    - Right click on **awips.product** and select **Debug As -> Debug**

### Configurations...

- Make sure **awips.product** is selected and click **Debug**.
- Alternatively:
  - Click the dropdown arrow next to the **Debug** button (bug icon) in the toolbar and select **awips.product**.
- Open the **Debug Perspective**:
  - You may get a popup message asking if you want to switch to the Debug perspective (when starting a Debug session). If you get this message, just click **Yes**.
  - If this is the first time opening the Debug Perspective, click the **Open Perspective** icon (box with a plus sign) in the upper right corner and select **Debug** (with a bug icon) from the dropdown menu.
  - Otherwise, in the upper right corner, click **Debug** (icon with a bug).
- **Note:** You should test your changes each time you change CAVE code (using this procedure). Generally, CAVE code is in the project whose name starts with **com.raytheon.viz** (or **com.raytheon.uf.viz**). CAVE does utilize some EDEX code, especially the code in **com.raytheon.edex.common**.
- \_\_\_\_\_ Verify or Set your CAVE Preferences:
  - **Note:** When you first open CAVE (no saved caveData), the Localization may need to be set. You may get a **Connectivity Preferences** popup window with the message: “**Error: Unable to connect to localization server.**”
  - Set the **Localization Server** from:
    - <http://localhost:9581/services> (for a local EDEX)
    - OR
    - <http://edex-server:9581/services> (<http://awips-int1:9581/services>) (for a Remote EDEX)
  - Set Site to **OAX** (the desired Site ID).
  - Click **Validate**. The **Localization Server** text box should change from red to white if CAVE can connect to the server.
  - Click **OK**.
  - **Note:** When you first open CAVE, you may get an error in AlertViz – “Error setting up menu item com.raytheon.uf.viz.core.exception.VizException: Undefined variable ‘id0’”. (This may be due to an incorrect Server Data Directory.)
  - Select **CAVE->Preferences**.
  - Select **Paths**.
    - Verify that the **Share Directory** is set to
      - [/awips2/edex/data/share](#)
  - Select **Hydro Apps**.
    - Verify that the **Database Connection String** is set to
      - `jdbc:postgresql://localhost:5432/hd_ob83oax`

- `?user=awips&password=awips` (for a local EDEX)
    - OR
    - o `jdbc:postgresql://db-machine-name:5432/hd_ob83oax?user=awips&password=awips` (for a Remote EDEX)
  - Select **Localization**.
    - Verify that the **Site** is set to [Site](#) (OAX).
    - Verify that the **Localization Server** is set to
      - o <http://localhost:9581/services> (for a local EDEX)
      - OR
      - o <http://edex-machine:9581/services> (<http://awips-dev1:9581/services>) (for a Remote EDEX).
  - Select **Performance**.
    - Verify that the **Video Card Texture Cache Size** is set to **128MB**.
  - Click the **OK** button.
  - If you made any changes to preferences (such as Localization), you will need to close CAVE (stop debugging) and reopen it (start debugging) using the same method as above. You do not need to restart AlertViz.
  - **Note:** Changes to the RadarServer are the only preferences (at this time) that do not require a restart of CAVE.
  - **Note:** Your preferences are saved in `/home/user/caveData` directory but do not take effect until you restart CAVE. Closing the debugger will not prevent them from being saved.
- \_\_\_\_\_ Start Running/Debugging
  - Put a breakpoint in your code and make the correct selections in the CAVE GUI to hit the breakpoint. Then step through the code.
  - *Example:*
    - Put a breakpoint in the **paintInternal ()** method of **com.raytheon.viz.core/src/com/raytheon/viz/core/topo/TopoResource.java**.
    - To hit the breakpoint, in your CAVE debug session select **Maps->Hires TOPO Image**.
      - o **Note:** If you did not install the TOPO Pack with EDEX (or if you did not set it up properly), you will get the error “error creating tile” (caused by a `FileNotFoundException`).