
TESTING SYSTEM USER GUIDE

1.	User guide.....	2
1.1.	Testing Station App	2
1.2.	Tested OBC App	8

1. User guide

1.1. Testing Station App

Configuring the System

The application might need specific configuration in order to run properly, because the IP address we used when communicating with the Tested OBC App can be different between system and operation modes. All the application configuration can be done using TestingStationAppSettings.INI file, found where the application is located. In the configuration file you can chose the communication types, UDP (Not recommended), TCP(Recommended) and Local Simulation. The local simulation is an asynchronous mode running without the Tested OBC App while simulating the full application capabilities and can be used to test the environment. For the UDP you need to place both the Testing Station PC IP in the local Ip address field and the Tested OBC PC IP in the destination IP address filed. For TCP you only need to update the local IP and port.

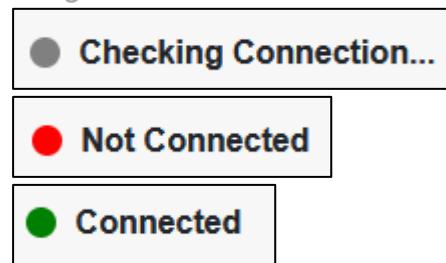
Image 1: Testing Station App configuration. The Source and Destination ports should be the opposite of the Tested OBC App ports.

```
[General]
#The type of the Communication channel. options: LocalSimulation , Tcp , Udp
CommChannelType = Tcp
#IP for any IP related comm channel
LocalIpAddress = 127.0.0.1
DestIpAddress = 127.0.0.1
#Local port
SourcePort = 8889
#destination port
DestinationPort = 8888
```

Top Menu

In the top of our application we have a menu, containing 3 things:
 Tested OBC App Connection indicator:

Image 2: Connection status indicators.



About link, leading to the project book, and User Guide, Leading to a PDF with this guide.

Image 3: Top Menu options.



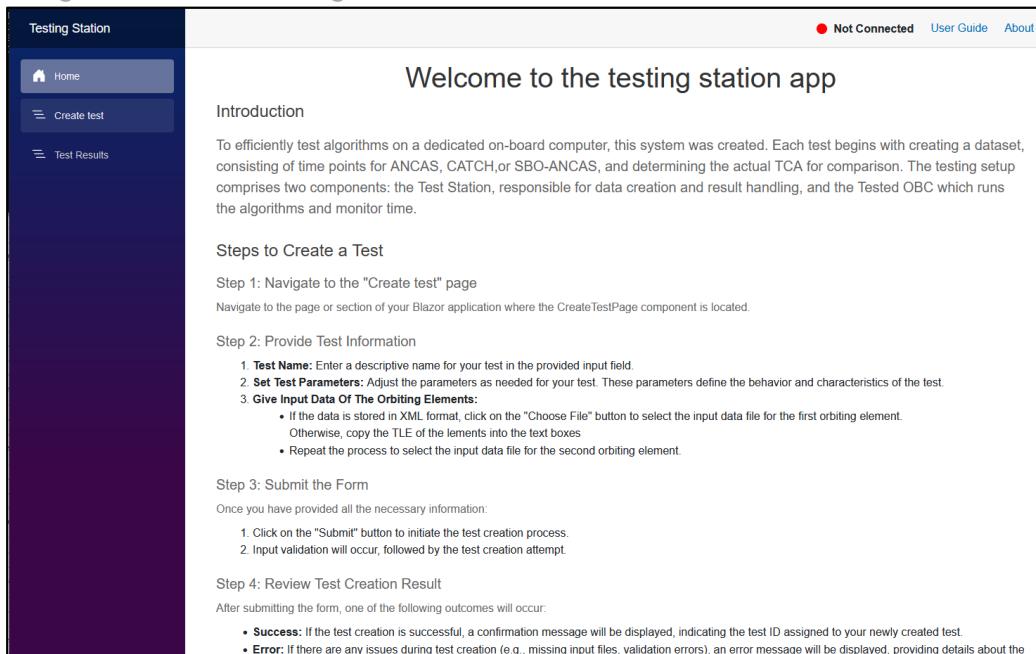
Menu

We have 3 options in the menu, leading to the different pages.

Home Page

The Home Page contain instructions on how to create a new test and view the results.

Image 4: Home Page and the Menu.



Welcome to the testing station app

Introduction

To efficiently test algorithms on a dedicated on-board computer, this system was created. Each test begins with creating a dataset, consisting of time points for ANCAS, CATCH, or SBO-ANCAS, and determining the actual TCA for comparison. The testing setup comprises two components: the Test Station, responsible for data creation and result handling, and the Tested OBC which runs the algorithms and monitor time.

Steps to Create a Test

Step 1: Navigate to the "Create test" page
Navigate to the page or section of your Blazor application where the CreateTestPage component is located.

Step 2: Provide Test Information

1. **Test Name:** Enter a descriptive name for your test in the provided input field.
2. **Set Test Parameters:** Adjust the parameters as needed for your test. These parameters define the behavior and characteristics of the test.
3. **Give Input Data Of The Orbiting Elements:**
 - If the data is stored in XML format, click on the "Choose File" button to select the input data file for the first orbiting element.
Otherwise, copy the TLE of the elements into the text boxes
 - Repeat the process to select the input data file for the second orbiting element.

Step 3: Submit the Form

Once you have provided all the necessary information:

1. Click on the "Submit" button to initiate the test creation process.
2. Input validation will occur, followed by the test creation attempt.

Step 4: Review Test Creation Result

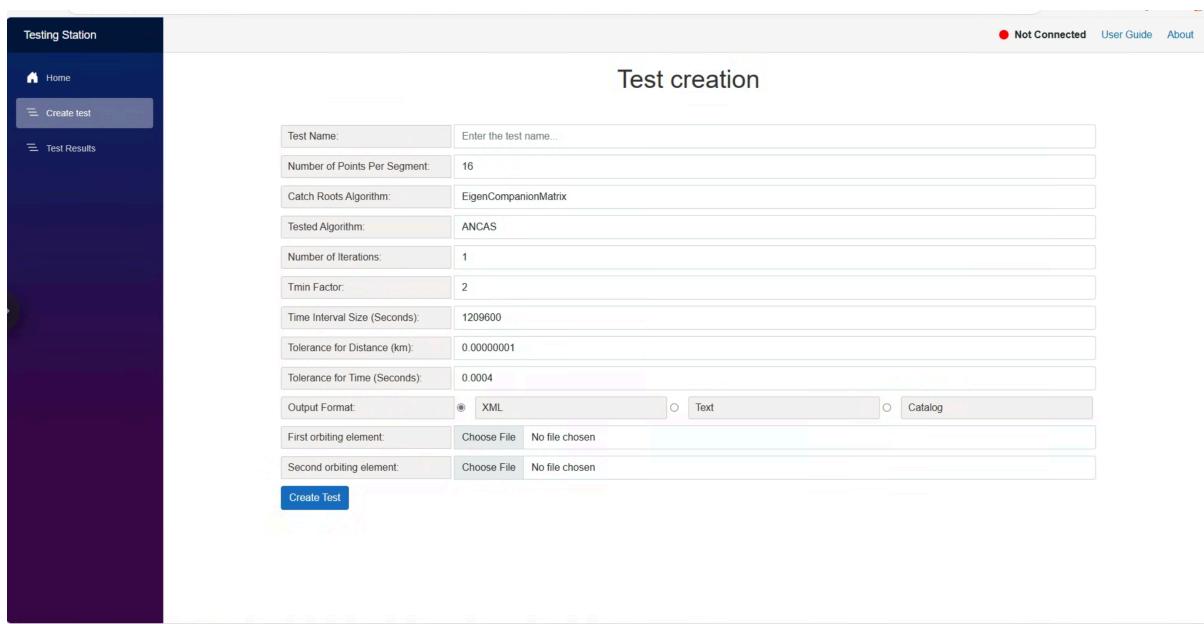
After submitting the form, one of the following outcomes will occur:

- **Success:** If the test creation is successful, a confirmation message will be displayed, indicating the test ID assigned to your newly created test.
- **Error:** If there are any issues during test creation (e.g., missing input files, validation errors), an error message will be displayed, providing details about the error.

Creating a new test

To create a new test, start by navigating to the Create test page:

Image 5: Create tests page.



The screenshot shows the 'Test creation' page of the 'Testing Station' application. The left sidebar has 'Home' and 'Create test' (which is selected) under 'Testing Station'. The main area has a title 'Test creation'. It contains several input fields:

- Test Name: Enter the test name...
- Number of Points Per Segment: 16
- Catch Roots Algorithm: EigenCompanionMatrix
- Tested Algorithm: ANCAS
- Number of Iterations: 1
- Tmin Factor: 2
- Time Interval Size (Seconds): 1209600
- Tolerance for Distance (km): 0.0000001
- Tolerance for Time (Seconds): 0.0004
- Output Format: XML (radio button selected)
- First orbiting element: Choose File (No file chosen)
- Second orbiting element: Choose File (No file chosen)

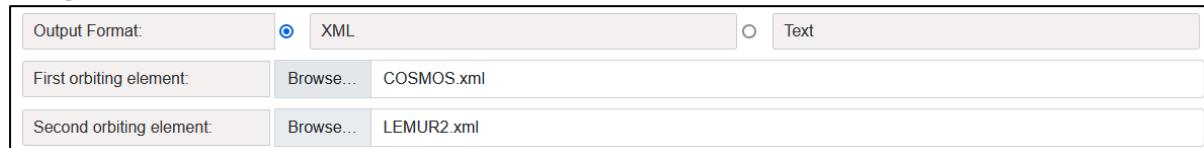
A 'Create Test' button is at the bottom.

The create test form come with some default values, with number of points per segment set to 16, number of iterations to 1, TminFactor to 2, the time interval to 1 week and the tolerances to their default values.

You start by entering the test name, doesn't need to be a unique name. after that choose the desired algorithm and enter the input satellites data.

The first input option is XML files, simply choose 2 files from your file system.

Image 6: Test creation with XML format.

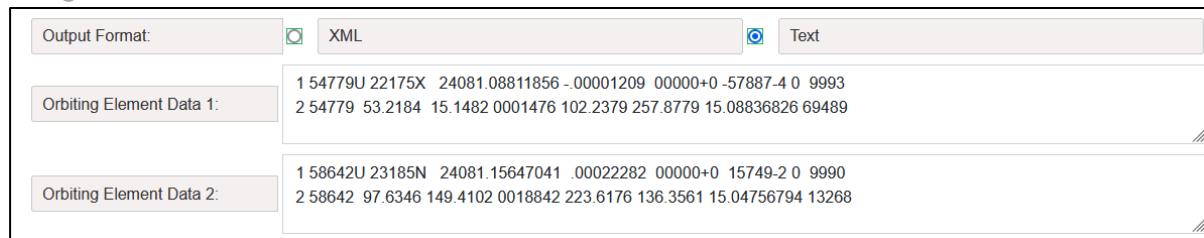


The screenshot shows the 'Output Format' section of the 'Test creation' form. It includes a radio button for 'XML' (selected) and another for 'Text'. Below are two file selection fields:

- First orbiting element: Browse... COSMOS.xml
- Second orbiting element: Browse... LEMUR2.xml

The second input option is TLE, placing the 2 lines of input for each satellite inside the text box as 2 lines.

Image 7: Test creation with the TLE format.



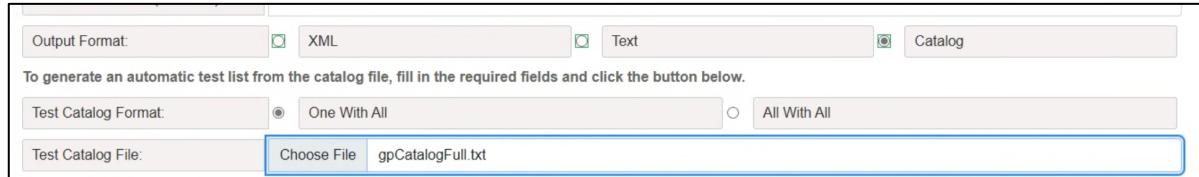
The screenshot shows the 'Orbiting Element Data' section of the 'Test creation' form. It includes two text input fields:

- Orbiting Element Data 1:
1 54779U 22175X 24081.08811856 -0.00001209 00000+0 -57887-4 0 9993
2 54779 53.2184 15.1482 0001476 102.2379 257.8779 15.08836826 69489
- Orbiting Element Data 2:
1 58642U 23185N 24081.15647041 00022282 00000+0 15749-2 0 9990
2 58642 97.6346 149.4102 0018842 223.6176 136.3561 15.04756794 13268

The third input option is a full catalog text file. Simply select a text file from your file system and choose the format for creating tests from the catalog (The maximum size of the file should be less then 512000 bytes):

- One with all - The first object in the file is tested against all other objects in the file.
- All with all-All objects in the file are tested against each other. The default value is One with all.

Image 8: Test creation with the Catalog format.



The screenshot shows a user interface for test creation. At the top, there are four radio button options: XML, Text, Catalog, with Catalog being selected. Below this, a note says: "To generate an automatic test list from the catalog file, fill in the required fields and click the button below." Under "Test Catalog Format", the "One With All" option is selected. In the "Test Catalog File" field, the path "gpCatalogFull.txt" is entered. A blue "Create Test" button is visible at the bottom left.

For XML or TLE format selection, after pressing the "Create Test" button, if your input was correct, the test will be created and you'll get the test ID with successful message.

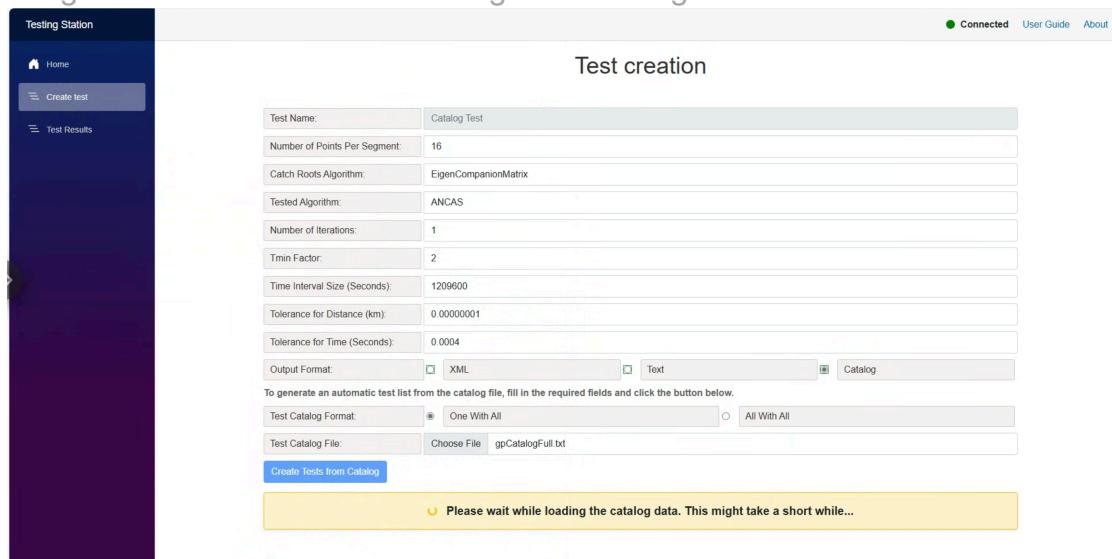
Image 9: Test create success message.



The screenshot shows the same interface as above, but now the "Create Test" button is highlighted in blue. Below it, a green bar displays the message "Test created successfully! Test ID: 2".

For catalog format selection, after pressing the "Create Test List" button, if your input was correct, the system will generate a list of tests based on the user's selection, One with all or All with all. During this process, a message will be displayed informing the user that the operation is in progress and that they must wait. The user will not be able to create additional tests until the test list generation is successfully completed.

Image 10: Load tests list In Progress message



The screenshot shows the interface with the "Create Test List" button pressed. A yellow message box at the bottom center says "Please wait while loading the catalog data. This might take a short while...". The rest of the interface remains the same, with the Catalog format selected and the file path "gpCatalogFull.txt" entered.

Once the entire catalog has been scanned and all tests have been successfully created and passed, the user will see a success message stating "Test List created successfully".

Image 11: Tests list create success message.

The screenshot shows the 'Testing Station' software interface. On the left is a sidebar with 'Testing Station' at the top, followed by 'Home', 'Create test' (which is highlighted in blue), and 'Test Results'. The main area is titled 'Test creation'. It contains several input fields:

Test Name:	Catalog Test
Number of Points Per Segment:	16
Catch Roots Algorithm:	EigenCompanionMatrix
Tested Algorithm:	ANCAS
Number of Iterations:	1
Tmin Factor:	2
Time Interval Size (Seconds):	1209600
Tolerance for Distance (km):	0.0000001
Tolerance for Time (Seconds):	0.0004
Output Format:	<input checked="" type="checkbox"/> XML <input type="checkbox"/> Text <input type="checkbox"/> Catalog

Below these fields is a note: "To generate an automatic test list from the catalog file, fill in the required fields and click the button below."

Underneath the note are two radio buttons: "One With All" (selected) and "All With All".

There is also a field labeled "Test Catalog File:" with the value "Choose File gpCatalogFull.txt".

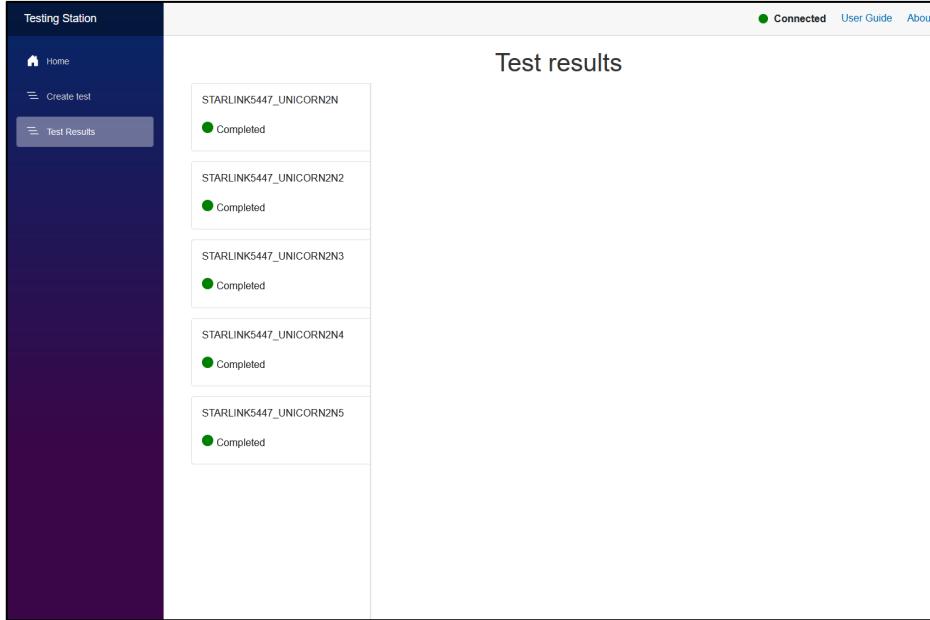
A blue button labeled "Create Tests from Catalog" is located below the file selection field.

A green success message box at the bottom states: "Tests List created successfully!"

Watching the tests results

After creating a test go to the Test results page, when opening you can see a list of the existing tests each with the test name and test status (Completed, In Progress, Failed).

Image 12: The test Results page.



When pressing on one of the tests the test results will be displayed. Almost every value includes tooltip for additional information (formulas, how it was calculated and so on).

Image 13: The test results view.

Test results	
Test Name:	STARLINK5447_UNICORN2N5
Test ID:	5
Test Status:	Completed
Number Of Points Per Segment:	16
Tested Algorithm:	SBO_ANCAS
Number Of Iterations:	1
Tmin Factor:	2
Time Interval Size (Sec):	1209600
TOLdKM:	1E-08
TOLtSec:	0.0004
Initial Number Of Points:	6331
Segment Size (Sec):	2863.1326632267655
Format:	Text
TCA (Sec):	577578.3970222491
Distance Of TCA (KM):	0.13874966805570968
Number Of Points The Algorithm Used:	10534
Average Run Time (Micro):	13959
Minimum Run Time (Micro):	13959
Real TCA (Sec):	577578.4064600052
Real Distance Of TCA (KM):	0.011885297650049195
Distance Of TCA Error (KM):	0.12686437040566048
TCA Error (Sec):	0.009437756147235632

1.2. Tested OBC App

To run the Tested OBC App you only need to set the wanted configuration in the INI file and start the application, found in the released versions folder, the application will continually try to connect to the Testing Station, and when connected will wait for a test request message.

The configuration file contains a few important options you will need to consider.

Image 14: Tested OBC App configuration. The Source and Destination ports should be the opposite of the Testing Station App ports.

```
#The type of the Communication channel. options: LocalSimulation , Tcp
CommChannelType = Tcp
#IP for any IP related comm channel
LocalIpAddress = 127.0.0.1
DestIpAddress = 127.0.0.1
#Local port
SourcePort = 8888
#destination port/server port
DestinationPort = 8889
```

We can decide between three operational modes, Tcp, Local Simulation and Udp (only on windows).

The Local Simulation doesn't require the Testing Station and run a simple tests case when activated, can be used for testing the target system.

The Tcp option used for communicating with the Testing Station. For TCP we only care about the destination Ip address and port, both should be updated in the file with the Testing System Ip and port. The Udp option is only supported on windows.

After configuring the INI file it should be place in the same folder as the Tested OBC.