

# GUnit Architected for Mainframe Dynamic Calling

## (2025-02-03)

---

Files included in this release:

In the copybooks folder:

- **GUNITFLD.cpy** - Copybook containing required fields for interacting with GUnit; This must be installed in a library accessible to the compile and link that creates the dynamically called GUnit modules; Required by GUnit modules (**GUAxxxxx**).
- **TSFIELDS.cpy** - Copybook used by any program that is using GUnit such as *TestDyn.cbl* that test dynamic calls to GUnit; Not required by GUnit modules. See instructions below on how a program can dynamically call the GUnit modules.

In the programs folder:

- **GUAEQNUM.cbl** - Program to handle Assert Equals for Numeric fields. Meant to be dynamically called; Must be compiled and linked to load library available at run time by anyone using it. Required by GUNIT.
- **GUAEQSTR.cbl** - Program to handle Assert Equals for String fields. Meant to be dynamically called; Must be compiled and linked to load library available at run time by anyone using it. Required by GUNIT.
- **GUANENUM.cbl** - Program to handle Assert Not Equals for Numeric fields. Meant to be dynamically called; Must be compiled and linked to load library available at run time by anyone using it. Required by GUNIT.
- **GUANESTR.cbl** - Program to handle Assert Not Equals for String fields. Meant to be dynamically called; Must be compiled and linked to load library available at run time by anyone using it. Required by GUNIT.
- **GUNIT.cbl** - Program to handle initializing GUnit fields to default values. Meant to be dynamically called; Must be compiled and linked to load library available at run time by anyone using it. Required by GUNIT.
- **TestDyn.cbl** - Program to test dynamic call to each of the GUnit modules (**GUAxxxxx**); May be used to verify GUnit modules are available to be dynamically called.
- **GUAxxxxx.dylib** - *FOR GNUCOBOL USE ONLY* - Compiled versions of the GUnit modules that will be dynamically loaded when called by a GUnit Test Suite is compiled and run using the GnuCOBOL **cobc** -x command.

How Test Suites Call GUnit Modules

```
*>*****
*> In the WORKING-STORAGE SECTION of the program
*>*****
      copy 'GUNITFLD'.  <= Fields used to interact with Gunit

*>*****
```

```

*> In the PROCEDURE DIVISION of the program
*>*****

*> Initialize the fields used to interact with GUnit
*> to their default values

    Call GU-InitializeDefaultValues using GUnit-Test-Fields.

*> Set the expected value for the test
    Move some-value to GU-Expected-Value-Numeric.
        --or--
        GU-Expected-Value-String.

*> If you'd like to see the values used in the GUnit Assert
    Set GU-Show-Values to True.

*> Set up data to be used in the test

*> Perform the process to produce a result

*> Set the actual value from the test
    Move value-from-test to GU-Actual-Value-Numeric.
        --or--
        GU-Expected-Value-String.

*> Call the appropriate GUnit module to do the Assert

    Call GU-AssertEquals-Numeric      using GUnit-Test-Fields.
        --or--
    GU-AssertNotEquals-Numeric
        --or--
    GU-AssertEquals-String
        --or--
    GU-AssertNotEquals-String

*> Check to see if test passed or failed using 88-levels supplied with
GUnit

    If GU-Test-Passed
        'processing-desired-when-test-passes'
        --or--
    If GU-Test-Failed
        'processing-desired-when-test-fails'

```

**It is strongly suggested a test suite wrapper for related tests be used as demonstrated in the *Sample-Test-Suite* project**

gnuCOBOL considerations

***If testing with gnuCOBOL use the following command from the folder containing the code to compile and link:***

```
cobc -x -I ../copybooks TestDyn.cbl
```

***and this command to run the test program*** (Note: the GUAxxxxx.dylib files must be in the same folder as the program you are running - see above comment on **.dylib** files):

```
./TestDyn
```