Getting Started with the Operator Interface

Contents

Introduction

Use the operator interface (OI) to run tests, monitor test status, and save the test results to a local directory or a remote server.

Deploying the Operator Interface in Production

Deploy the operator interface (OI) to run in a production environment, including installing the required packages, connecting the OI to the server, configuring the station/lot settings, etc.

Installing the OI

Procedure

1. Placeholder.

Configuring a Handler/Prober Driver

Procedure

1. In TestStand, select **Semiconductor Module>>Configure Station**.
2. In the **General** tab, select **Enable Handlder/Prober Driver (Real or Simulated)**.
3. Select an appropriate handler/prober driver for your handler or prober from the pull-down menu.

   If you do not have access to a real handler, select **Built-in Simulated Handler Driver** to simulate handler functions. For more information about which driver to select, ask NI engineers for help.

   The following table provides general guidelines about which handler/prober driver to select for your specific handler/prober.

Handler/Prober Type	Handler/Prober Driver to Select
No handler/prober available	**Built-in Simulated Handler Driver**—Use this option for simulating handler functions when you do not have access to a real handler.
Accretech (Tokyo Seimitsu) prober **Examples** • Accretech(TSK) UF3000Exe • Accretech(TSK) FP2000	**Accretech GPIB Prober Driver**
TEL (Tokyo Electron) Prober **Examples** • TEL PrecioXL • TEL P8 • TEL P8XL	**TEL GPIB Prober Driver**
Handler with the GPIB interface bus driver **Examples** • HT-9045/9046 • Zhongyi GPIB Handler	**General GPIB Handler Driver**

Handler/Prober Type	Handler/Prober Driver to Select
Handler with the TTL interface bus driver (For example, Ismeca)	General TTL Handler Driver

4. Select **Configure Handler/Prober**.
5. Specify the settings for the selected handler/prober driver.

   Click the **Help** button at the botton left for more information about each option.
6. Click **OK**.
7. Click **OK**.

Enabling STDF Log for Processing Test Results

Procedure

Note If this configuration is not done, the OI throws an error **Datalog does not exist. Call mechanical engineer and platform test engineer.** after you click the **Save Test Results** button.

1. In TestStand, select **Semiconductor Module>>Configure Station**.
2. In the **Advanced** tab, select **Result Processing**.
3. If the **STDF Log** option is not listed in the **Output Name** column, click the right arrow on the top right and select **STDF Log** from the list.
4. Verify that the **STDF Log** option is listed in the **Output Name** column and place a checkmark in the **Enabled** column.
5. Click **OK** .
6. Click **OK**.

Connecting the OI to Server

You must connect the OI to a test program server and a test data server so that it can download test program from and upload test data to the correct locations.

Procedure

1. Open [*Configuration Dir*]/db/env_config.txt.
2. Update the [server_data_IP_1] variable values to match the test data server you are connecting to. For example, change the *server_data_IP_1* value to the IP address of your server and confirm the customer ID value is correct.
3. To connect to a second test data server for another customer, complete the following steps:

   a. Update the *server_data_FTP_count* value to 2.

      📝 **Note:** If the *server_data_FTP_count* value is 1, the OI saves the test data to the first test data server even if you have configured a second test data server.

   b. Update the [server_data_IP_2] variable values to match the second test data server you are connecting to. For example, change the *server_data_IP_2* value to the IP address of your second server and confirm the customer ID value is correct.
4. Update the [server_pmg_IP] variable values to match the test program server you are connecting to. For example, change the *server_pmg_IP* value to the IP address of your server and confirm the customer ID value is correct.

Operator Interface Environment

The operator interface (OI) has two user interfaces: the main panel and the default operator interface.
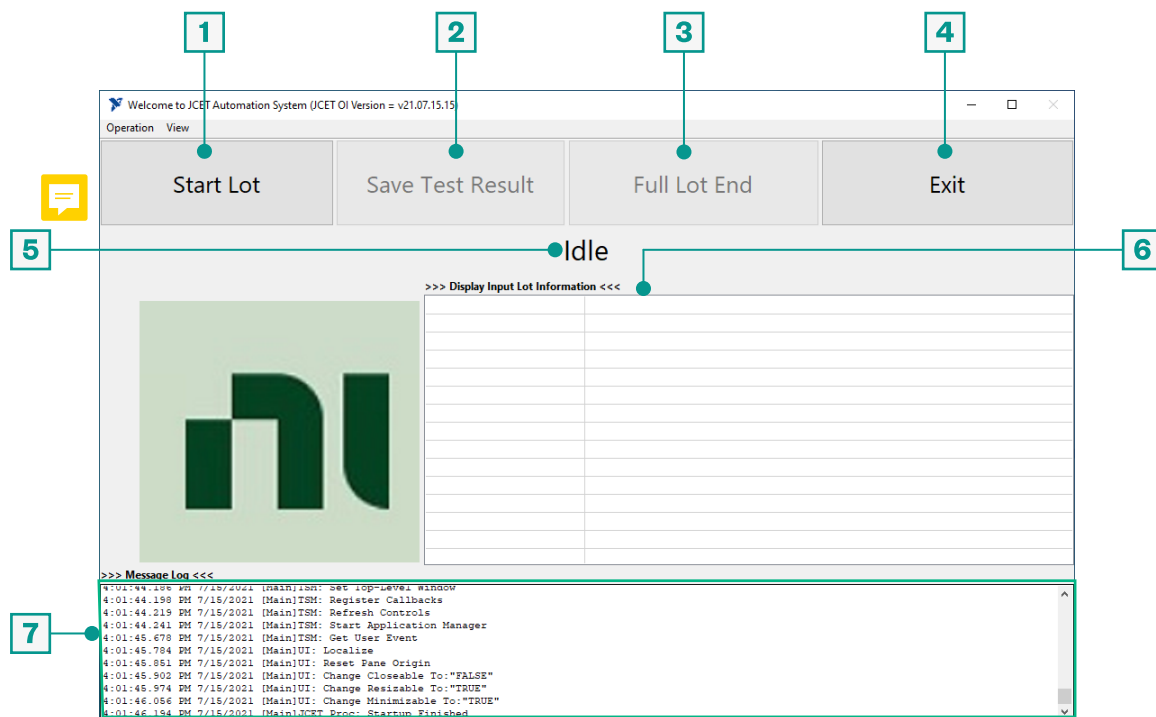
Figure 1: Main Panel

Table 1: Main Panel Key

1 **Initialize Lot**	Initializes the lot by downloading the test program and entering or scanning lot information.
2 **Save Intermediate Data**	Saves the intermediate test data and uploads it to a remote server to the following directories. To modify the locations, contact NI engineers. • `C:\testdata\DLOG\CustomerID\DevName\Customer_LotNo\SubLotNo\csvdir` • `[server_data]\CustomerID\DevName\Customer_LotNo\SubLotNo\csvdir` 📝 **Note:** You can configure the OI to delete the local copy of intermediate test data after uploading it to server by changing the value of the *JCET_csvDir_Cut_flag* variable to 1 in the corresponding recipe file at `[server_pgm_dir]/NI_STS/Product/[Customer_ID]/Recipe/[Zipfile_Name]/[Mode_Code]_recipe.cfg`.

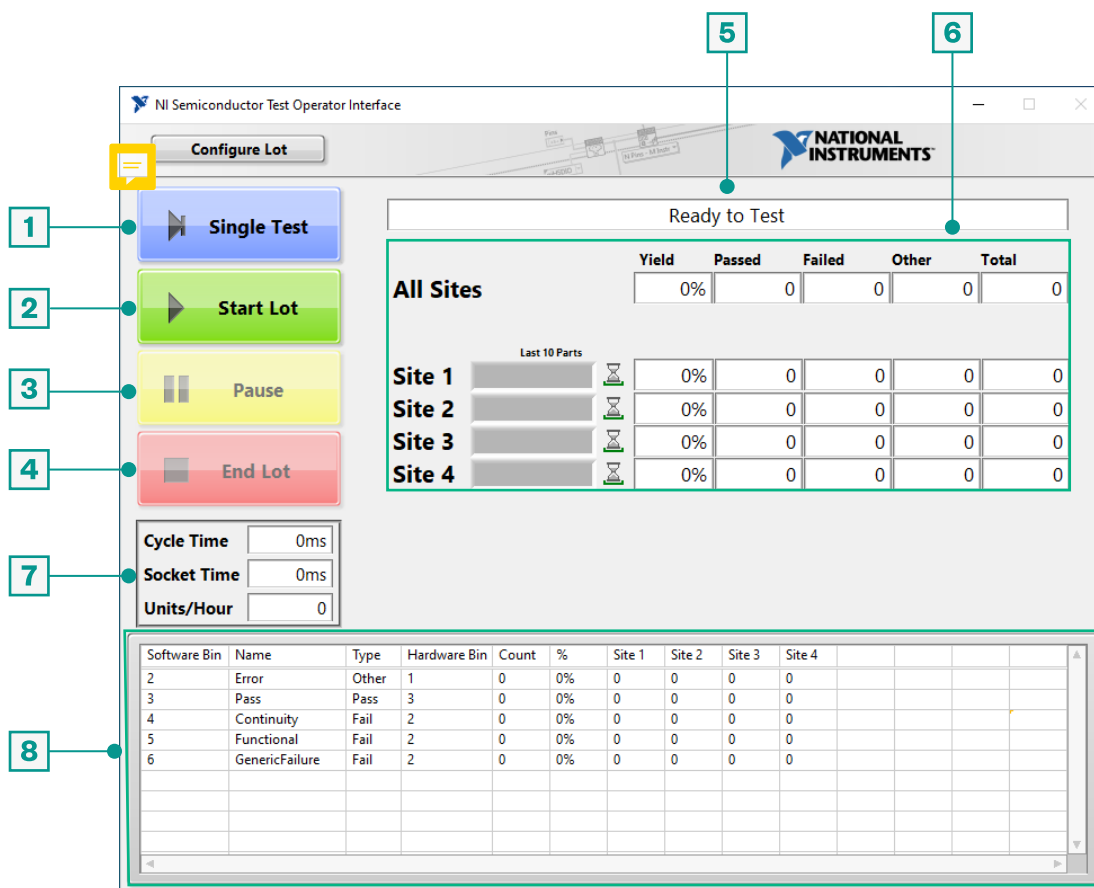|	
3 **Save Summary Report**	Generates and saves summary and transfer reports from the intermediate test data, and uploads the reports to a remote server to the following directories. To modify the locations, contact NI engineers. • `C:\testdata\DLOG\CustomerID` `\DevName\Customer_LotNo\SubLotNo` `\FullLotFile` • `[`*`server_data`*`]\CustomerID` `\DevName\Customer_LotNo\SubLotNo` `\FullLotFile` 📝 **Note:** You can configure the OI to delete the local copy of summary report after uploading it to server by changing the value of the *JCET_FullLotFileDir_Cut_flag* variable to 1 in the corresponding recipe file at `[`*`server_pgm_dir`*`]/` `NI_STS/Product/[`*`Customer_ID`*`]/` `Recipe/[`*`Zipfile_Name`*`]/` `[`*`Mode_Code`*`]_recipe.cfg`.
4 **Exit**	Exits the OI completely.
5 **Status Message**	Displays the OI status in real time.
6 **Lot Information**	Displays basic information of the current lot. The information is either manually entered by the operator or scanned from a barcode.
7 **OI Log**	Displays the OI log in real time, which is helpful for troubleshooting.

Figure 2: Default Operator Interface

Table 2: Default Operator Interface Key

1 **Single Test**	Runs the test program once.
2 **Start Lot**	Runs lot testing, which runs the test program continuously until all DUTs in the lot have been tested.
3 **Pause/Resume**	Pauses or resumes test running.
4 **End Lot**	Stops running the lot.
5 **Test Status Message**	Displays the status of the tester in real time.
6 **Site Execution Data**	Displays statistical and status information for each site in real time.

| 7 | **Statistics Indicator** | Displays statistical information for a lot execution, which can be used for measuring test system performance:

• **Cycle Time**—Socket time plus index time, such as the time required to perform other tasks, such as binning, placing DUTs, and generating reports.
• **Socket Time**—Time it takes to perform test code or time spent by the DUT in the test socket, as measured by the time elapsed between receiving the start-of-test (SOT) notification from the handler or prober and sending the end-of-test (EOT) notification to the handler or prober.
• **Units/Hour**—Number of DUTs tested per hour. |
| 8 | **Bin Table** | Displays information about the binning of DUTs for the current lot, such as the names of the soft bins, the associated hard bins, and the DUT counts for each site. |

Running the Test Program

Procedure

1. Double-click the Semi OI icon on the desktop to launch the operator interface (OI).
2. When prompted, select **Operator** or **MES**, depending on whether you want to run the OI in operator mode for mass production or engineer mode for debugging. The following table lists the differences between the modes.

Mode	Verification Password Required?	Alarm Password Required?	Lot and Program Definition Source
Operator	Yes	Yes	`BarCodeDefinition.txt`
MES	No	Yes	`.mes` file

3. Click **Initialize Lot** on the main panel.
4. When prompted to enter program information, scan from a barcode to automatically pull the information or enter it manually.

   • **(Operator Mode)** Enter the **Customer ID**, then the download information **Program Name** and **Mode Code** as defined in `BarCodeDefinition.txt`, located at `[server_pgm_dir]/NI_STS/Product/[Customer_ID]`.
   • **(MES Mode)** Enter the **Sub LotNo** and **Mode Code** as defined in the `.mes` file, located at `[server_pgm_dir]/MesFile/[Sub_LotNo]/[Sub_LotNo]_[Mode_Code].mes`.

5. Click **Download** to download the test program from the server.

   The OI downloads the test program from the following default directory: `[server_pgm_dir]/NI_STS/Product/[Customer_ID]/[Zipfile_Name]`, where *Zipfile_Name* is the *Zipfile_Name* value defined in `BarCodeDefinition.txt` or `.mes` file, depending on whether if you are running the OI in operator or MES mode.
6. After the download is complete, scan from a barcode to automatically pull lot information or enter lot information in the dialog box that appears.

- **(Operator Mode)** Enter **Operator_ID**, **Device_Name**, etc, as defined in `BarCodeDefinition.txt`.

  The **Tester_ID** field is automatically populated with the computer name, accessible by right-clicking **My Computer** and selecting **Properties**. The other auto-populated fields are pulled from `BarCodeDefinition.txt`.

- **(MES Mode)** Enter **Operator_ID**, **Test_Code**, and **Test_BinNo**.

  The auto-populated fields are pulled from `Mesfile.mes`.

7. Click **OK**.
8. Confirm the lot information and click **OK**.
9. Click **Lot Test** on the popup window.
10. When the **Wait for SOT** dialog box appears, click **Single Test** or **Start Lot** on the default operator interface to run a single test or continuous tests.

    > **Note:** If the test program runs only once after you click **Start Lot**, select **Semiconductor Module>>Configure Station** in TestStand to verify the handler settings.

11. Monitor the test status in the following ways:

    - View the real-time lot test progress in the **Test Status Message** section of the default operator interface.
    - Monitor the real-time status of lot testing by selecting **View>>Monitor Window** on the main panel.

      > **Note:** NI recommends that you use this window for debugging only. For mass production, set the *Real_Time_Monitor_flag* variable in `env_config.txt` to 0 to disable this window.

    - View mid-lot summary by selecting **View>>Mid-Lot Summary** on the main panel.

      > **Note:** In case that the default operator interface hides or disappears, select **View>>STM-OI Window** on the main panel to bring it to the front.

12. Perform the following steps on the default operator interface when necessary.

    - To pause testing at any time, click **Pause**.
    - To resume a paused test, click **Resume**.
    - To stop testing the lot, click **End Lot**.

13. When the lot testing completes, proceed with the following steps on the main panel.

    - To view test reports, select **View>>Report** on the main panel.
    - To save intermediate test data and upload it to the server, click **Save Intermediate Data**.

      > **Note:** By default, the OI keeps the local copy of intermediate test data after uploading it to the server. You can configure the OI to delete the local copy after the upload by changing the value of the *JCET_csvDir_Cut_flag* variable to 1 in the corresponding recipe file at `[server_pgm_dir]/NI_STS/Product/[Customer_ID]/Recipe/[Zipfile_Name]/[Mode_Code]_recipe.cfg`.

      > **Note:** Select **View>>Configuration Window** in the main panel to see where you can find the test data files locally.

    - To start the next lot, click **Initialize Lot** on the main panel and repeat the previous steps of lot testing.

      > **Note:** You must save the test result before starting the next lot. If you do not click **Save Intermediate Data** on the main panel after the lot completes, the **Initialize Lot** button on the main panel is dimmed.

    - To generate summary and transfer reports from the intermediate test data, click **Save Summary Report**.

      > **Note:** By default, the OI keeps the local copy of summary report after uploading it to the server. You can configure the OI to delete the local copy after the upload by changing the value of the *JCET_FullLotFileDir_Cut_flag* variable to 1 in the corresponding recipe file at `[server_pgm_dir]/NI_STS/Product/[Customer_ID]/Recipe/[Zipfile_Name]/[Mode_Code]_recipe.cfg`.

    - To exit the OI completely, click **Exit**.

Test Program and Data Structure

By default, test program and configuration files are placed in the test program server, test data and OI log are placed in the test data server, with predefined folder structures as illustrated below, in which blue indicates folders, orange indicates files, and green indicates packaged zip files. The test data is also saved locally in `C:\testdata\DLOG` with the same folder structures as the test data server.

📝 **Note:** Some of the paths and folder names can be customized in the `env_config.txt` file. To modify the location of the packaged zip files, contact NI engineers.
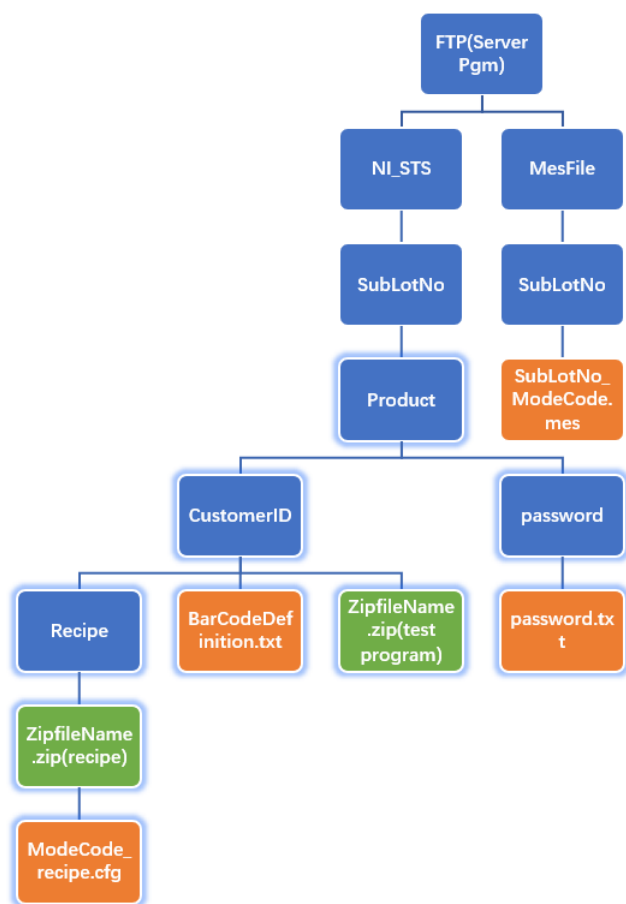

Figure 3: Test Program Structure

📝 **Note:** When packaging files or folders, place them at the top level of the package file rather than a folder with the same name as the package file. For example, use `.../Zipfile_Name.zip/Mode_Code_recipe.cfg` instead of `.../Zipfile_Name.zip/ZipFileName/ModeCode_recipe.cfg`.

Table 3: Test Program Folders and Files Details

Folder/File	File Type(s) Included	Description
`[Sub_LotNo]_[Mode_Code].mes`	Mes file	Defines the lot and program information in MES mode.

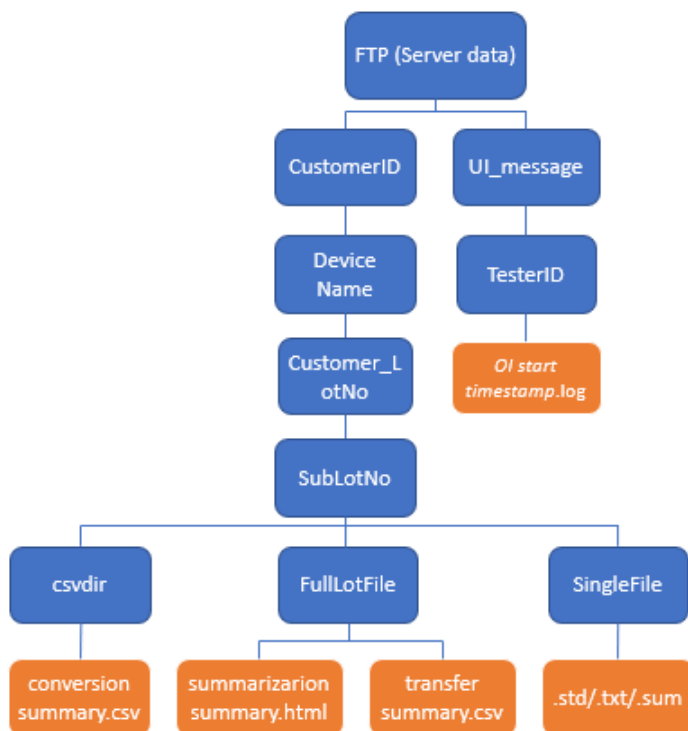| Folder/File	File Type(s) Included	Description
*Mode_Code*_recipe.cfg	Recipe file	Defines the following information: • Test data paths, naming patterns, and compression type • Lot information fields that the user enters or the OI pulls from other configuration files • Test code conversion configuration • STDF attribute configuration • Bin configuration • Configuration options of whether to keep the local copy of test program when starting or exiting lot • Configuration options of whether to keep the local copy of intermediate data or summary report after uploading it to server
BarCodeDefinition.txt	Barcode file	Defines the following information in operator mode: • Program information • Tester OS version • Mode code
Zipfile_Name.zip	Zipped test program	Zipped test program for the OI to download. *Zipfile_Name* is the *Zipfile_Name* value defined in BarCodeDefinition.txt or .mes file, depending on whether you are running the OI in operator or MES mode.
password.txt	Password file	Defines the verification password(s) in operator mode and the alarm password.

Figure 4: Test Data Structure

Table 4: Test Data Folders and Files Details

Folder/File	File Type(s) Included	Description
UIlog	.log	Records detailed operation steps, alarm information, and results. The data is saved in real time.
csvdir	Conversion summary of test data (.csv)	Converted from the tester summary in .csv format, which is generated by clicking **Save Intermediate Data**. It is used as the source for generating the summarization summary (.html).
	[*Mode_Code*]_[*Tester_ID*]_conclusion.txt	Records lot information, start and end time, version information, and conversion summary (.csv) path. It is generated by clicking **Save Intermediate Data**.
FullLotFile	Summarization summary of test data (.html)	Generated by clicking **Save Summary Report** which combines multiple conversion summary (.csv) files into one report according to certain calculation logic and the config information. Use this report to view test results in the production line.

Folder/File	File Type(s) Included	Description
	Transfer summary (`.csv`)	Generated by clicking **Save Summary Report** which combines multiple conversion summary (`.csv`) files into one report according to certain calculation logic and the config information. This report is overwritten each time you click **Save Summary Report**. Use this report to transfer in the production line.
`SingleFile`	`.std, .txt, .sum`	Stores the compressed tester datalog, including tester summary and STDF files generated by the tester.

Configuration Files

The operator interface (OI) uses the following configuration files to define the behavior. You can also customize the behavior by changing some of the configurations.

Configuration File	Description	Default Directory	Can Be Modified?
`env_config.txt`	Defines the following information: • IP address of the program server and data server • Server or local test program path • Server or local test data path • Mes file path • Function call definitions • Alarm flag definitions • Other custom definitions	`[OI program dir]/ db/env_config.txt`	No.

Configuration File	Description	Default Directory	Can Be Modified?
BarCodeDefinition.txt	Defines the following information in operator mode: • Program information • Tester OS version • Mode code 📝 **Note:** If a test program is shared by multiple mode codes, sepatate the mode codes using underscores. For example, FT1_QA1 means FT and QA share a program. To define multiple test programs, put each program in an independent row.	[*server_pgm_dir*]/ NI_STS/Product/ [*Customer_ID*]/ BarCodeDefinition.txt	Yes.
.mes	Defines the lot and program information in MES mode.	[*server_pgm_dir*]/ MesFile/ [*Sub_LotNo*]/ [*Sub_LotNo*]_[*Mode_Code*].mes, where *Sub_LotNo* and *Mode_Code* represent the corresponding values in the .mes file.	Yes. Can be modified in MES mode.

Configuration File	Description	Default Directory	Can Be Modified?
[*Mode_Code*]_recipe.cfg	Defines the following information: • Test data paths, naming patterns, and compression type • Lot information fields that the user enters or the OI pulls from other configuration files • Test code conversion configuration • STDF attribute configuration • Bin configuration • Configuration options of whether to keep the local copy of test program when starting or exiting lot • Configuration options of whether to keep the local copy of intermediate data or summary report after uploading it to server	[*server_pgm_dir*]/ NI_STS/Product/ [*Customer_ID*]/ Recipe/ [*Zipfile_Name*]/ [*Mode_Code*]_recipe.cfg, where *Customer_ID*, *Zipfile_Name*, and *Mode_Code* represent the corresponding values in the BarCodeDefinition.txt or .mes file, depending on whether you run the OI in operator or MES mode. 📝 **Note:** Each mode code has a corresponding recipe file.	Yes.
password.txt	Defines the verification password(s) in operator mode and the alarm password.	[*server_pgm_dir*]/ NI_STS/Product/ password/ password.txt	No.

Troubleshooting

The following are the common errors or issues that you may run into when using the operator interface (OI), and the corresponding solutions.

Test Data Found in Local Folder But Not on Server

First check the UI log message or the UI message file in C:\testdata\DLOG. If the upload is successful, refresh the server folder.

Compression Unsuccessful with Selected Compression Type

First check the notes of the configuration file to determine whether the test program supports this compression type. If it is supported, copy the 7-Zip folder in the 7-Zip installation directory to C drive.
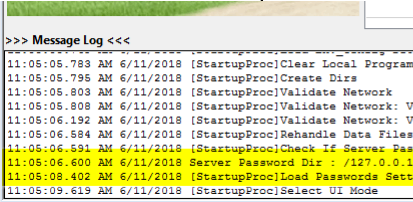
Chinese Characters in OI

If the OI shows Chinese labels for controls and indicators, check the language settings in TestStand by selecting **Configure>>Station Options>>Localization**.

Common Alarm Codes

Error/Alarm Code	Description	Possible Reason	Solution
U01	Cannot connect to server. Call mechanical engineer.	The server is not configured correctly.	Verify that the configuration for the test program and test data servers is correct in `[OI exe dir]/db/env_config.txt`. You can verify whether the configuration is correct by entering the IP address with the following format in the address bar in Windows Explorer: `ftp://username:password@ip:port`. For example, `ftp://user:password@127.0.0.1:21`.
U02	Datalog does not exist. Call mechanical engineer and platform test engineer.	STDF log is not enabled for processing test results.	Follow the instructions in Enabling STDF Log for Processing Test Results to enable STDF log.
U03	Cannot connect to server. Call mechanical engineer.		
U05	Summary does not exist. Call manufacturing leader.		
U06	Cannot connect to server. Call mechanical engineer.		
7zipNotInstalled	7zip is not installed. Install 7zip application and config path option in environment.		
O01	Password failed.		
U04	Config does not exist. Call manufacturing leader.		

Error/Alarm Code	Description	Possible Reason	Solution
M01	Mesfile does not exist in server. Call IT.	• Program or configuration file is placed in the wrong location in the program server • FTP permission issues.	• Verify that the program or configuration file is placed in the correct location in the program server. The OI log displays the server location where the program attempts to read the test program or configuration file. The following image shows an example path of `password.txt` that the OI attempts to read from. • If the path is found on the server but the test program or configuration file does exist, check FTP permissions.
M02	Program does not exist in server. Call IT.		
O02	Barcodefile does not exist in server. Call IT.		
O05	Program does not exist in server. Call platform test engineer.		
M03	Mesfile modecode check failed. Call IT.		
M04	Mesfile program definition error. Call platform test engineer.		
M05	Current program does not match last program. Call platform test engineer.		
M06	Recipe file does not exist. Call platform test engineer.		
M9	Tester_ID check failed. Call manufacturing leader.		
M10	Lot information check failed. Reinput values.		
M11	Testcode input failed. Reinput values.		
M12	Device_name does not match. Call mechanical engineer setup.		

Error/Alarm Code	Description	Possible Reason	Solution
M13	Tester software version check failed. Call plaform test engineer.		
M14	SubLotNo check failed. Called IT.		
M15	Test program file does not exist. Call platform test engineer.		
O03	Program name does not exist in barcodefile. Call platform test engineer.		
O04	More than one program name exists in barcodefile. Call platform test engineer.		
O06	Barcodefile modecode check failed. Call manufacturing leader.		
O07	Barcodefile program-defined error. Call platform test engineer.		
O08	Current program does not match last program.		
O09	Recipe file does not exist. Call platform test engineer.		
O10	Tester ID check failed. Call manufacturing leader.		
O11	Lot information check failed. Reinput values.		
O12	Testcode input failed. Reinput values.		
O13	Device name does not match. Call mechanical engineer setup.		
O14	Tester software version check failed. Call platform test engineer.		
O15	Input2&input3 program name failed. Reinput values.		
O16	Test program file does not exist. Call platform test engineer.		
550		Permission issues of file or folder in FTP server or local machine.	Check permission of file or folder in FTP server or local machine.

Error/Alarm Code	Description	Possible Reason	Solution
-15425	Failed to establish FTP connection.	• The server is not configured correctly. • Firewall or anti-virus software may cause FTP connection failure.	• Verify that the configuration for the test program and test data servers is correct in [*OI exe dir*]/db/env_config.txt. You can verify whether the configuration is correct by entering the IP address with the following format in the address bar in Windows Explorer: ftp://username:password@ip:port. For example, ftp://user:password@127.0.0.1:21. • Try turning off firewall or anti-virus software.
-17500	Operation Failed. Location: Step 'UUT Report' of Sequence 'Module Plugin – UUT Done' in 'xxxx.seq'. Test Socket 0.	This error is from NI's Model Plug-in. The features customized by some manufacturers, such as customized reports, are implemented in the Model Plug-in.	Contact NI engineers.
-1074118575	Location: Step 'XXX' of sequence 'MainSequence(or ProcessSetup or ProcessCleanup)' in 'YYY.seq'. Test Socket 0. User-defined error code. Site: 0. niDCPower/niDigital/RFmx/niRFSG...	Test program error.	Contact NI engineers.
-1073807304	Location: Step 'Wait for SOT' of sequence **StartOfTest** in "Seiko Epson_160831.seq" User-defined error code. Site: N/A.	This error comes from Seiko Epson_XXX.seq, an NI handler driver program.	Contact NI engineers.

If the error is not listed above or if the listed solution does not work, refer to OperatorInterfaceErrors.log in the OI program directory for detailed error messages that occurred when the OI ran, including errors that occurred in OI, handler driver, and test program. To seek for NI assistance, include the OperatorInterfaceErrors.log file and OI log to help NI engineers locate errors.