**gfeClient TextProductTest[¶](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/gfeClientGfeTextProductTest" \l "gfeClientTextProductTest" \o "Link to this section)**

Is one of the scripts that can be run through gfeClient. It is an example of how to get a command line interface to the text formatters.

**Syntax**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/gfeClientGfeTextProductTest#Syntax)

{Path}/gfeClient.sh {path}/etc/gfe/itool/TextProductTest.py

If you do that, you should get an SWT GUI created by ProcessVariableList which gives you checkboxes of tests to run. The first time you run it you should do the SetupTextEA option at the bottom.

**Other details**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/gfeClientGfeTextProductTest#Otherdetails)

The TextFormatter in AWIPS II resembles the TextFormatter in AWIPS I as much as possible. The histogram and sampling code that existed in C++ has been ported to Java, the python of the text formatter remains the same with small modifications.

TextProductTest is just a modified version of AWIPS I's. The biggest change to this is that the files are no longer modified by the test script. Instead, the desired changes are passed along to the formatter runner, where they are evaluated at runtime and the module's methods and/or fields are changed on the fly.

There are a few differences between CAVE's text formatter and TextProductTest's. CAVE's is multi-threaded, each formatter tab can run simultaneously on a separate thread which has a brand new interpreter. In contrast, the TextProductTest's python interpreter is the one created by the GFEClient app. Also, CAVE uses the active table which is now a database table in the database EDEX is using. TextProductTest currently just uses a pickled version of the active table like AWIPS I did. Both use the same active table merging code which was directly taken from AWIPS I.

When either runs a product it goes through FormatterRunner.py which is mostly just a port of some of the initial setup steps from TextFormatter.py. The interpreters have references to Java objects like the DataManager, and can use that to get the ParmManager, RefSetManager, etc.

**Notes**[**¶**](https://collaborate.nws.noaa.gov/trac/ncladt/wiki/gfeClientGfeTextProductTest#Notes)

* TimeRange.py, AbsTime.py, and WeatherSubKey.py are there to reduce the number of changes necessary to work with the text formatter. They wrap Java objects to present an interface that matches AWIPS I.
* Many of the python files for the text formatter reside on EDEX for localization and are pulled down to caveData. These include the active table code and the product specific code that comes from configureTextProducts. Changing a python file in caveData will probably be useless as the next time the file is needed, if it differs from the version on EDEX, the version from EDEX will be pulled down to replace the local version.