ATTACHMENT H - TO8 Quick Start Guide

H.1 *Overview*

The quick start guide provides a few procedures for verifying AWIPS II operation. This document assumes that CAVE and EDEX were installed using the procedures outlined in attachments A, B, D and F of the TO8 installation procedure. This document covers checking system services, viewing mule log file and verifying data ingest. The following are the general steps for checking AWIPS II operation:

Check for running EDEX services

Check system log file for exceptions

Check for data ingest

Check CAVE operation

To quickly verify that the system is operational after ingesting data, use the "ps" command to verify that the EDEX services are running (H.2.1.1), use the uEngineWeb application to check for data (H.5.6) and use CAVE to retrieve a few products (H.6).

H.2 *EDEX services*

The status of the EDEX services can be checked using the "ps –ef | grep java" command, the GUI services utility or the service command. The following describes how to verify that the EDEX services edex_activemq, edex_mule, edex_postgres and edex_tomcat are running. The edex_activemq, edex_mule and edex_postgres have to be running for normal AWIPS II operation. The edex_tomcat service is not considered an operational subsystem of AWIPS II, so running the tomcat service is optional.

H.2.1 Using the process command

Verify the java EDEX services are running by typing "ps –ef | grep java" and verify the edex_postgres service is running by type "ps –ef | grep post". The ActiveMQ, Postgres and Mule services should be running for normal operation of the EDEX server (Mule/ESB). The Tomcat service is optional for use with the "uEngineWeb" test web application.

H.2.1.1 Use the "ps –ef | grep post" command

```
[awips@localhost ~]$ ps -ef |grep java
```

The following three sections show the running status of the edex_activemq, edex_mule and edex_tomcat services.

H.2.1.2 ActiveMQ Service

awips 4174 4169 0 06:57 ? 00:00:37 /opt/to8/jdk1.6.0_01/bin/java -Xmx512M -Dorg.apache.activemq.UseDedicatedTaskRunner=true -

Dderby.system.home=/opt/to8/edex/activemq/data -
Dderby.storage.fileSyncTransactionLog=true -Dcom.sun.management.jmxremote -
Djavax.net.ssl.keyStorePassword=password -
Djavax.net.ssl.trustStorePassword=password -
Djavax.net.ssl.keyStore=/opt/to8/edex/activemq/conf/broker.ks -
Djavax.net.ssl.trustStore=/opt/to8/edex/activemq/conf/broker.ts -classpath -
Dactivemq.home=/opt/to8/edex/activemq -Dactivemq.base=/opt/to8/edex/activemq -jar
/opt/to8/edex/activemq/bin/run.jar xbean:file:../conf/activemq.xml

H.2.1.3 Mule Service

awips 5710 5701 0 06:59 ? 00:01:58 java -
Dmule.home=/opt/to8/edex/opt/esb/bin/../../../mule -
Dmule.base=/opt/to8/edex/opt/esb/bin/../../../mule -Dm2.repo="%M2_REPO%" -
Dcom.sun.management.jmxremote -Duser.timezone=GMT -Dedex.dev.mode=off -
Xdebug -Xnoagent -Djava.compiler=NONE -
Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=5005 -Xms512m -
Xmx1500m -Djava.library.path=/opt/to8/lib:/opt/to8/edex/opt/esb/bin/../../../mule/lib/boot -
classpath
/opt/to8/edex/mule/bin/../conf:%MULE_LIB%:/opt/to8/edex/opt/esb/bin/../../../mule/lib/boot
/mule-module-boot.jar:/opt/to8/edex/opt/esb/bin/../../../mule/lib/user/services/junit-

… (Does not show the middle part of the output command listing)

:/opt/to8/edex/opt/esb/bin/../../../mule/lib/user/services/collaboration.jar:/opt/to8/edex/opt/e
sb/bin/../../../mule/lib/user/services/ingestSrv.jar:/opt/to8/edex/opt/esb/bin/../../../mule/lib/u
ser/services/asm-commons-2

H.2.1.4 Tomcat Service (Optional)

awips 5778 1 0 06:59 ? 00:00:12 /opt/to8/jdk1.6.0_01/bin/java -
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -
Djava.util.logging.config.file=/opt/to8/edex/tomcat/conf/logging.properties -
Djava.endorsed.dirs=/opt/to8/edex/tomcat/common/endorsed -classpath
:/opt/to8/edex/tomcat/bin/bootstrap.jar:/opt/to8/edex/tomcat/bin/commons-logging-api.jar -
Dcatalina.base=/opt/to8/edex/tomcat -Dcatalina.home=/opt/to8/edex/tomcat -
Djava.io.tmpdir=/opt/to8/edex/tomcat/temp org.apache.catalina.startup.Bootstrap start

H.2.1.5 Use the "ps –ef | grep post" command

Use the "**`ps –ef | grep post`**" command to check the status of the Postgres database. This is similar to checking the Postgres status in AWIPS. The following the process listing on a test system:

awips 4207 1 0 06:52 ? 00:00:00 /opt/to8/bin/postmaster -D /opt/to8/data -p 5432

awips 5045 4207 0 06:52 ? 00:00:00 postgres: logger process

awips 5160 4207 0 06:52 ? 00:00:00 postgres: writer process

awips 5163 4207 0 06:52 ? 00:00:00 postgres: stats collector process

awips 11334 11332 99 08:57 ? 00:01:14 java -Dmule.home=/opt/to8/edex/opt/esb/bin/../../../mule -Dmule.base=/opt/to8/edex/opt/esb/bin/../../../mule -Dm2.repo="%M2_REPO%" -Dcom.sun.management.jmxremote -Duser.timezone=GMT -Dedex.dev.mode=off -Xdebug -Xnoagent -Djava.compiler=NONE -Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=5005 -Xms512m -Xmx1500m -Djava.library.path=/opt/to8/lib:/opt/to8/edex/opt/esb/bin/../../../mule/lib/boot -classpath /opt/to8/edex/mule/bin/../conf:%MULE_LIB%:/opt/to8/edex/opt/esb/bin/../../../mule/lib/boot/mule-module-boot.jar

… (Omitted the middle part of the java command listing)

/opt/to8/edex/opt/esb/bin/../../../mule/lib/user/services/collaboration.jar:/opt/to8/edex/opt/esb/bin/../../../mule/lib/user/services/ingestSrv.jar:/opt/to8/edex/opt/esb/bin/../../../mule/lib/user/services/asm-commons-2

awips 11460 4207 0 08:58 ? 00:00:00 postgres: pguser fxatext 127.0.0.1(32827) idle

... (Omitted similar fxatext process listings)

awips 11479 4207 0 08:58 ? 00:00:00 postgres: pguser fxatext 127.0.0.1(32846) idle

awips 11485 4207 0 08:58 ? 00:00:00 postgres: awips metadata 127.0.0.1(32847) idle

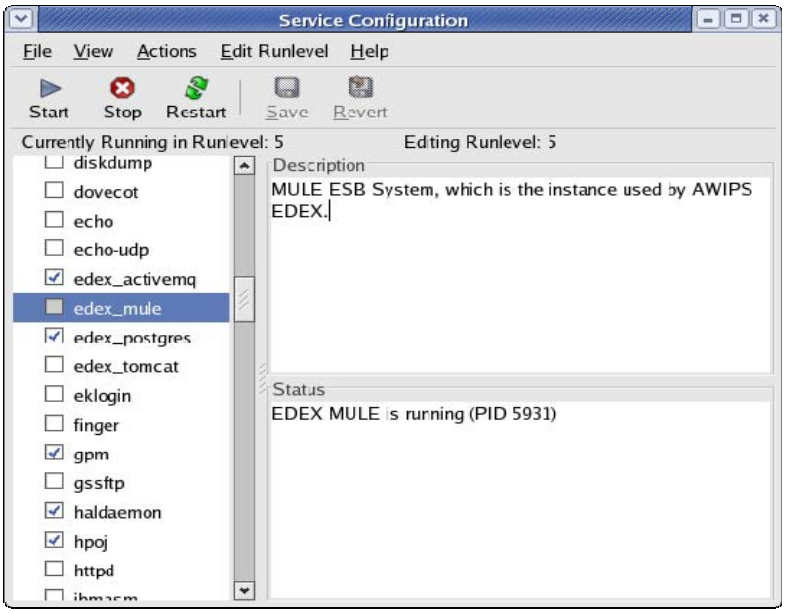
... (Omitted similar metadata process listings)

awips 11504 4207 1 08:58 ? 00:00:00 postgres: awips metadata 127.0.0.1(32866) idle

H.2.2 Using GUI services utility

In Red Hat 4 with KDE, the services utility is found in the following menu *Applications/System Settings/Server Settings/Services*. Root user access is required to use the services utility.

Select a service to view the status; the screenshot below shows the mule service status



H.2.3 Using services status

The following commands can be run as root user to see the status of the EDEX services:

```
service edex_activemq status
service edex_mule status
service edex_postgres status
service edex_tomcat status
```

The following example shows what this looks like for the mule service:

Service edex_mule status

EDEX MULE is running (PID5931)

H.3 *System log file*

The mule log file shows EDEX server status and error messages for the logging level set in the wrapper.conf or log4j.properties file in the mule/conf directory. The logging configuration is beyond the scope of this quick start guide (see AWIPS II FAQ). In TO8 the mule log file shows any data ingest, data decoding, data storage or product notification errors on the server side. Also, the file captures server side errors for CAVE product requests and responses. The following steps will cover were to find the file, what to look for and how to identify the area of the system causing a problem.

H.3.4 Where to find the mule log file

The mule log file is located in the *{edex install path}/edex/mule/logs* directory. The file

name assuming the standalone installation will be "*localhost.localdomain-standalone-yyyymmdd.log*, so it should look something like "*localhost.localdomain-standalone-20080403.log*". The default configuration is for the file to roll over at the start of a new day, so you will want to check that the log files are not growing too large for your system to handle.

H.3.5 What to look for in the mule log file

The mule log file provides a quick look at the current and past system status, so you can track down the general cause of a problem. The log file shows the system startup, data ingest flow, data storage and product requests. Also, finding and viewing exceptions in the log file is a quick way to identify problems.

H.3.5.6 Mule startup sequence

When the EDEX service is started, Mule logs the startup sequence status messages and displays a box indicating that mule started successfully. You should not see any Exception messages in the startup sequence. The following mule log file excerpt shows a normal startup of the EDEX server:

```
Wrapper Started as Console
--------------------------------------------------------------------------
The JVM is being launched with a debugger enabled and could possibly be
suspended.  To avoid unwanted shutdowns, timeouts will be disabled,
removing the ability to detect and restart frozen JVMs.
--------------------------------------------------------------------------
Launching a JVM...
Listening for transport dt_socket at address: 5005
Wrapper (Version 3.2.3) http://wrapper.tanukisoftware.org
Copyright 1999-2006 Tanuki Software, Inc.  All Rights Reserved.

**************************************************************************
* Mule ESB and Integration Platform version 1.4.0                        *
* MuleSource, Inc.                                                       *
* For more information go to http://mule.mulesource.org                  *
*
* Server started: Wednesday, April 2, 2008 11:31:06 AM GMT               *
```

* Server ID: 478be90c-00a8-11dd-a2a3-6335df616497 *

* JDK: 1.6.0_01 (mixed mode) *

* OS: Linux (2.6.9-67.0.7.ELsmp, i386) *

* Host: localhost.localdomain (127.0.0.1) *

*

* Agents Running: *

* Rmi Registry: rmi://localhost:1099 *

* JMX Agent: service:jmx:rmi:///jndi/rmi://localhost:1099/Mule *

* MX4J Http adaptor: http://localhost:9999 *

* Mule Admin: accepting connections on tcp://localhost:60504 *

* Wrapper Manager: Mule PID #6114, Wrapper PID #6112 *

****

INFO 2008-04-02 11:31:29,337 [Awips.Edex.Service.PurgeSrv.1] DbManager: Begin data maintenance routine: Normal Execution.

INFO 2008-04-02 11:31:34,370 [Awips.Edex.Service.PurgeSrv.1] DbManager: Data maintenance routine complete.

INFO 2008-04-02 11:31:34,825 [Awips.Edex.Service.PurgeSrv.1] PurgeSrv: N/A-Used Memory: 32.609909 MB, Free Memory: 474.077591 MB, Heap Size: 506.687500 MB, Available: 1450.765091 MB, Max Memory: 1483.375000 MB

The above shows the messages for a standalone system startup in Mule (edex service).

H.3.5.7 Locating ERROR messages in the file

To identify any errors in the log file, the following steps may be helpful:

In smaller log files, use vi to view and search for exceptions from within vi try the "/ERROR" command to find problems. The first field in the file shows the type of log message, so the default configuration shows INFO, WARN and ERROR messages. By searching for lines beginning with ERROR, the exceptions in the log can be viewed.

In larger log files, use the following command to capture all the exceptions in one file "grep ERROR *log filename* > *errorfile*". Now you can use vi to view any exceptions in the *errorfile*.

H.3.5.8 Understanding ERROR messages

The log file shows ERROR messages and Java Exceptions that may indicate a system

problem. An error message not associated with a Java Exception could be an unimplemented feature or a data issue. When an ERROR message contains a Java Exception, it may be an indication of a system problem, but this is not always the case. The following steps will try to help decipher the error messages, but an understanding of Java and Java Exceptions is required to track the cause of some of the issues.

An ERROR message that does not have a Java Exception message associated with it may indicate a feature not implemented in TO8, so it may not be a real system problem at this time. The following is an example of such a case:

ERROR 2008-04-02 11:33:25,100 [Awips.Edex.Service.IngestSrv-Radar.1] PacketFactory: No class registered for packet ID:17

ERROR 2008-04-02 11:33:25,433 [Awips.Edex.Service.IngestSrv-Radar.1] RadarDecoder: Radar file contains 44 unrecognized symbology packet types.

ERROR 2008-04-02 11:33:25,783 [Awips.Edex.Service.IngestSrv-Radar.1] RadarDecoder: Couldn't properly handle your radar file; take a look at this error: This does not appear to be a symbology block

The above error messages are related and show a radar product that does not appear to be handled in the system. For TO8 this may not be implemented, but in future releases this would most likely be a problem with the data.

The next error message containing a Java Exception shows the caused by clauses of an error message. This message could be an indication of a data problem, but it is most likely an unimplemented feature in TO8. The thing to notice about this message is that the top level shows a generic error message ("unable to decode grib"), but as you follow the caused by statements in the exception message you find a more detailed message ("java.lang.NegativeArraySizeException"). The caused by clauses of the Java Exception usually indicates the area of the problem. Working with Java Exceptions requires understanding the Java Exception mechanism and experience with Java.

ERROR 2008-04-02 11:34:48,853 [Awips.Edex.Service.IngestSrv-grib.3] IngestSrv: cd13b7dd-00a8-11dd-a2a3-6335df616497

com.raytheon.edex.exception.PluginException: Unable to decode grib

at com.raytheon.edex.plugin.PluginDecoderProxy.next(PluginDecoderProxy.java:157)

at com.raytheon.edex.plugin.PluginDecoderProxy.next(PluginDecoderProxy.java:59)

at com.raytheon.edex.services.IngestSrv.process(IngestSrv.java:213)

at com.raytheon.edex.util.AbstractMessageSrv.onCall(AbstractMessageSrv.java:119)

…

Caused by: com.raytheon.edex.exception.DecoderException: Unable to decode grib

at com.raytheon.edex.plugin.grib.GribDecoder.decode(GribDecoder.java:131)

at com.raytheon.edex.plugin.grib.GribDecoder.decode(GribDecoder.java:82)

at com.raytheon.edex.plugin.PluginDecoderProxy.next(PluginDecoderProxy.java:155)

... 15 more

Caused by: java.lang.NegativeArraySizeException

at ucar.grib.grib1.Grib1BinaryDataSection.<init>(Grib1BinaryDataSection.java:163)

at ucar.grib.grib1.Grib1Data.getData(Grib1Data.java:163)

at com.raytheon.edex.plugin.grib.GribDecoder.decodeGrib1(GribDecoder.java:262)

at com.raytheon.edex.plugin.grib.GribDecoder.decode(GribDecoder.java:123)

... 17 more

A case where a java exception may not be an issue is a “duplicate key violates unique constraint” message as seen in the following:

ERROR 2008-04-02 11:34:20,627 [Awips.Edex.Service.IndexSrv.3] JDBCExceptionReporter: Batch entry 0 insert into awips.sfcobs (datauri, timeObs, refHour, dataTime, reportType, wmoHeader, corIndicator, messageData, platformDirection, platformMovement, stationId, geometry, elevation, locationDefined, latitude, longitude, temp, dwpt, humidity, seaTemp, wetBulb, pressure_altimeter, pressure_sealevel, pressure_station, windSpeed, windDirection, totalCloudCover, wx_past_1, wx_past_2, wx_present, wx_report_type, obsId) values (/sfcobs/2008-03-30_02:00:00.0/1003/null/PHET/39.5/-72.8, 2008-03-30 02:00:00.000000 +0000, 2008-03-30 02:00:00.000000 +0000, 2008-03-30 02:00:00.0, 1003, SMVD01 PANC 300207, NULL, BBXX PHET 30021 99395 70728 41698 10216 10050 21051 40290 54000 70000 8/1// 22234 20501 334// 40601 5////, 135, 9.17, PHET, <stream of 21 bytes>, 0, 0, 39.5, -72.8, 278.15, 268.04999999999995, NULL, NULL, NULL, NULL, 102900, NULL, 16.0, 20, 1, 0, 0, 0, NULL, 43143) was aborted. Call getNextException to see the cause.

ERROR 2008-04-02 11:34:20,627 [Awips.Edex.Service.IndexSrv.3] JDBCExceptionReporter: ERROR: duplicate key violates unique constraint "sfcobs_1_datauri_key"

ERROR 2008-04-02 11:34:20,627 [Awips.Edex.Service.IndexSrv.3] AbstractFlushingEventListener: Could not synchronize database state with session

org.hibernate.exception.ConstraintViolationException: Could not execute JDBC batch update

at org.hibernate.exception.SQLStateConverter.convert(SQLStateConverter.java:71)

at org.hibernate.exception.JDBCExceptionHelper.convert(JDBCExceptionHelper.java:43)

at org.hibernate.jdbc.AbstractBatcher.executeBatch(AbstractBatcher.java:249) …

The above error message is most likely caused by a duplicate product already stored in the database, so it may not be an operational issue. The product is most likely available to the user, but pgadmin3 can be used to query for the product in the metadata tables

(see the section on verifying products are in the database).

H.4 *Data ingest*

Checking the data ingest stage of system operation can be done by looking at the log file and by looking at the messages on the queues. The easiest way to check for data ingest is to look at the mule log file, so the following are a few steps for checking data ingest log entries:

In the INFO logging level, the log file captures limited information about specific products. The product header or the message id can be used to trace the data flow through the StagingSrv and IngestSrv stages. The following mule log file excerpt shows the flow of the "FTUS44KMEG" product through the StagingSrv (first stage) and IngestSrv (second stage).

INFO 2008-04-02 11:33:01,023 [Awips.Mule.Service.StagingSrv-taf.1] StagingSrv: 8cf31775-00a8-11dd-a2a3-6335df616497=FTUS44KMEG.30020757.153

...

INFO 2008-04-02 11:33:01,401 [Awips.Edex.Service.IngestSrv-taf.2] IngestSrv: 8cf31775-00a8-11dd-a2a3-6335df616497-Processed file: ../../processing/FTUS44KMEG.30020757.153

INFO 2008-04-02 11:33:01,401 [Awips.Edex.Service.IngestSrv-taf.2] IngestSrv: 8cf31775-00a8-11dd-a2a3-6335df616497-Ingested (1 records) in 0.276s

The third data ingest stage is the IndexSrv, but the product header or message id are not shown in the log file. To verify that the product was stored in the database by the IndexSrv service, query the Postgres metadata table to verify that the product is in the database.

H.5 *Verifying Data Storage*

The IndexSrv service stores the metadata about the ingested products in the Postgres database, and the binary products in the hdf5 repository. A quick way to verify that the products are being stored is to use the "uEngineWeb" web application running in Tomcat. Also, the pgadmin3 application can be used to query the metadata tables to verify that the products are stored in the database. The storage of binary products in the hdf5 repository can be verified using the hdfview application. The following sections cover using Tomcat, pgadmin3 and hdfview to verify that the products are being stored in the repositories.

H.5.6 Tomcat uEngineWeb web application

The uEnigneWeb Java Web Application running in Tomcat provides a quick way to verify that products are being stored in the metadata tables and the hdf5 repository. Tomcat and the uEngineWeb application are installed as part of the EDEX subsystem installation, so the following section shows how to start and use the uEngineWeb application.

During the EDEX installation the edex_tomcat service is created, so this service has to be running for the web application to work. If previous section on checking the EDEX services status did not show tomcat running, then you will need to start tomcat.

As the awips user or non-root user start tomcat as follows:

Change to the {edex install path}/edex/tomcat/bin directory

Execute **`./Catalina.sh start`** to start tomcat

The preferred way to start tomcat is using the edex_tomcat service as follows:

As the root user start the edex_tomcat service

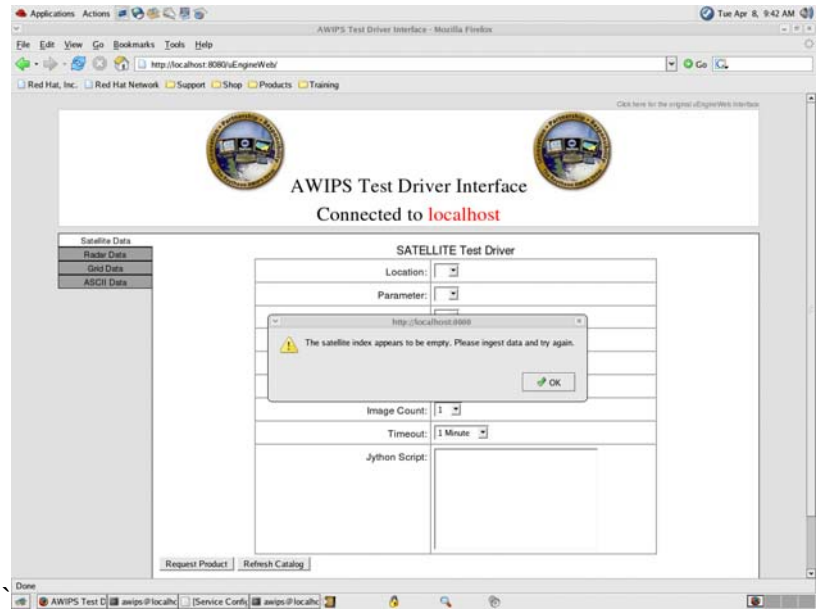
`service edex_tomcat start`

or use the services GUI utility to start the tomcat service

With tomcat running on the local system, open a web browser on the workstation and enter the following URL;

http://localhost:8080/uEngineWeb

If you get the following warning dialog box, then there are not any products for this type of data. So try a different type of data or see the procedure for loading canned data into the EDEX subsystem.



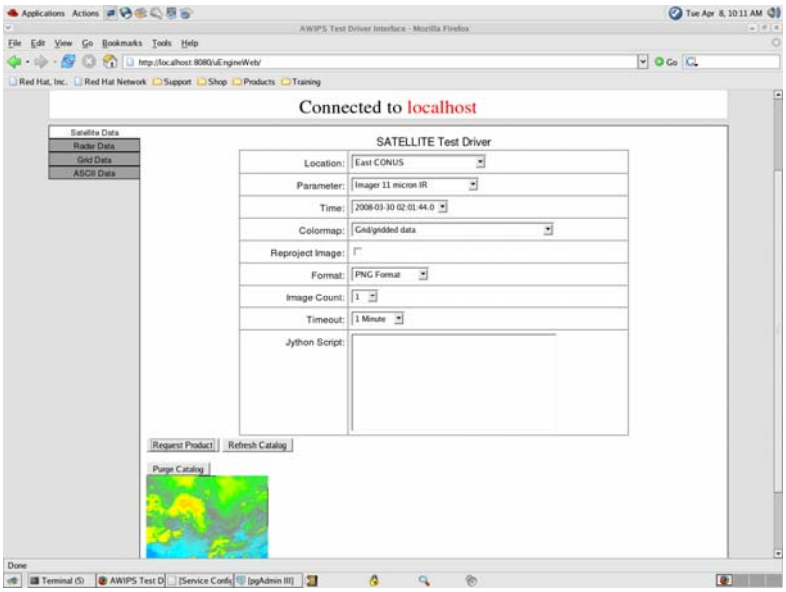
H.5.6.9 To check for satellite products, select the “Satellite Data” tab on the left side of the page (default)

Choose a **Type**, **Location**, and **Time** for the desired satellite product

Click on the **“Request Product**” button

Warning (Do Not click on the “Purge Catalog” button, this clears the database)

The satellite product should display within a few seconds as shown below



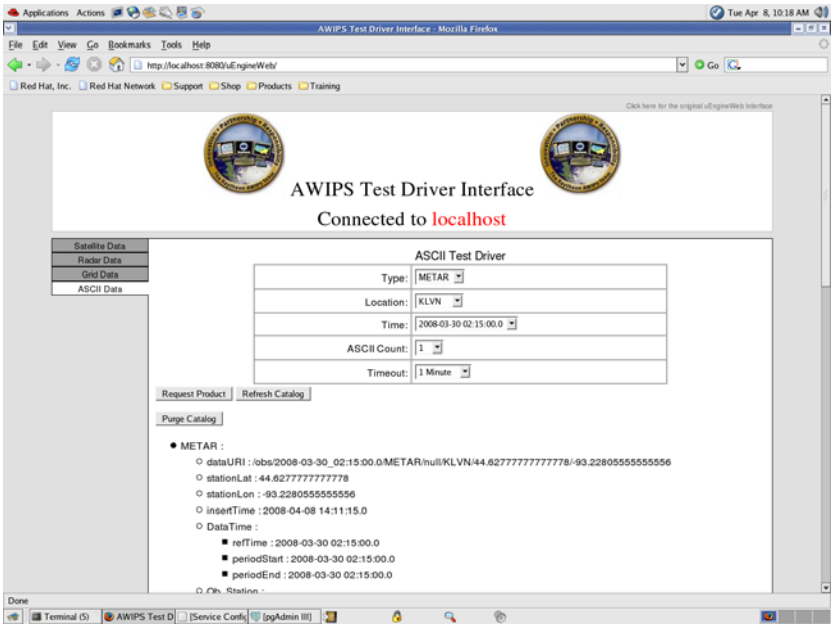
H.5.6.10 To check for ascii products, select the “ASCII Data” tab on the left side of the page

Choose a **Type**, **Location**, and **Time** for the desired text product

Click on the “**Request Product**” button

Warning (Do Not click on the “Purge Catalog” button, this clears the database)

The ascii product should display within a few seconds as shown below



The uEngineWeb application provides a quick way to check on data in the metadata tables and hdf5 repositories. A word of warning about the purge catalog button is necessary, since selecting this option clears the data out of Postgres and HDF5. Also for TO8 there are some questions about how well this feature cleans out the data. So in TO8 **use the Purge Catalog option with caution**.

Also, the uEngineWeb application provides a quick way for local application developers to view data stored in the repositories. The source code for the web application is provided in the ADE, so it may be possible to customize the web pages for your local use in administering the system.

H.6 *Checking CAVE Operation*

A quick way to check CAVE operation is to just retrieve a few products. This procedure shows displaying a satellite, radar and observation product to test CAVE's ability to retrieve data. Also, the procedure displays the Topo background image to verify access to the topo file. This procedure assumes that the data ingest script was run to load data into the EDEX subsystem.

Start CAVE by changing to the {cave install path} directory and execute:

./cave.sh

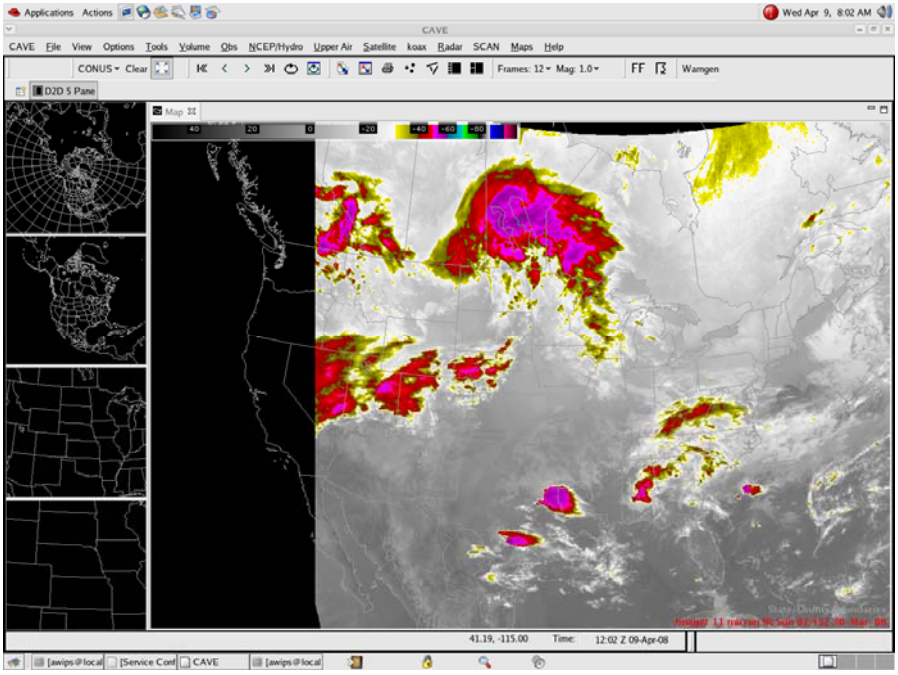
In the D2D window (default view)

H.6.7 Check displaying an East CONUS IR satellite product

Select "**IR Window**" under the "**Satellite**" menu to display the IR Satellite product

The Satellite Image should be displayed in the main window with the product legend.

Note: When the product legend (lower-right hand corner in CAVE) is selected. The popup context menu displays by holding a right-click on the legend for ~1 second.



In the toolbar click on the "**Clear**" button to clear the display

H.6.8 Check displaying an Observation product

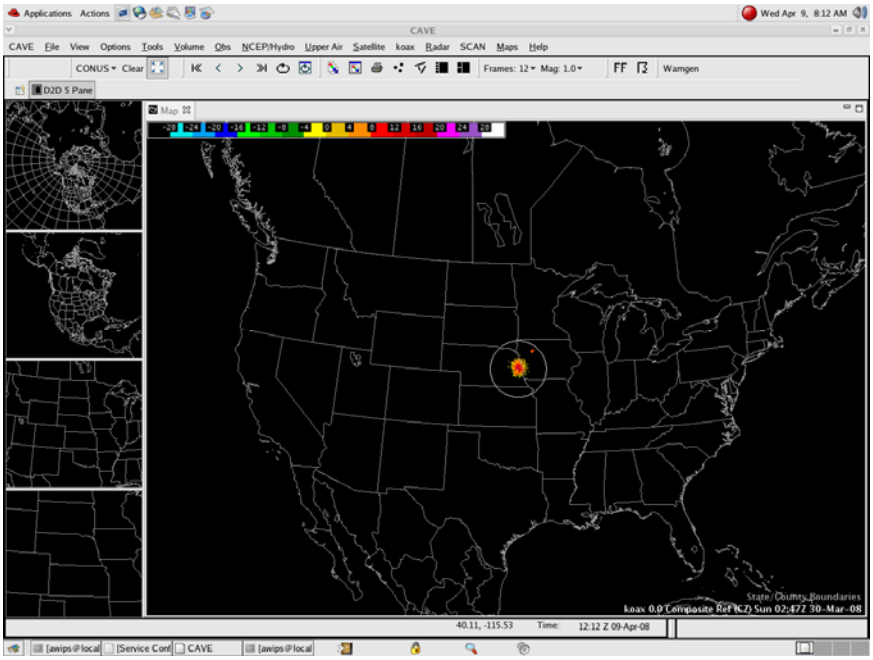
Select "**Surface Plot**" under the "**Obs**" menu to display observations

The Surface Plot should be displayed in the main window with the product legend

In the toolbar click on the "**Clear**" button to clear the display

H.6.9 Check displaying the Comp Ref 4bit (CZ) Radar product

Select "Comp Ref 4bit (CZ)" under the "koax"/"koax 4 Bit Products" menus



In the toolbar click on the "**Clear**" button to clear the display

H.6.10 Check displaying the "HiRes Topo Image" to verify access to the srtm-topo.hdf file

Select the "**HiRes Topo Image**" under the "**Maps"** menu

The Topological background displays in the main window (This may take a moment)

In the toolbar click on the "**Clear**" button to clear the display