

# **Test Case TextDB Command Line Interface**

**for**

**Contract DG133W-05-CQ-1067**

**Advanced Weather Interactive Processing System (AWIPS)  
Operations & Maintenance**

**AWP.TE.SWCTR/TO10-0019**

Prepared for:

U.S. Department of Commerce  
NOAA/NWS Acquisition Management Division  
SSMC2, Room 11220  
1325 East-West Highway  
Silver Spring, MD 20910

Prepared by:

Raytheon Company  
STC Office  
6825 Pine Street  
Omaha, NE 68106

6 February 2009

---

*This document includes data that shall not be duplicated, used, or disclosed – in whole or in part – outside the Government for any purpose other than to the extent provided in contract DG133W-05-CQ-1067. However, the Government shall have the right to duplicate, use, or disclose the data to the extent provided in the contract. This restriction does not limit the Government's right to use information contained in this data if it is obtained from another source without restriction. The data subject to this restriction are contained in all sheets.*

HARD COPY UNCONTROLLED

Submitted By:

\_\_\_\_\_

Test Engineer

\_\_\_\_\_

Date

Approved By:

\_\_\_\_\_

Program Manager

\_\_\_\_\_

Date

\_\_\_\_\_

Mission Assurance Quality

\_\_\_\_\_

Date

## Change History

Revision	Date	Affected Pages	Explanation of Change
Draft	21 Nov. 2008	ALL	Initial Draft
1	10 Jan. 2009	ALL	Result of NWS comments and PDT.
2	6 Feb. 2009	3-7	Result of DT

## Table of Contents

	<i>Page</i>
1.0 SCOPE .....	1
2.0 APPLICABLE DOCUMENTS .....	2
2.1 Source Documents .....	2
2.2 Reference Documents .....	2
3.0 TEST CASE DESCRIPTION.....	3
3.1 Assumptions, Constraints and Preconditions.....	3
3.2 Recommended Hardware.....	3
3.3 Test Inputs.....	3
3.4 Test Outputs.....	3
4.0 TEST SCENARIO .....	4
5.0 REQUIREMENTS VERIFICATION TRACEABILITY MATRIX (RVTM).....	7

## **1.0 SCOPE**

See TO10 Software Test Plan.

## **2.0 APPLICABLE DOCUMENTS**

### **2.1 Source Documents**

- None.

### **2.2 Reference Documents**

- TO10 Software Test Plan for the Advanced Weather Interactive Processing System Project, Contract #DG133W-05-CQ-1067, January 2009.
- The Silver Spring NWS AWIPS I test bed application.
- Rational RequisitePro.

### 3.0 TEST CASE DESCRIPTION

This test case illustrates that the text database command line interface and options work in AWIPS II as it does in the legacy AWIPS I system; reading and writing data to the text database.

#### 3.1 Assumptions, Constraints, and Preconditions

- TO10 software has been installed successfully.
- CAVE, EDEX and pgAdmin III are running.
- Data has been ingested.
- Actions, Results, and Requirements highlighted in gray indicate requirements and/or capabilities to be included in the scope of future task orders. They are included here for purposes of continuity and traceability with the original AWIPS I test case documents.

#### 3.2 Recommended Hardware

See TO10 Software Test Plan.

#### 3.3 Test Inputs

Section 4.0 contains the test procedures for this test case. Sections 2.2 – 2.9 of the TO10 Software Test Plan contain general test inputs applicable to all TO10 test cases. Grayed out test step(s) indicate functionality not yet delivered.

#### 3.4 Test Outputs

The results outlined in section 4.0 are met.

#### 4.0 TEST SCENARIO

Step #	Action	Result	Pass/Fail
1.	Launch a terminal window.	Terminal window is at user prompt.	
2.	ssh root@awips-int1<workstation>[[x1-5]	You are prompted for password.	
3.	Enter password to log in. cd /awips/fxa/bin	You are at the awips-int1 # sign prompt in the directory with the command line interfaces scripts.	
4.	In the terminal window execute the following: ./textdb -r AFOSCmd (Ex. OMAMTROFK)	Read data from the text database.	
5.	Open the PG Admin database tool. Connect to the testbed database and select the fxatext database. Expand the database tree until the tables are displayed	The database tables are displayed.	
6.	Query the database; open an SQL query window and enter an SQL command to retrieve data matching the AFOS command from the step above. select * from stdtextproducts where cccid='OMA' and nnnid='MTR' and xxxid='OFK';	The data displayed matches the data displayed in the first terminal window.	
7.	In the terminal window execute the following: ./textdb -rd AFOSCmd(use OMAMTROFK)	Displays latest products with product count and product lengths.	
8.	In the terminal window execute the following: ./textdb -rw wmoid (ex. SAUS43)	Read data from the text database using all or part of TTAAll.	
9.	In the terminal window execute the following: ./textdb -rs site (ex. KOAX)	Read data from the text database using all or part of CCCC.	
10.	In the terminal window execute the following: ./textdb -rt ddhhmm	Read data from the text database using all or part of date/time group ddhhmm (day, hour, minute).	
11.	In the terminal window execute the following: ./textdb -ri NNNXXX (ex. TEX T09 or 18A A11)	Read data from the text database using all or part of AWIPS ID NNNXXX.	
12.	In the terminal window execute the following: ./textdb -rb bbb (ex. NOR)	Read data from the text database using bbb.	
13.	In the terminal window execute the following: ./textdb -rh HH	Read data from the text database with special headers.	

Step #	Action	Result	Pass/Fail
14.	In the terminal window execute the following: ./textdb -w productID Note: hit return after typing this line, then type the text product. Example: SAUS45 KOAX 062044 This is a test. This is only a test. <ctrl D>	Write to the text database. You should get a message similar to "NORMAL: saved CCCNNNXXX" on the command line.	
15.	In the terminal window execute the following: ./textdb -r (ProductID from step above)	Read data from the text database to show the product added in the step above..	
16.	Query the database. This can be done via SQL with a statement similar to the one in step 6 or refresh the SQL window. select * from stdtextproducts where site like 'KOAX';	The productID in the first terminal window is stored in the database.	
17.	In the terminal window execute the following: ./textdb -t productID {productID...}	Write nicely formatted time of last version(s).	
18.	In the terminal window execute the following: ./textdb -tU productID {productID...}	Write time in seconds (Unix integer) of last version(s).	
19.	In the terminal window execute the following: ./textdb -A productID	Get all times (nicely formatted ) for one productID.	
20.	In the terminal window execute the following: ./textdb -AU productID	Get time in seconds (Unix integer) for one productID.	
21.	In the terminal window execute the following: ./textdb -v productID versions	Change the number of versions to keep in the database.	
22.	Query the database	The number of versions in the database has been changed.	
23.	In the terminal window execute the following: ./textdb -s -a SS XXX CCC; place any state in place of SS(e.g. MD)	Add another ID to the SS.NNN lookup list (state, [SS], XXX, and CCC are character sequences). The tool should respond: NORMAL:Adding a new state-ccc. This does not validate that the SS, XXX, or CCC are valid tokens.	
24.	Query the database. The SQL statement to use is select * from statematch where state='SS' and xxx='XXX' and ccc='CCC'	The addition of the state ID to the database is verified.	

Step #	Action	Result	Pass/Fail
25.	In the terminal window execute the following: ./textdb -s -d SS XXX CCC	Delete an ID from the SS.NNN lookup list (state, [SS], XXX, and CCC are character sequences). The tool should respond "NORMAL:Deleting state-ccc"	
26.	Query the database The SQL statement to use is select * from statematch where state='SS' and xxx='XXX' and ccc='CCC'	The deletion of the state ID to the database is verified.	
27.	In the terminal window execute the following: ./textdb -s -r SS	Display current list for state in SS.NNN lookup list (state is a character sequence).	
28.	Query the database The SQL statement to use is select * from statematch where state='SS'	The listing of the state IDs in the database matches the listing in the first window.	
29.	In the terminal window execute the following: ./textdb -ldad -a productID script	Add the productID and the script path to the watchWarn table for sending to LDAD.	
30.	In the metadata table, query the database (from pgadmin, right-click subscriptions, click refresh, then view data)	The addition of the productID into the database is verified.	
31.	In the terminal window execute the following: ./textdb -ldad -d productID script	Delete the productID and the script path to the watchWarn table for sending to LDAD.	
32.	Query the database (from pgadmin, right-click subscriptions, click refresh, then view data)	The deletion of the productID from the database is verified.	
33.	Exit the database and close the window.	The database is exited and closed.	
34.	Exit to log out and close the terminal window.	The window is closed.	
End of Test			

**5.0 REQUIREMENTS VERIFICATION TRACEABILITY MATRIX (RVTM)**

Number	Description	Test Step(s)
SYSR3124	The AWIPS system shall implement command line Interfaces (TextDB).	ALL