

Keysight Technologies
Shopfloor Operation of the
Keysight i1000 In-Circuit Test Software

Application Note



Introduction

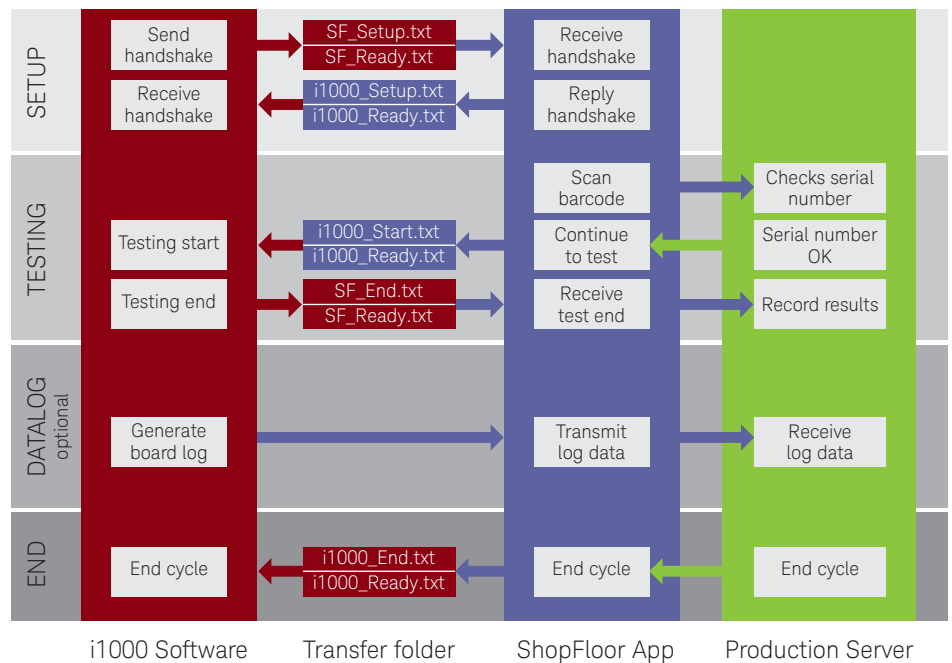
As different users have different setups for their shopfloor operations, it is impossible to have a generic application that will be able to support all kinds of shopfloor server communications. Therefore, shopfloor operations require a standalone shopfloor client application to handle the communication between the i1000 software and the shopfloor server. This allows the i1000 software to operate more efficiently and software revisions are much manageable. The i1000 software offers a simple method of information transfer between itself and the external shopfloor client application. Instead of a complicated software to software communications using DLLs, Command sets, etc. which may be difficult for the user to learn and comprehend, the software uses a set of files with a pre-defined format as the medium of information transfer. Although this may seem outdated and old fashioned, it is easy to understand and effective. This allows the user to get the shopfloor client application up and running faster, without the need to spend too much time working on the communication code itself.

Shopfloor File Transfer

Information exchange between the i1000 and the shopfloor application is accomplished by exchanging files. Information such as board serial number, board selection, operator ID, etc. are embedded in these files and passed between the software. There are four main stages in a typical shopfloor operation cycle:

1. SETUP Stage
2. TESTING Stage
3. DATALOG Stage (Optional)
4. END Stage

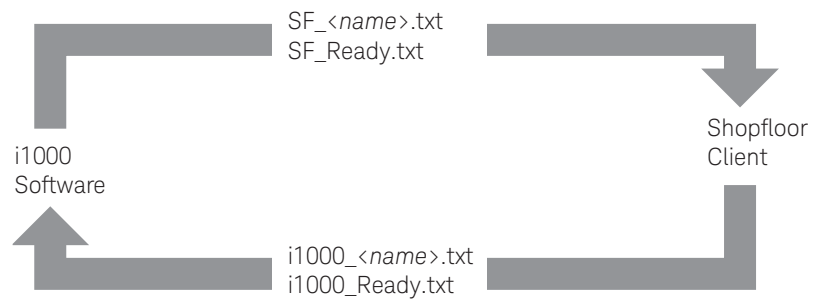
At every stage, information is written into text files and saved into the transfer folder where the shopfloor client software resides. This action is done by both the i1000 software and the shopfloor client. In order to prevent premature reading of files while the other software is still in the process of writing, the communication uses a 'two files' transfer method.



Typical Shopfloor File transfer operations of a test cycle

Two Files Transfer Method

The Transfer file that contains the information is written first. Once the data is written, the software closes that file and creates another empty file. This file is called the READY file. The READY file is used to signal to the other software that the Transfer file is already being created and is ready to be read. The software constantly monitors the transfer folder for the Ready file. Once it detects its existence, it means that there are Transfer files available for import. The software then proceeds to import the transfer files and renames both the Transfer and READY files with an .OLD extension. Alternatively, the Shopfloor client software may choose to delete the used files all together. Transfer and Ready filenames are prefixed with the system name. For the files that i1000 generates to the Shopfloor, the filenames starts with “SF,” which signifies “Shopfloor.” Similarly for files that the Shopfloor client generates to the i1000, the filename starts with “i1000.” This applies for both the Transfer and the Ready files.



Ready Files and Transfer Files

Ready files are empty text files. They do not contain any information and are zero byte in size. The software does not need to read in this file at all. All it needs to do is to monitor its existence. Transfer files are text based and contain different useful information which need to be imported and processed by the software.

Generated by i1000	Generated by Shopfloor Client
SF_Setup.txt	i1000_Setup.txt
SF_Start.txt	i1000_Start.txt
SF_End.txt	i1000_End.txt
SF_Ready.txt	i1000_Ready.txt
SF_Close.txt	i1000_Abort.txt

Table 1. List of typical files used in transfer. Details can be found in the i1000 Online User documentation

Table 1 shows only a list of typical files generated and expected by the i1000 software. In the actual operation, there are several other files generated together with these files. These additional files include:

Additional files generated by i1000
SF_HideSetup.txt
SF_Show.txt
SF_Repeat.txt
SF_Report.txt
ExtendedReport.txt

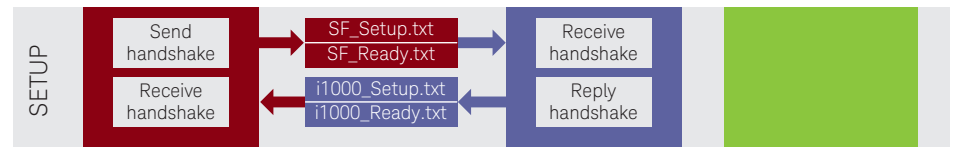
Table 2. List of additional files generated by i1000 software during the shopfloor operation cycle. These additional files are generated to support certain customized options by users. Typical operations of the shopfloor cycle do not require any actions on them.

SETUP Stage

When the Shopfloor client is assigned to the ATD in the i1000 software, the i1000 software will commence the SETUP stage and launch the Shopfloor client. The stage starts with the i1000 software sending SF_Setup.txt followed by SF_Ready.txt into the transfer folder where the Shopfloor client resides.

The Shopfloor client starts up and checks the transfer folder for SF_Ready.txt. Once the file is detected, the Shopfloor client proceeds to check which transfer file is present. It sees SF_Setup.txt and imports it. In response, the Shopfloor client sends i1000_Setup.txt and i1000_Ready.txt back into the transfer folder. At this time, there is no information required by the i1000 from the Shopfloor client; therefore, both the i1000_Setup.txt and i1000_Ready.txt are empty files. I1000 software then detects the presence of i1000_Ready.txt and reads in the i1000_Setup.txt file. The Shopfloor link (SFCS) is now established.

Files that are imported by the i1000 software will get a change to their filename extension so that it will be mistakenly imported in the next stage. Similarly, the Shopfloor client will also need to rename or remove the files that it has processed.

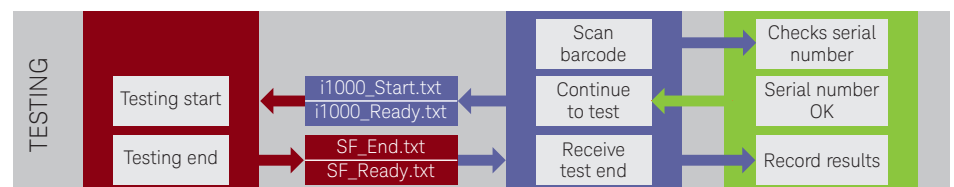


TESTING Stage

With the SFCS link established, the i1000 software now waits for the input from the Shopfloor client. At this time, Shopfloor client may prompt the user to scan the Barcode of the DUT or input other parameters. It then processes the information through the Shopfloor Server and obtains the necessary information and authorization.

The i1000 software may be setup to enforce the transfer of serial numbers from the Shopfloor client. Testing will not be allowed to start if serial number is not received from the Shopfloor Client. This is done in the i1000_SF.INI file.

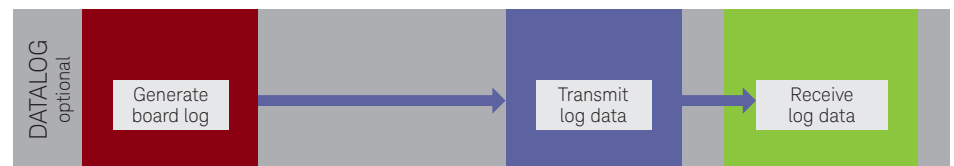
Shopfloor client writes the DUT Serial number into the i1000_Start.txt and sends it to the transfer folder followed by the i1000_Ready.txt file. I1000 software imports the i1000_Start.txt and populates the serial number into its interface and gets ready for the start of test. Start of test is triggered either manually with the operator clicking the test button on the interface, or automatically by the press sensor when it detects that DUT is engaged.



DATALOG Stage (Optional)

The Datalog stage is where the Shopfloor client extracts whatever log files and information after the test cycle. There is no physical file transfer. At this time, the i1000 software had completed all tests and had generated all results and log files. The results and log files are already generated in their respective path as defined in the user's test program. Therefore, Shopfloor client can simply pick up the required files and process them accordingly against the Shopfloor Server

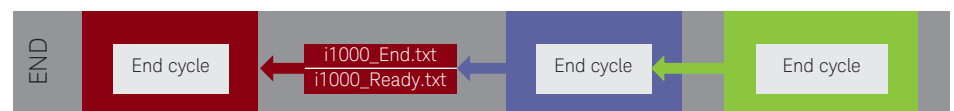
If there is no datalog processing required, then the cycle simply moves straight into the END stage.



END Stage

At the end of test, i1000 software sends SF_End.txt to signal the end of the test cycle. Again, the Shopfloor client detects the SF_Ready.txt file and checks for the SF_End.txt file. SF_End.txt contains information of the test results. Other test information is recorded in the respective directories of the test program by the i1000 software and may be accessed by the Shopfloor client directly. Shopfloor client may then update the test results to the Shopfloor Server.

Shopfloor client will create i1000_End.txt followed by i1000_Ready.txt into the transfer folder. Once the i1000 software detects the file, it renames the file extension to "OLD" and then proceeds to get ready for the next test cycle where it will wait for the shopfloor client to send the i1000_Start.txt file.



Exception Cases

Software closure

When the i1000D software closes, it will attempt to notify the Shopfloor client. This is done by transferring the SF_Close.txt file. Therefore, the shopfloor client application can monitor the existence of this file and terminate itself accordingly.

Test abortion

Shopfloor client may force an Abort on the i1000 software during a test cycle by sending an empty file named i1000_Abort.txt followed by the ready file. The i1000 software will abort testing and return to wait for the next start file.

Types of Transfer Files

SF_Setup

This file is sent from the i1000 software to the Shopfloor client. SF_Setup is generated by the i1000 software every time the software is launched or when a new test program (ATD) is being loaded to the software. It is used to transfer information about the test systems for the shopfloor client to process with the server. Information like Test system ID, Operator ID, Test program name, panel DUT selection, etc. are some of the examples. The file is divided into 2 sections:

Parameter Section

The first section is the Parameter section that contains the board setup of the test program. This tells the Shopfloor client what is the total number of DUTs in the test program and which DUT in the test program is enabled or disabled.

The Parameter section is contained within the header “{>>>>Parameter Area>>>>}” and footer “{<<<<Parameter Area<<<<}.” The parameters entry is separated by “|”.

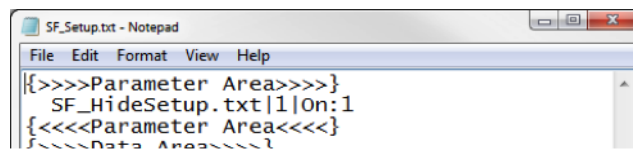
```
<Not in Use>|<Total DUT count>|<Enabled DUT number>|<Disabled DUT number>
```

Starting from the left,

The 1st parameter defines a filename that is not in use. Therefore, it can be ignored.

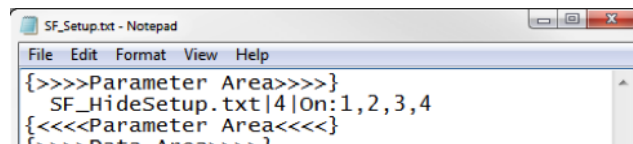
The 2nd parameter represents the total number of boards in the test program.

The 3rd and 4th parameter shows the DUT number which is enabled or disabled



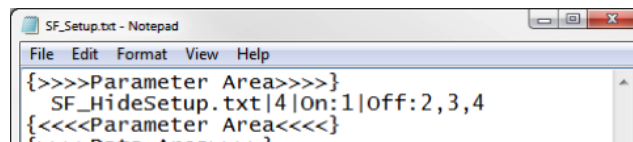
```
SF_Setup.txt - Notepad
File Edit Format View Help
{>>>>Parameter Area>>>>}
SF_HideSetup.txt|1|On:1
{<<<<Parameter Area<<<<}
{<<<<Data Area<<<<}
```

Example: 1up test program with DUT enabled for testing.



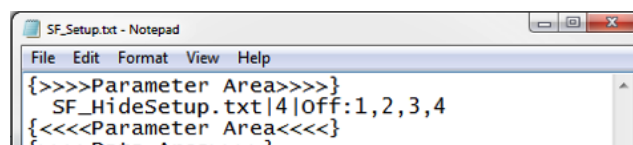
```
SF_Setup.txt - Notepad
File Edit Format View Help
{>>>>Parameter Area>>>>}
SF_HideSetup.txt|4|On:1,2,3,4
{<<<<Parameter Area<<<<}
{<<<<Data Area<<<<}
```

Example: 4up Panel test program with all 4 DUT enabled in the test program.



```
SF_Setup.txt - Notepad
File Edit Format View Help
{>>>>Parameter Area>>>>}
SF_HideSetup.txt|4|On:1|Off:2,3,4
{<<<<Parameter Area<<<<}
{<<<<Data Area<<<<}
```

Example: 4up Test program with DUT1 enabled and DUT2, 3 and 4 disabled for testing.



```
SF_Setup.txt - Notepad
File Edit Format View Help
{>>>>Parameter Area>>>>}
SF_HideSetup.txt|4|Off:1,2,3,4
{<<<<Parameter Area<<<<}
{<<<<Data Area<<<<}
```

Example: 4up Test program with all DUT disabled for testing.

Based on this information, the Shopfloor client can then process the required number of barcodes accordingly.

Data Section

The next section of the file is the data section which sends the general information of the test program and system. Usage of this information is optional on the Shopfloor client. These parameters are constants in the test program. The user set these parameters during the development of the test program; however, the i1000 software allows these to be overwritten by the shopfloor client using specific keywords within the transferred files.

SF_Setup.txt is the first file to be transferred to the shopfloor client during the initial startup. Therefore, the i1000 software includes all these parameters into the Data section for the shopfloor client to import. This sets the initial state of all the parameters. As the operation cycle moves forward, the shopfloor client may then decide whether to overwrite any of these parameters.

Keywords are case sensitive. The keywords are listed in the i1000_SF.INI file. The user may edit the file to customize the keywords if required. More details of the i1000_SF.INI file will be discussed later.

```
{>>>>Data Area>>>>}
  Fixture id :           [Fixture ID entry in System Parameter>Datalogging Tab]
  Operator id :         [Current User name in the i1000 software]
  Security :            [Current logged in User privilege level]
  Line :                [Line entry in System Parameter>Datalogging Tab]
  UUT type :           [UUT Type entry in System Parameter>Datalogging Tab]
  Controller :         [Controller entry in System Parameter>Datalogging Tab]
  Factory :            [Factory entry in System Parameter>Datalogging Tab]
  Department :         [Department entry in System Parameter>Datalogging Tab]
  Option 1 :           [Options entry in System Parameter>Datalogging Tab]
  Option 2 :           [Options entry in System Parameter>Datalogging Tab]
  Option 3 :           [Options entry in System Parameter>Datalogging Tab]
  Testplan id :        [Current loaded ATD file name]
  Testplan Rev. :     [Version number defined in ATD file]
  Shift :              [Work Shift number defined in Operator Interface]
  i1000 Version :     [i1000 software version]
{<<<<Data Area<<<<}
```


i1000_setup.tx

This file is sent from the Shopfloor client to the i1000 software to overwrite parameters in the i1000 software. Overwriting of parameters is usually not required unless there are specific requirements from the Shopfloor client or Server to push a certain value into the i1000 parameter.

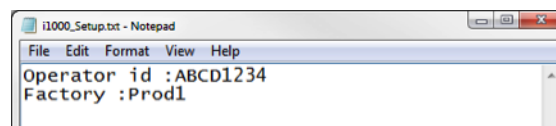
Therefore, in most cases, this file is just an empty file to signal to the i1000 software that the shopfloor client has received the SF_setup.txt file.

Parameters that can be overwritten by this file are listed in the Data Section of the SF_setup file where their initial value is being set at startup.

```
{>>>Data Area>>>}
  Fixture id :           [Fixture ID entry in System Parameter>Datalogging Tab]
  Operator id :          [Current User name in the i1000 software]
  Security :             [Current logged in User privilege level]
  Line :                 [Line entry in System Parameter>Datalogging Tab]
  UUT type :             [UUT Type entry in System Parameter>Datalogging Tab]
  Controller :           [Controller entry in System Parameter>Datalogging Tab]
  Factory :              [Factory entry in System Parameter>Datalogging Tab]
  Department :           [Department entry in System Parameter>Datalogging Tab]
  Option 1 :             [Options entry in System Parameter>Datalogging Tab]
  Option 2 :             [Options entry in System Parameter>Datalogging Tab]
  Option 3 :             [Options entry in System Parameter>Datalogging Tab]
  Testplan id :          [Current loaded ATD file name]
  Testplan Rev. :        [Version number defined in ATD file]
  Shift :                [Work Shift number defined in Operator Interface]
  i1000 Version :        [i1000 software version]
{<<<<Data Area<<<<}
```

Parameters that can be overwritten using the i1000_setup.txt file

Shopfloor client uses the keywords defined in the i1000_SF.INI file to select the target parameters. The keyword is entered to the file first followed by the desired value for that parameter. The example below shows the Operator ID and Factory parameter of the i1000 software being overwritten by the Shopfloor client to a value of “ABCD1234” and “Prod1” respectively using the i1000_setup.txt file.



Contents of i1000_setup.txt to overwrite Operator ID and Factor parameter

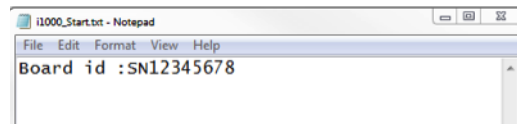
The Shopfloor client only needs to write the keywords of the selected parameters into the file. No headers and footers are required. There is no particular sequence for the writing of keywords as the i1000 software decodes each line in the file and identifies the parameter for change. However, it is imperative that the keywords **MUST** be entered correctly, including case and spaces as specified in the i1000_SF.INI file.

i1000_start.txt

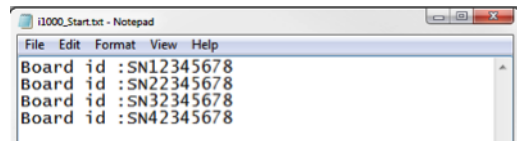
This file is sent from the Shopfloor client to the i1000 software. It signifies the “go ahead” to the i1000 software allowing testing to commence. It does not trigger the start of test, just allowing the trigger to start the test. Test start is only triggered either by the detection of the press engaged in the DOWN position or the manual click of the START button on the operator interface.

In almost all shopfloor setups, the serialization of the DUT is required even though it is not compulsory. Serializing the DUT allows the server to track each DUT along the production line and be able to identify and record all operations that the DUT has undergone. It also allows the control of the flow of the DUT so as to ensure that it does not miss any station along the production line.

Shopfloor clients will scan the serial labels on the DUT and then write them into the i1000_start.txt file using the defined keyword. If multiple serial numbers are required for a panel test, the shopfloor client simply append a new line below the first record and so on. In that case, the serial number for DUT1 will be written as the 1st line in the file followed by DUT2 in the 2nd line and so on.

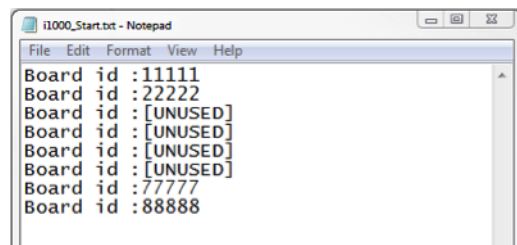


Example: i1000_start.txt file for a 1up test program



Example: i1000_start.txt file for a 4up test program

Using this file, the shopfloor client will also control the status of the DUT and can selectively disable them by writing their serial numbers as “[UNUSED]”. This will overwrite the Panel setup setting in the i1000 software. Therefore, when working with a panel test program that contains multiple DUT, shopfloor client is required to know which DUT in the test program is enabled or disabled so that it can set the correct status back into the i1000_Start.txt file. The example below shows an 8up Panel test program with boards 3 to 6 disabled.



Example: i1000_Start.txt file with an 8up test program with boards 3 to 6 disabled

SF_Start.txt

This file is sent from the i1000 software to the Shopfloor client. In the rare scenario where the serial number scanning is required to be handled by the i1000 software, the i1000 software will perform the scanning of the DUT barcode label and transfer it to the Shopfloor client.

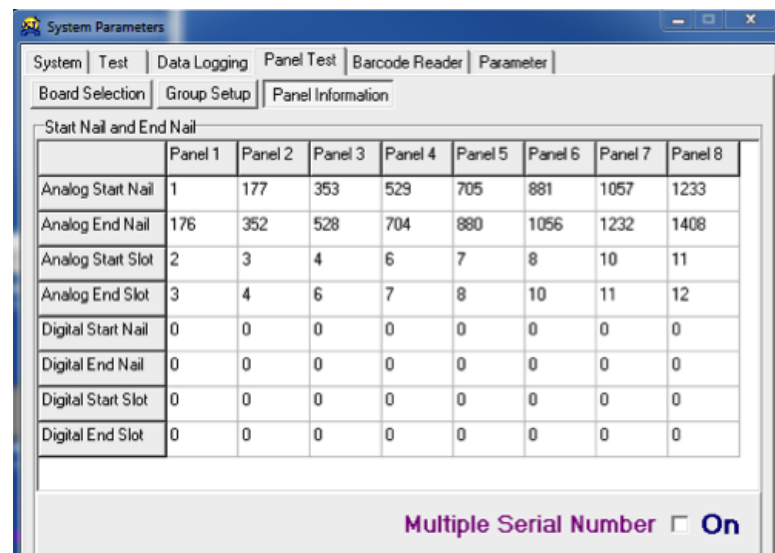
SF_Start.txt is created by the i1000 software when there is a serial number being entered into the i1000 operator interface under the Serial number field following a carriage return. This should normally happen before the transfer of i1000_Start.txt by the shopfloor client so that the serial number can be received and processed by the shopfloor client

When using SF_Start.txt, the serial number will first be populated into the Operator interface and upon carriage return, the i1000 software will generate SF_Start.txt into the transfer folder. If Barcode checking is not defined in the i1000_SF.INI file, the i1000 software will not wait for the i1000_Start.txt file and will allow testing to start. If Barcode check is defined, then the i1000 software will wait for the i1000_Start.txt to be received and replace the serial number with the updated entries from that file. Test can then commence.

The i1000 software allows the scanning of only one barcode label if shopfloor client is connected. In a panel test program that contains multiple DUTs, the user may choose to scan a panel barcode instead of multiple barcodes if it is available, or scan only the barcode of one of the DUTs, and use that as a panel barcode.

Although only one barcode is scanned for a panel test program, the file format of the SF_Start.txt will still contain entries of all other DUTs in the panel; however there will not be any data available for them.

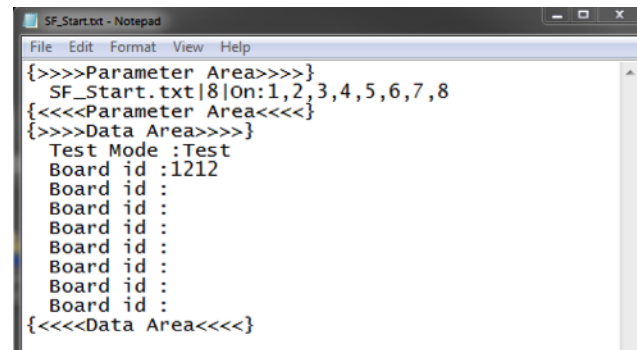
To enable scanning of just one DUT in a panel test program, the “Multiple Serial Number” option in the System parameter>Panel Test interface must be unchecked, otherwise the i1000 software will expect serial number inputs for all active DUTs in the test program.



System Parameter interface showing Panel Information. Multiple Serial Number option must be unchecked

The format of SF_Start.txt is divided into 2 sections. The Parameter Section is similar to the SF_Setup.txt.

The Data Section contains only the serial number entries. The example below shows a SF_Start.txt file for an 8up Panel test program. Only one serial number is scanned and written into the file while the rest of the Board id entries are left empty. Because the i1000 software scans only one serial number, it will default the serial to the first DUT number. Therefore, in the example below, the serial number <1212> is being placed onto the 1st entry for Board id.



```

{>>>Parameter Area>>>}
SF_Start.txt|8|On:1,2,3,4,5,6,7,8
{<<<Parameter Area<<<}
{>>>Data Area>>>}
Test Mode :Test
Board id :1212
Board id :
Board id :
Board id :
Board id :
Board id :
Board id :
Board id :
Board id :
{<<<Data Area<<<}

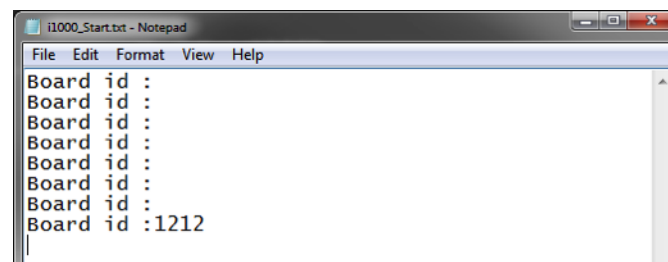
```

Example: Serial number written by i1000 software into SF_Start.txt

Upon receiving SF_Start.txt, the shopfloor client will process the serial number and then return it to the i1000_Start.txt to allow testing to start.

However, the i1000 software reads the Board ids sequentially and place them onto its Serial number field in the Operator Interface. This creates a problem when the shopfloor client writes the serial number into the 1st DUT in the i1000_Start.txt file because the serial number will eventually get overwritten by the blanks from the other DUTs that follow. If this happens, the Serial Number field in the Operator Interface will be left blank and the i1000 software will not allow the test to start because of a missing serial number entry in its interface.

To resolve this problem, the shopfloor client needs to write the serial number into the last DUT Board id entry in the i1000_Start.txt file. This will ensure that the serial number will not get overwritten. The example below shows the i1000_Start.txt file in response to the SF_Start.txt example that is shown above. Note that the serial number <1212> is being placed at the last Board id.



```

Board id :
Board id :
Board id :
Board id :
Board id :
Board id :
Board id :
Board id :1212

```

Example: Shopfloor client writes serial number into last DUT position

SF_End.txt

This file is sent from the i1000 software to the Shopfloor client. This file is generated at the end of a test cycle. It contains the results of the test cycle and also other parameters like test time and failure information.

The file contains a Parameter Section that is similar to that of the SF_SETUP.txt and also a Data Section that exports the details of the test cycle results using keywords listed in the i1000_SF.INI file. Shopfloor client may extract the required information from this file and upload them to the Shopfloor Server accordingly.

Upon processing this file, shopfloor client will then transfer the i1000_End.txt file as a response to the i1000 software. This signals the completion of a test cycle and shopfloor client will route back to wait for the serial number entry for the next test cycle.

```
{>>>>Parameter Area>>>>}
  SF_End.txt|1|0n:1
{<<<<Parameter Area<<<<}
{>>>>Data Area>>>>}
  Test Mode :                               [Mode. Test or Retest]
  Board id :                                 [Serial number of boards tested in the cycle]
  Board Result :                             [Test result, Failed Serial Number]
  Pretest Status :                           [Pretest Status]
  Discharge Status :                         [Discharge Status]
  Open Status :                              [Opens Test Status]
  Short Status :                             [Shorts Test Status]
  Component Status :                         [Component Test Status]
  Function Power On Status :                  [PowerON group Status]
  Function Status :                          [Function group Status]
  Digital Status :                           [Digital Library Test Status]
  Function Power Off Status :                 [PowerOFF group Status]
  Hi POT Status :                            [HIPOT Test Status]
  Test Result_<Group Type> Fail :            [Failure Group Type]
  Test Result :                              [Overall Test Result of the test cycle]
  Fail Report :                              [Location of Failure message file]
  Start Date :                               [Start Date format : YYMMDD]
  Start Time :                               [Start Time format : HHMMSS]
  Start Datetime :                           [Start DateTime format : YYMMDDHHMMSS]
  Duration :                                 [Test time in seconds]
  End Datetime :                              [End DateTime format : YYMMDDHHMMSS]
  End Date :                                 [End Date format : YYMMDD]
  End Time :                                 [End Time format : HHMMSS]
{<<<<Data Area<<<<}
```

Test result	Failure group type keywords
Open Fail	Test Result_Open Fail :
Short Fail	Test Result_Short Fail :
Component Fail	Test Result_Component Fail :
Function Fail	Test Result_Function Fail :
Digital Fail	Test Result_Digital Fail :
Pretest Fail	Test Result_Pretest Fail :
Discharge Fail	Test Result_Discharge Fail :
Function Power On Fail	Test Result_Function Power On Fail :
Function Power Off Fail	Test Result_Function Power Off Fail :
Function Discharge Fail	Test Result_Function Discharge Fail :
Hi POT Fail	Test Result_Hi POT Fail :
PostTest Fail	Test Result_PostTest Fail :

Keywords for failure group types

```

SF_End.txt - Notepad
File Edit Format View Help
{>>>Parameter Area>>>}
SF_End.txt|8|On:1,2|Off:3,4,5,6,7,8
{<<<Parameter Area<<<}
{>>>Data Area>>>}
Test Mode :Test

Board id :1111
Board Result :Pass,1111
Pretest Status :Tested
Discharge Status :Tested
Open Status :Tested
Short Status :Untested
Component Status :Tested
Function Power On Status :Tested
Function Status :Tested
Digital Status :Untested
Function Power Off Status :Tested
Hi POT Status :Tested

Board id :2222
Board Result :Fail,2222
Pretest Status :Tested
Discharge Status :Tested
Open Status :Tested
Short Status :Untested
Component Status :Tested
Test Result_Component Fail :Fail
Function Power On Status :Tested
Function Status :Tested
Digital Status :Untested
Function Power Off Status :Tested
Hi POT Status :Tested

Test Result :Fail
Fail Report :
Start Date :150312
Start Time :161946
Start datetime :150312161946
Duration :0.218
End datetime :150312161947
End Date :150312
End Time :161947
{<<<Data Area<<<}
    
```

Example: SF_End.txt file of a 2up panel test cycle

Empty Files

Several files used in this '2 files' transfer methods are empty files. Some of these files may contain data, but are not essential for the shopfloor client or the i1000 software to import and process.

Empty files	Behavior
SF_Ready.txt	This file is generated by the i1000 software whenever there is a file transfer operation. This signals to the shopfloor client that the transfer file is ready to be imported. Shopfloor client must clear or rename this file after each transfer operation so that it will not conflict with the next transfer operation.
I1000_Ready.txt	This file must be generated by the shopfloor client whenever there is a file transfer operation. This signals to the i1000 software that the transfer file is ready to be imported. I1000 software will rename this file to i1000_Ready.old after each transfer operation so that it will not conflict with the next transfer operation.
I1000_End.txt	This file must be generated by the shopfloor client upon the receipt of SF_END.TXT. This ends the test cycle operation at the i1000 software. The i1000 software will rename this file to i1000_End.old.
SF_Close.txt	This file is generated by the i1000 software each time the software closes, or a new ATD test program is loaded. Shopfloor client should monitor the existence of this file and close itself when detected. No further file transfer is required from the shopfloor client.
I1000_Abort.txt	This file is generated by the shopfloor client to force a test cycle to abort on the i1000. Upon abortion, the i1000 software returns to the idle state, waiting for the next test cycle to begin.
SF_Repeat.txt	Unused. To be removed. Generated at each end of test.
SF_Report.txt	Unused. To be removed. Generated at each end of test.
SF_HideSetup.txt	Unused. To be removed. Generated at the initial startup of the i1000 software.
SF_Show.txt	Unused. To be removed. Generated at each end of test.
ExtendedReport.txt	Unused. To be removed. Generated at each end of test.

List of empty files generated during a test cycle

Ensuring Exclusive Execution of Shopfloor Client Using Mutex

The i1000 software starts the Shopfloor client each time it launches or each time an ATD is loaded. If multiple instances of the client are running, it will create confusion and errors to the communication flow. As such, it is important to ensure that only one instance of the client is running at any time. To ensure this, a named Mutual Exclusion (Mutex) object must be created in the shopfloor client to allow the i1000 software to lock its access from other applications, including its own.

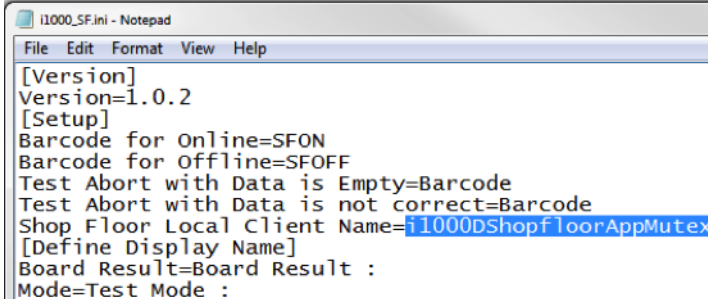
A Mutex is a program object that allows multiple threads to share the same program resource sequentially so that each thread can take turns to access to the resource and will not clash with each other. Mutex can be named or unnamed. An unnamed Mutex is usually created and used within a program only and it's not accessible to the other programs at the system wide level. A named Mutex on the other hand is global and can be accessed by other programs at the system level.

To ensure exclusive use of the shopfloor client to the i1000 software, this required a named Mutex that is accessible at the system level. With a named Mutex, the i1000 software can acquire the control of the resource by locking the Mutex so that other programs cannot use it. Similarly, the i1000 software can also check if the shopfloor client application is already running simply by checking the presence of the Mutex name in the system environment. This will prevent multiple copies of the shopfloor client to be launched at the same time. The Mutex name must be defined in the i1000 software so that before the i1000 software launches the shopfloor client, the i1000 will then attempt to acquire and lock the shopfloor client for its use.

The Mutex will continue to be locked by the i1000 software to ensure its exclusivity. When the i1000 software terminates, it triggers the shopfloor client to terminate as well. At this point, the shopfloor client needs to close and release the Mutex. There are different coding methods to create a Mutex depending on the programming language used for the Shopfloor client application. Examples and sample codes can be easily found on the internet for references.

Defining Mutex Name

Mutex name definition for the i1000 software is done with the i1000_SF.INI file. The file is located under C:\Agilent_i1000\INI folder.



```

i1000_SF.ini - Notepad
File Edit Format View Help
[Version]
Version=1.0.2
[Setup]
Barcode for Online=SFON
Barcode for Offline=SFOFF
Test Abort with Data is Empty=Barcode
Test Abort with Data is not correct=Barcode
Shop Floor Local Client Name=i1000ShopfloorAppMutex
[Define Display Name]
Board Result=Board Result :
Mode=Test Mode :
  
```

Example: i1000SF.INI file with Mutex defined

In the i1000_SF.ini file, there is an entry named “Shop Floor Local Client Name” which holds the Mutex name. The user needs to enter the Mutex in exactly the same manner that it was created in the Shopfloor client. As with all INI file editions, the i1000 software will need to be restarted for the edition to take place.

I1000_SF.INI

This INI file defines the parameters and keywords used during the shopfloor operation. Shopfloor client software must follow the parameters defined in this file. Parameters are user editable. The file is imported each time the i1000 software starts. Therefore, changes to the parameters in the file will not take effect until next time when the i1000 software starts. All values and keywords are case sensitive.

The file is divided into three sections.

The first section named [Version] tracks the revision of the file structure.

The second section named [Setup] defines the features of the i1000 shopfloor server. The software uses the keywords entered into the Serial Number field on the operator interface to control the startup or shutdown of the i1000 Shopfloor server. This provides a convenient method for the user to trigger the shopfloor server without needing to enter into the System Parameter interface. The keywords are defined under the statement “Barcode for Online” for ON, and “Barcode for Offline” for OFF.

The shopfloor server can be setup to validate the i1000_Start.txt file for the required data such as Serial Numbers or other parameters transmitted by the i1000_Setup.txt file.

In the third section [Define Display Name], any of the parameters listed here can be validated for their availability and accuracy, however not all will make sense to be validated. The common parameters used for validation are Barcode and Product. If the values are missing or incorrectly written into the i1000_Start.txt file, then the shopfloor server will generate an error and will abort the test. Multiple keywords each separated by a comma may be entered for validation.

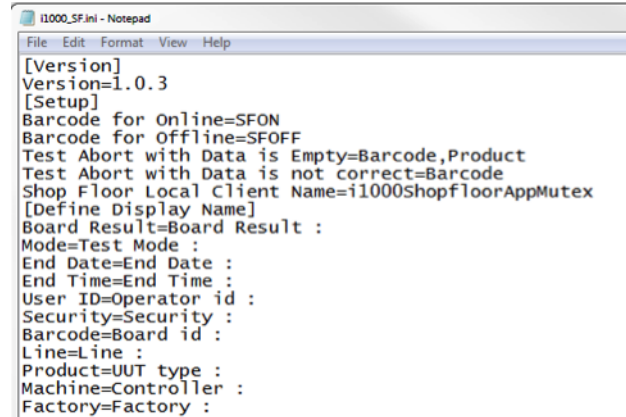
The Shop Floor Local Client Name defines the Mutex name used in the shopfloor client application as described earlier.

```
[Setup]
Barcode for Online=           [Keyword to turn ON shopfloor server]
Barcode for Offline=         [Keyword to turn OFF shopfloor server]
Test Abort with Data is Empty= [Keywords for validation]
Test Abort with Data is not correct= [Keywords for validation]
Shop Floor Local Client Name= [Defined the Mutex Name in the Shopfloor Client App]
[Define Display Name]
Board Result=Board Result :
Mode=Test Mode :
End Date=End Date :
End Time=End Time :
User ID=Operator id :
Security=Security :
Barcode=Board id :
...
...
```

The third section [Define Display Name] allows the user to customize the keywords to be used to represent the parameters in the communication between the shopfloor server and the client app. It is important that both the server and the client follow the definition in this section so that the information can be transmitted and received by each party.

When the shopfloor client sends the i1000_Setup.txt file, it can selectively include any of the keywords with the intended values into the file for the i1000 shopfloor server to import. However, only those parameters that are visible on the i1000 interface will be accepted by the shopfloor server. If any of the other keywords is used, the i1000 shopfloor server simply ignores them.

In the [Define Display Name] section, the parameter names are listed on the left side and the custom keywords are listed on the right. The user must not change the parameter names as these are already hardcoded into the i1000 software. Changing the keywords on the right of the “=” sign will also cause the display name on the i1000 interface to change.



```
i1000_SF.INI - Notepad
File Edit Format View Help
[Version]
Version=1.0.3
[Setup]
Barcode for Online=SFON
Barcode for Offline=SFOFF
Test Abort with Data is Empty=Barcode,Product
Test Abort with Data is not correct=Barcode
Shop Floor Local Client Name=i1000ShopfloorAppMutex
[Define Display Name]
Board Result=Board Result :
Mode=Test Mode :
End Date=End Date :
End Time=End Time :
User ID=Operator id :
Security=Security :
Barcode=Board id :
Line=Line :
Product=UUT type :
Machine=Controller :
Factory=Factory :
```

Example: i1000_SF.INI file showing validation for the parameter “Barcode” and “Product”

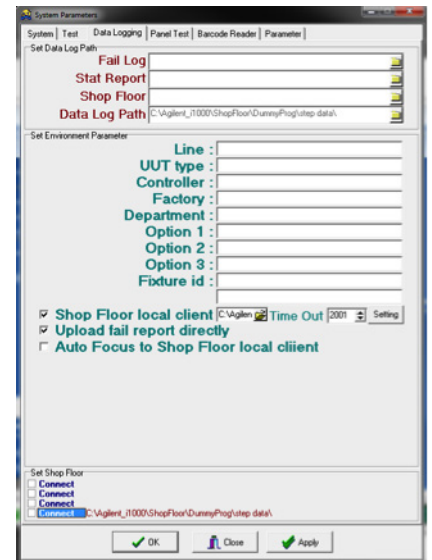
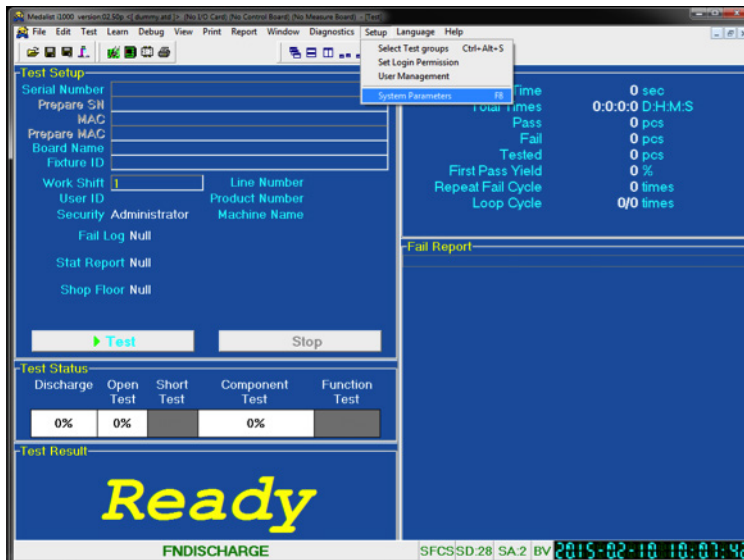
Setting up Shopfloor Client in the i1000 Interface

At the i1000 operator interface, the Shopfloor client can be set in the System Parameters window under the Datalogging tab.

1. Select the “Shop Floor local client” checkbox.
2. Click on the Open File icon on the right to select the Shopfloor client executable file.
3. Click Apply and the i1000 will launch the Shopfloor client immediately.
4. Click OK and Save the settings.

The settings are saved into the ATD test program so the next time the ATD is loaded to the i1000 software, the shopfloor client will be started automatically.

Shopfloor client will also be restarted each time the user clicks on the Apply or OK button in the System Parameter interface.



The path to the shopfloor client is absolute and not relative to the test program folder. Therefore, it is advisable to place the shopfloor client executable in a folder that is external to the test program. This will ensure that the path to the executable remains the same regardless of where the test program folder is placed.

Conclusion

Shopfloor processes work in conjunction with the parameter setup in the i1000SF.INI file. Therefore it is important for any developer of the shopfloor client to fully understand the format and behavior of each of the i1000 software files described in this document.



myKeysight

www.keysight.com/find/mykeysight

A personalized view into the information most relevant to you.

www.keysight.com/go/quality

Keysight Technologies, Inc.

DEKRA Certified ISO 9001:2008

Quality Management System

Keysight Channel Partners

www.keysight.com/find/channelpartners

Get the best of both worlds: Keysight's measurement expertise and product breadth, combined with channel partner convenience.

www.keysight.com/find/i1000D



For more information on Keysight Technologies' products, applications or services, please contact your local Keysight office. The complete list is available at: www.keysight.com/find/contactus

Americas

Canada	(877) 894 4414
Brazil	55 11 3351 7010
Mexico	001 800 254 2440
United States	(800) 829 4444

Asia Pacific

Australia	1 800 629 485
China	800 810 0189
Hong Kong	800 938 693
India	0124 229 2010
Japan	0120 (421) 345
Korea	080 769 0800
Malaysia	1 800 888 848
Singapore	1 800 375 8100
Taiwan	0800 047 866
Other AP Countries	(65) 6375 8100

Europe & Middle East

Austria	0800 001122
Belgium	0800 58580
Finland	0800 523252
France	0805 980333
Germany	0800 6270999
Ireland	1800 832700
Israel	1 809 343051
Italy	800 599100
Luxembourg	+32 800 58580
Netherlands	0800 0233200
Russia	8800 5009286
Spain	800 000154
Sweden	0200 882255
Switzerland	0800 805353
	Opt. 1 (DE)
	Opt. 2 (FR)
	Opt. 3 (IT)
United Kingdom	0800 0260637

For other unlisted countries:
www.keysight.com/find/contactus
(BP-02-06-15)

