MX100003A
MT1000A/MT1100A
Scenario Edit
Environment Kit
Operation Manual

13th Edition

- **For safety and warning information, please read this manual before attempting to use the equipment.**
- **Additional safety and warning information is provided within the M-W3933AE MT1000A Transport Modules Operation Manual the M-W3810AE MT1000A OTDR Modules Operation Manual, or M-W3735AE MT1100A Network Master Flex Operation Manual. Please also refer to them before using the equipment.**
- **Keep this manual with the equipment.**

ANRITSU CORPORATION

Document No.: M-W3858AE-13.0

Safety Symbols

To prevent the risk of personal injury or loss related to equipment malfunction, Anritsu Corporation uses the following safety symbols to indicate safety-related information. Ensure that you clearly understand the meanings of the symbols BEFORE using the equipment. Some or all of the following symbols may be used on all Anritsu equipment. In addition, there may be other labels attached to products that are not shown in the diagrams in this manual.

Symbols used in manual

⚠ DANGER This indicates a very dangerous procedure that could result in serious injury or death if not performed properly.

⚠ WARNING This indicates a hazardous procedure that could result in serious injury or death if not performed properly.

⚠ CAUTION This indicates a hazardous procedure or danger that could result in light-to-severe injury, or loss related to equipment malfunction, if proper precautions are not taken.

Safety Symbols Used on Equipment and in Manual

The following safety symbols are used inside or on the equipment near operation locations to provide information about safety items and operation precautions. Ensure that you clearly understand the meanings of the symbols and take the necessary precautions BEFORE using the equipment.

This indicates a prohibited operation. The prohibited operation is indicated symbolically in or near the barred circle.

This indicates an obligatory safety precaution. The obligatory operation is indicated symbolically in or near the circle.

This indicates a warning or caution. The contents are indicated symbolically in or near the triangle.

This indicates a note. The contents are described in the box.

These indicate that the marked part should be recycled.

MX100003A
MT1000A/MT1100A Scenario Edit Environment Kit
Operation Manual

18 March 2016 (First Edition)
10 February 2020 (13th Edition)

Equipment Certificate

Anritsu Corporation guarantees that this equipment was inspected at shipment and meets the published specifications.

Anritsu Warranty

- During the warranty period, Anritsu Corporation will repair or exchange this software free-of-charge if it proves defective when used as described in the operation manual.
- The warranty period is 6 months from the purchase date.
- The warranty period after repair or exchange will remain 6 months from the original purchase date, or 30 days from the date of repair or exchange, depending on whichever is longer.
- This warranty does not cover damage to this software caused by Acts of God, natural disasters, and misuse or mishandling by the customer.

In addition, this warranty is valid only for the original equipment purchaser. It is not transferable if the equipment is resold.

Anritsu Corporation shall assume no liability for injury or financial loss of the customer due to the use of or a failure to be able to use this equipment.

Anritsu Corporation Contact

In the event of this equipment malfunctions, contact an Anritsu Service and Sales office. Contact information can be found on the last page of the printed version of this manual, and is available in a separate file on the PDF version.

Notes On Export Management

This product and its manuals may require an Export License/Approval by the Government of the product's country of origin for re-export from your country.

Before re-exporting the product or manuals, please contact us to confirm whether they are export-controlled items or not.

When you dispose of export-controlled items, the products/manuals need to be broken/shredded so as not to be unlawfully used for military purpose.

Software End-User License Agreement (EULA)

Please read this Software End-User License Agreement (hereafter this EULA) carefully before using (includes executing, copying, registering, etc.) this software (includes programs, databases, scenarios, etc., used to operate, set, etc., Anritsu electronic equipment). By reading this EULA and using this software, you are agreeing to be bound by the terms of its contents and Anritsu Corporation (hereafter Anritsu) hereby grants you the right to use this Software with the Anritsu-specified equipment (hereafter Equipment) for the purposes set out in this EULA.

1. Grant of License and Limitations

1. Regardless of whether this Software was purchased from or provided free-of-charge by Anritsu, you agree not to rent, lease, lend, or otherwise distribute this Software to third parties and further agree not to disassemble, recompile, reverse engineer, modify, or create derivative works of this Software.
2. You may make one copy of this Software for backup purposes only.
3. You are not permitted to reverse engineer this software.
4. This EULA allows you to install one copy of this Software on one piece of Equipment.

2. Disclaimers

To the extent not prohibited by law, in no event shall Anritsu be liable for personal injury, or any incidental, special, indirect or consequential damages whatsoever, including, without limitation, damages for loss of profits, loss of data, business interruption or any other commercial damages or losses, arising out of or related to your use or inability to use this Software.

3. Limitation of Liability

a. If a fault (bug) is discovered in this Software, preventing operation as described in the operation manual or specifications whether or not the customer uses this software as described in the manual, Anritsu shall at its own discretion, fix the bug, or exchange the software, or suggest a workaround, free-of-charge. However, notwithstanding the above, the following items shall be excluded from repair and warranty.
   i) If this Software is deemed to be used for purposes not described in the operation manual or specifications.
   ii) If this Software is used in conjunction with other non-Anritsu-approved software.
   iii) Recovery of lost or damaged data.
   iv) If this Software or the Equipment has been modified, repaired, or otherwise altered without Anritsu's prior approval.
   v) For any other reasons out of Anritsu's direct control and responsibility, such as but not limited to, natural disasters, software virus infections, etc.

b. Expenses incurred for transport, hotel, daily allowance, etc., for on-site repairs by Anritsu engineers necessitated by the above faults shall be borne by you.

c. The warranty period for faults listed in article 3a above covered by this EULA shall be either 6 months from the date of purchase of this Software or 30 days after the date of repair, whichever is longer.

4. Export Restrictions

You may not use or otherwise export or re-export directly or indirectly this Software except as authorized by Japanese and United States law. In particular, this software may not be exported or re-exported (a) into any Japanese or US embargoed countries or (b) to anyone on the Japanese or US Treasury Department's list of Specially Designated Nationals or the US Department of Commerce Denied Persons List or Entity List. By using this Software, you warrant that you are not located in any such country or on any such list. You also agree that you will not use this Software for any purposes prohibited by Japanese and US law, including, without limitation, the development, design and manufacture or production of missiles or nuclear, chemical or biological weapons of mass destruction.

5. Termination

Anritsu shall deem this EULA terminated if you violate any conditions described herein. This EULA shall also be terminated if the conditions herein cannot be continued for any good reason, such as violation of copyrights, patents, or other laws and ordinances.

6. Reparations

If Anritsu suffers any loss, financial or otherwise, due to your violation of the terms of this EULA, Anritsu shall have the right to seek proportional damages from you.

7. Responsibility after Termination

Upon termination of this EULA in accordance with item 5, you shall cease all use of this Software immediately and shall as directed by Anritsu either destroy or return this Software and any backup copies, full or partial, to Anritsu.

8. Dispute Resolution

If matters of dispute or items not covered by this EULA arise, they shall be resolved by negotiations in good faith between you and Anritsu.

9. Court of Jurisdiction

This EULA shall be interpreted in accordance with Japanese law and any disputes that cannot be resolved by negotiation described in Article 8 shall be settled by the Japanese courts.

Protection Against Computer Virus Infections

Prior to the software installation

Before installing this software or any other software recommended or approved by Anritsu, run a virus scan on your computer, including removable media (e.g. USB flash drive and CF memory card) you want to connect to your computer.

When using this software and connecting with the measuring instrument

- Copying files and data
  On your computer, do not save any copies other than the following:
  • Files and data provided by Anritsu
  • Files created by this software
  • Files specified in this document
  Before copying these files and/or data, run a virus scan, including removable media (e.g. USB flash drive and CF memory card).
- Connecting to network
  Connect your computer to the network that provides adequate protection against computer viruses.
- Protection against malware (malicious software such as viruses).
  To connect your computer to network, the following is advised.
  • Activate Firewall.
  • Install important updates of Windows.
  • Use antivirus software.

Cautions on Proper Operation of Software

This software may not operate normally if any of the following operations are performed on your computer:

- Simultaneously running any software other than that recommended or approved by Anritsu
- Closing the lid (Laptop computer)
- Turning on the screen saver function
- Turning on the battery-power saving function (Laptop computer)

For how to turn off the functions, refer to the operation manual that came with your computer.

Use of Microsoft Copyrighted Content

Screenshots of Microsoft products in this documnent are used with permission from Microsoft.

About This Manual

This operation manual explains the operation of the MX100003A MT1000A/MT1100A Scenario Edit Environment Kit.

For the operation of the Network Master, refer to the following operation manual.

MT1000A Transport Modules Operation Manual (M-W3933AE)
MT1100A Network Master Flex Operation Manual (M-W3735AE)
MT1000A Network Master Pro OTDR Modules Operation Manual (M-W3810AE)

For the SCPI commands, refer to the following operation manual.

MT1000A Network Master Pro MT1100A Network Master Flex Remote Scripting Operation Manual (M-W3736AE)
MT1000A Network Master Pro OTDR Modules Remote Scripting Operation Manual (M-W3859AE)

This manual assumes that readers has the following knowledge.

- How to operate the Network Master
- Basic of software programming
  Experience of programming (using C or BASIC etc.)
  Knowledge of variable, subroutine, IF sentence, etc.
- Operation on Microsoft Windows
  Able to operate the mouse (click, drag, drop), file (load, copy), etc.

Table of Contents

IV.

Chapter 1 Outline of Scenario Edit Environment Kit

This section outlines the operation environment and functions of MX100003A MT1000A/MT1100A Scenario Edit Environment Kit (hereinafter referred to as "MX100003A").

1.1 Outline

MX100003A is the editing software of the scenario which runs on the following products.

- MT1000A Network Master Pro
- MT1100A Network Master Flex
- MT9083/MT9085 Series ACCESS Master

Scenario is the program which describes the following:
the process order of the commands and applications (sequence), the pass/fail thresholds values, the loading file name, and the saving file names.
By running the scenario created by MX100003A on the Network Master, the test automation will be realized.

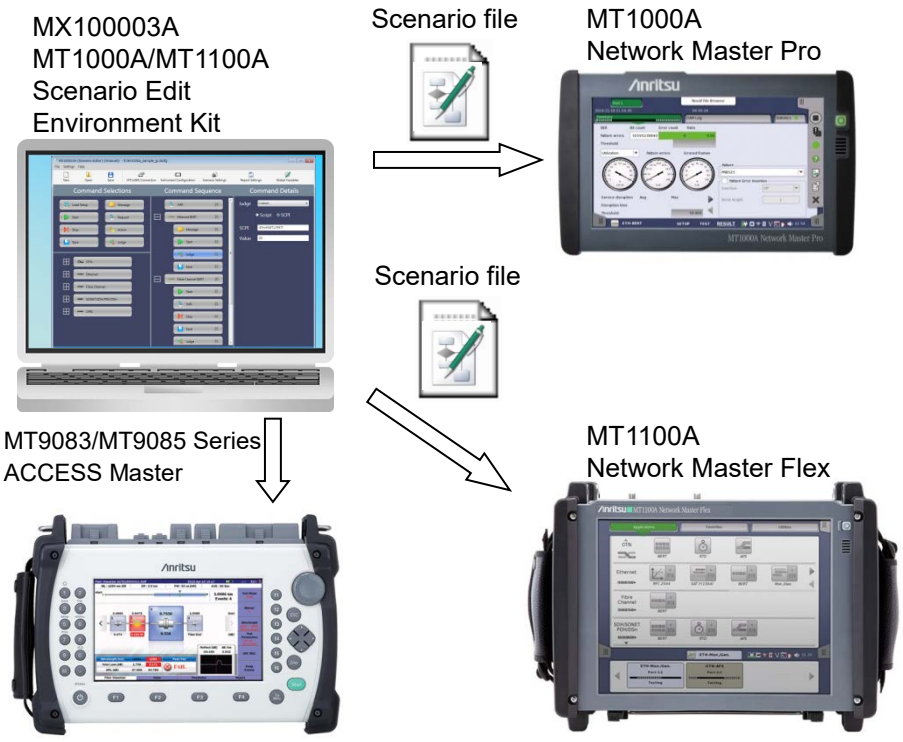


Figure 1.1-1 Use of MX100003A

Features

- Sequence can be edited by the drag & drop operation.
- Allows to edit the file stored in the Network Master by connecting to the Network Master via Ethernet.
- The scenario can be described by using the SCPI command used for remote controlling of the Network Master.
- Editing the script by using the test command allows creating the scenario which is suitable measurement method or judgement

condition to your purpose.
MX100003A can load the scripts from the file created by the text editor such as Note Pad of Windows.

1.2 Operating Environment

The following table shows the operating environment of MX100003A.

Table 1.2-1 Operating Environment

Item	Specifications
OS	Microsoft Windows 7, 8, 8.1 32 bit or 64 bit Microsoft Windows 10 64 bit
Display	1024 × 768 or more

Chapter 2 Installation and Uninstallation

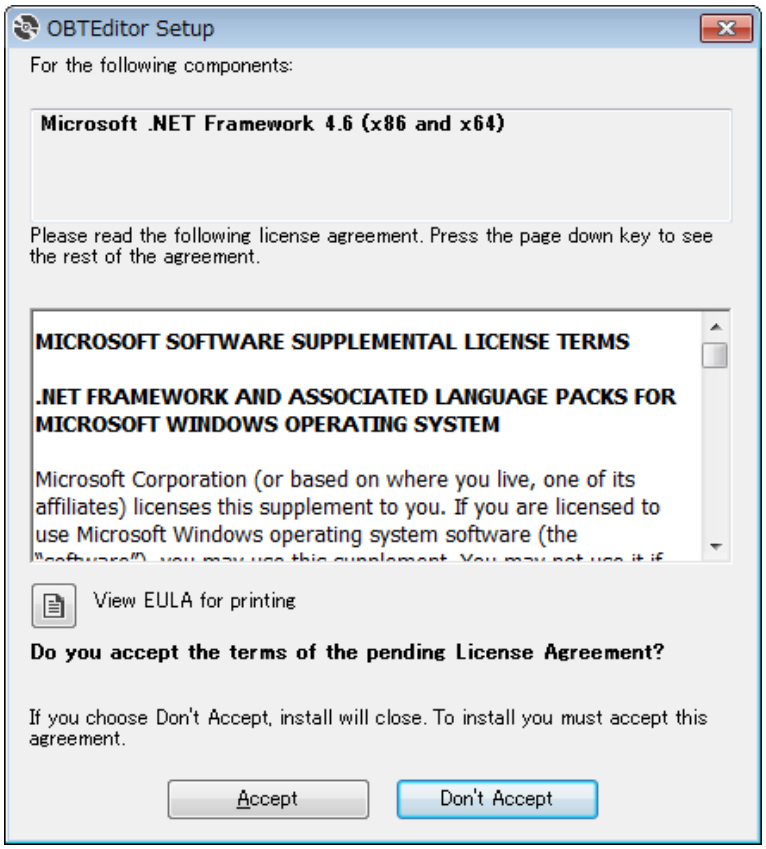
This chapter describes the installation and uninstallation of MX100003A, using examples when the OS of the personal computer (hereafter ,PC) is Windows 7.

2.1 Installation

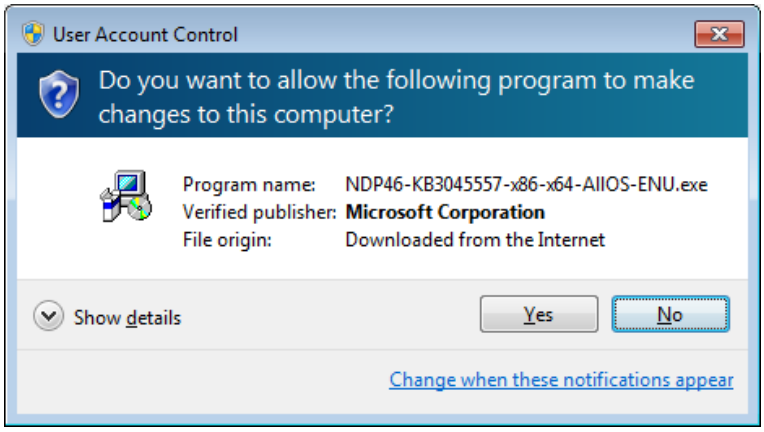
Install the MX100003A by the following procedure.

When NET Framework 4.6 is not installed in the PC

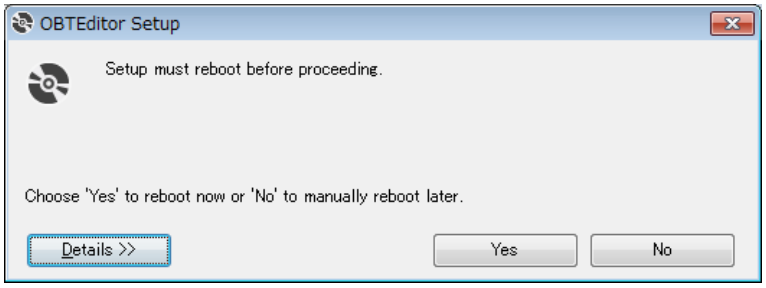
1. Copy MX100003A_xxx.zip (xxx is replaced by the version name) to PC.
2. Double-click the copied file to the PC. Double-click “Setup.exe” in the uncompressed folder. Double-click the “Setup.exe” in the extracted folder
3. The following dialog box appears. Click **Accept**.



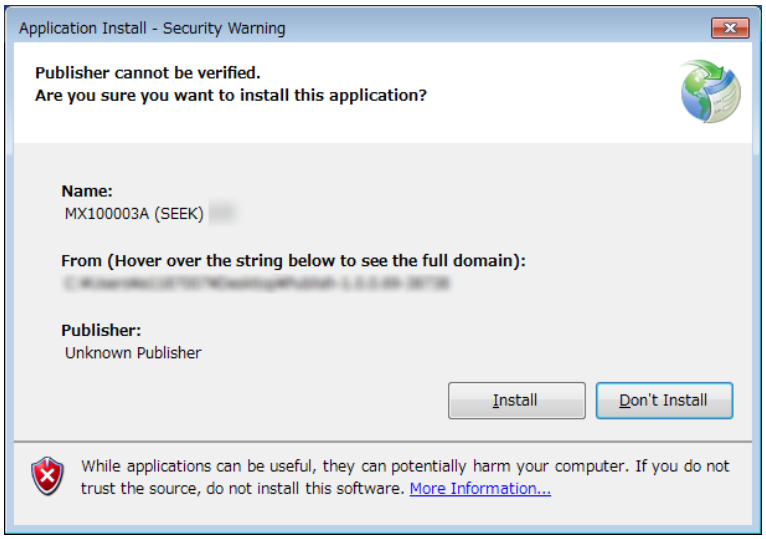
4. The following dialog box appears. Click **Yes**.



5. The following dialog box appears. Click **Yes**.



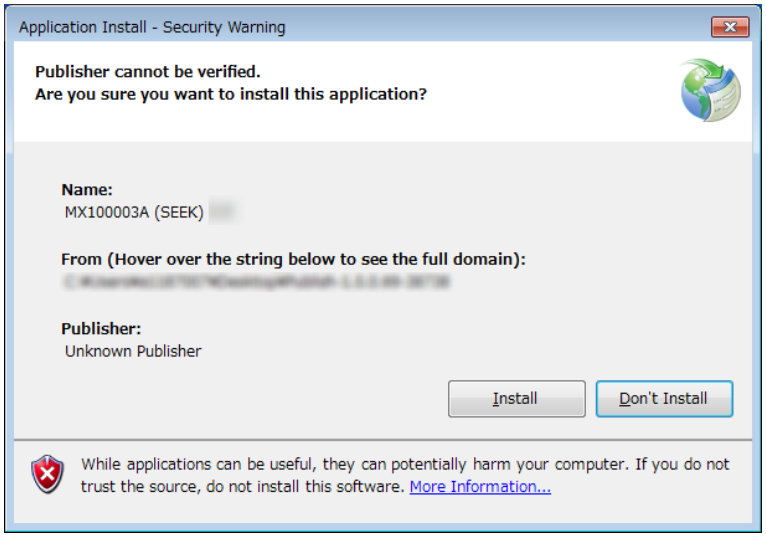
6. After PC reboots, the following dialog box appears. Click **Install**.



The MX100003A window appears after the installation has finished.

When NET Framework 4.6 is installed in the PC

1. Copy MX100003A_xxx.zip (xxx is replaced by the version name) to PC.
2. Double-click the copied file to the PC. Double-click “Setup.exe” in the uncompressed folder. Double-click the “Setup.exe” in the extracted folder

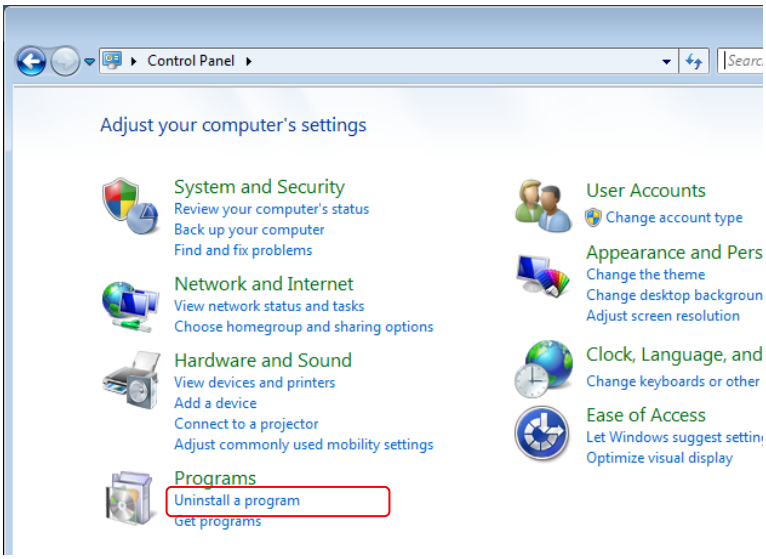


3. Click **Install**. The MX100003A window appears after the installation has finished.

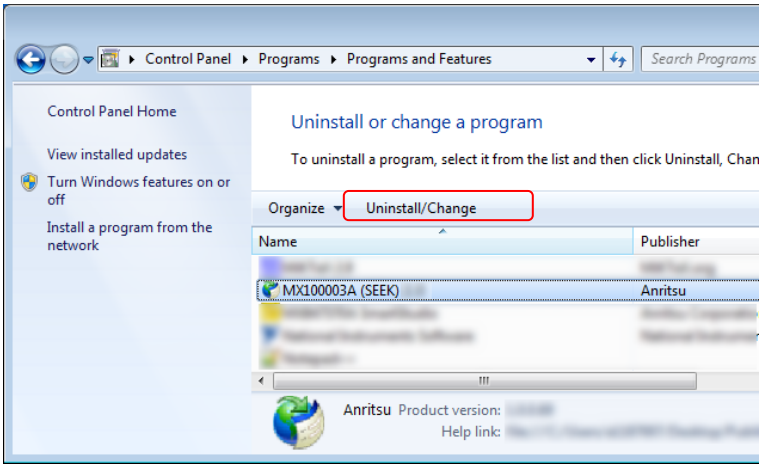
2.2 Uninstallation

This section describes the procedure for uninstalling MX100003A.

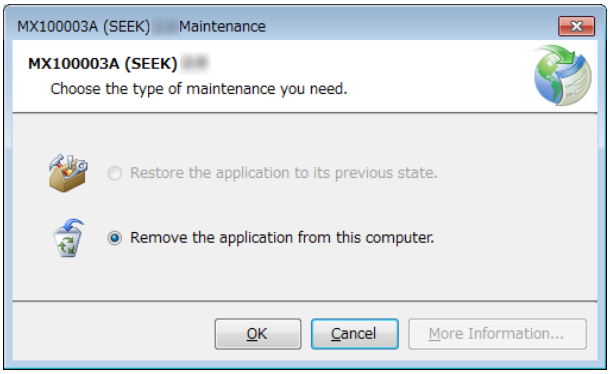
1. Select **Control Panel** in the **Start** menu to open the Control Panel.
2. Click **Uninstall a program**.



3. Click **MX100003A (SEEK)** in the list, and click **Uninstall/Change**.



4. Click **OK** on the following dialog box.



Chapter 3 Operation

This chapter explains the operations of the MX100003A.

3.1 Starting and Exiting

This section describes startup procedure and exit procedure of the MX100003A.

3.1.1 How to Start the Software

Click **Start Menu**, and click **MX100003A (SEEK)**.

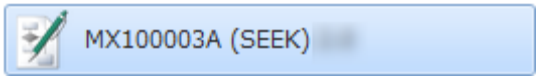


Figure 3.1.1-1 The Button on the Start Menu

3.1.2 How to Exit the Software

Exit the MX100003A by one of following operation.

- Click **Close** on the **File** menu.
- Click  at right-top of the window.

If the scenario on the way of editing has not saved, confirming message appears.

Yes: Exits with saving the scenario.

No: Exits without saving the scenario.

Cancel: Cancels exiting the MX100003A.

3.2 Explanation of the Window

3.2.1 Window Configurations

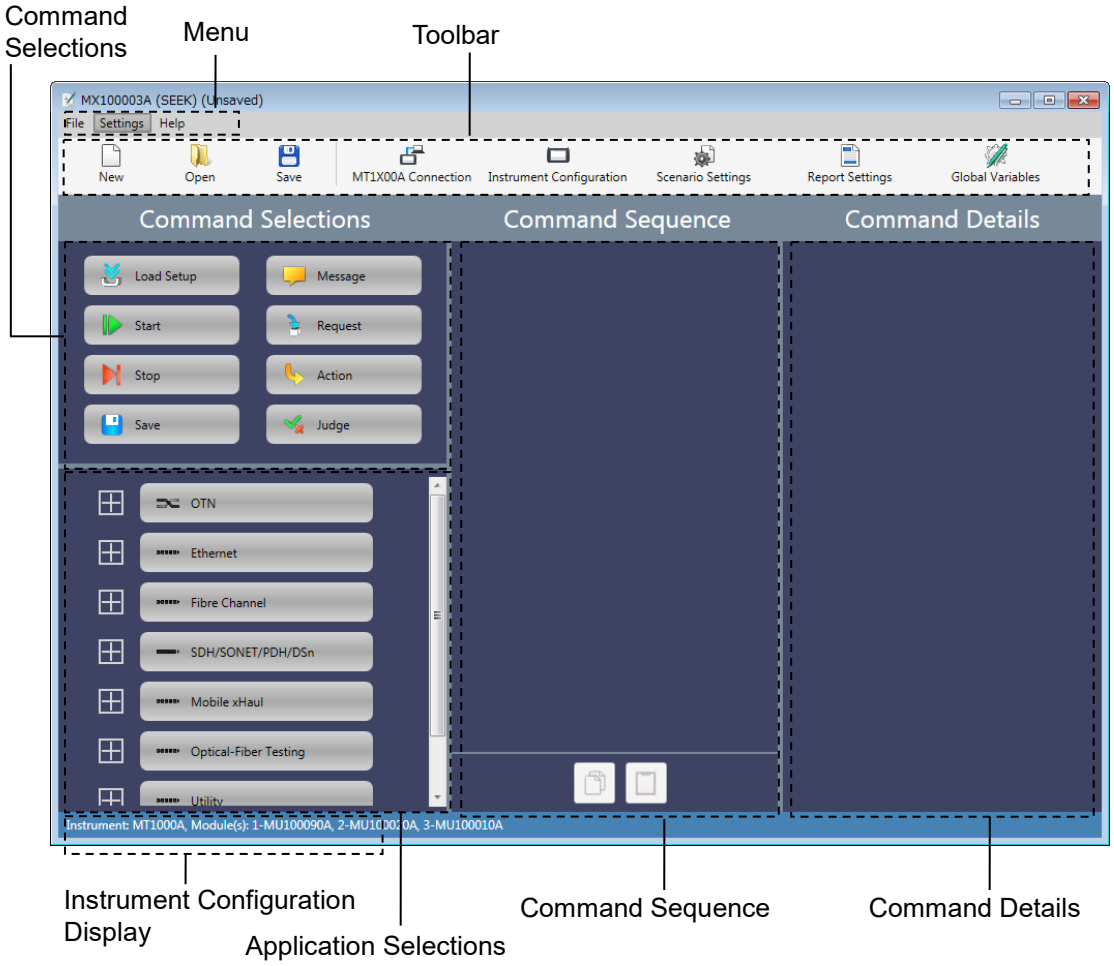


Figure 3.2.1-1 Window Components

Default language of MX100003A is English.

3.2.2 Menu

Menu configuration and icons displayed on the toolbar are shown following table.

Table 3.2.2-1 Menu Configuration

Menu	Icon	Description
File		
New		Creates the new scenario.
Open		Loads the scenario from the file.
Save		Saves the scenario to the file.
Save As	–	Saves the scenario to a file as another file.
Save To MT1x00A*	–	When the Network Master is connected via Ethernet, Saves the scenario in Internal folder of the Network Master.
Close	–	Exits MX100003A.
Settings		
Select Platform	–	Selects a platform (MT1x00A or MT9083) on which you want to run the scenario.
MT1x00A Connection*		Checks the connection to the Network Master which the IP address is specified.
Instrument Configuration		Sets the configuration of the Network Master and modules where the scenario will run.
Scenario Settings*		Sets the name and icon displayed on the Network Master.
Report Settings*		Sets the Report file information used when the report file is saved.
Global Variables*		Sets the variables used in the scenario. The variables can be referred from multiple applications executed in the scenario.
Language	–	Changes the languages of the MX100003A.
Help		
About	–	Displays the MX100003A information.

*: Not displayed when **MT9083** is selected for **Select Platform**.

3.2.3 Toolbar

On the toolbar, icons of File menu items and Setting menu items are placed. For the correspondence between the icon and the menu item, refer to Table 3.2.2-1 “Menu Configuration”.

3.2.4 Command Selections

"Command" is the operation to the application. The following commands are prepared in Command Selector.

Table 3.2.4-1 Commands List

Command	Description
Load Setup	Loads a setup file (*.cfg) of the application.
Start	Starts measurements.
Stop	Stops measurements.
Save	Saves scenario logs to files. Test results and Report file of the application can be also saved.
Message	Displays a message.
Request	Displays a dialog box to enter a value.
Action	Performs the user defined action.
Judge	Judges the test result of the application. If judged as "Fail", the running scenario stops.

The correspondence between icons on the Network Master screen and the commands is shown in the following figure. There are no icons corresponding to Message, Request, and Action.

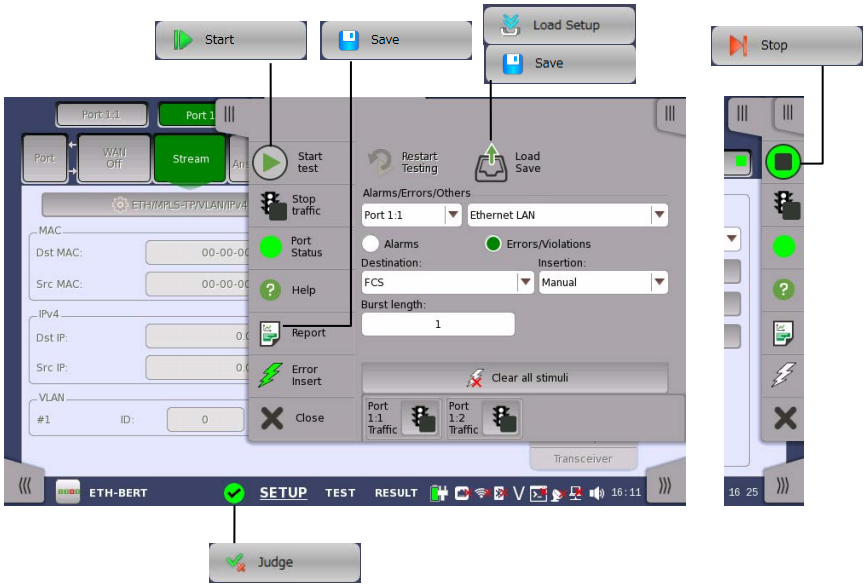


Figure 3.2.4-1 Correspondence Between Network Master Icons and Commands (Except Standard OTDR and VIP Applications)

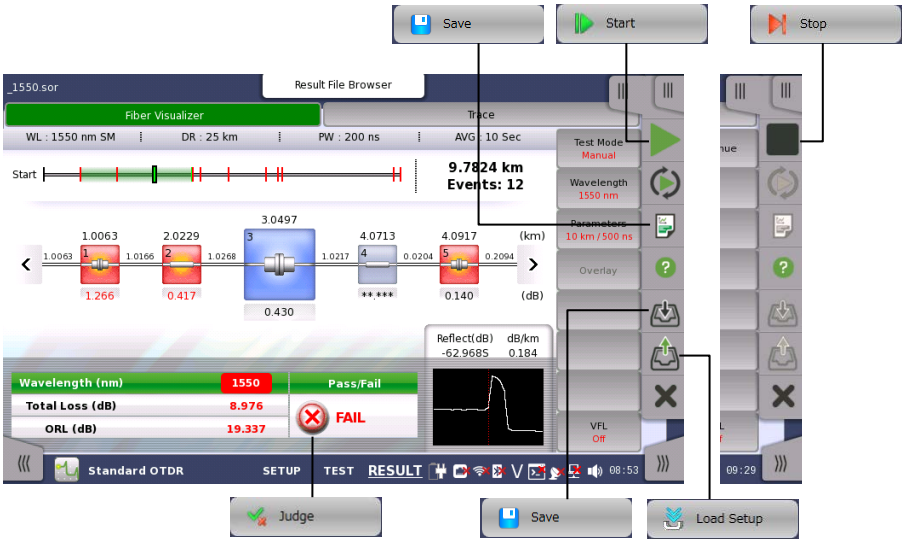


Figure 3.2.4-2 Correspondence Between Network Master Icons and Commands (Standard OTDR Application)

Figure 3.2.4-3 Correspondence Between Network Master Icons and Commands (VIP Application)

3.2.5 Application Selections

In Application Selections, the lists of application are displayed.
You can open the list by operating as below to the button of OTN, Ethernet, Fibre Channel, SDH/SONET/PDH/DSn, CPRI, Optical-Fiber Testing, and Other.

- Click  .
- Click the button and press the right arrow key of the keyboard.
- Double-click the button.

You can close the list by operating as below.

- Click  .
- Click the button and press the left arrow key of the keyboard.
- Double-click the button.

3.2.6 Command Sequence

Edit the order of processing by placing commands and applications to this area.
An application or a command can be placed by drag & drop operation.
The mouse icon changes to  when an application or a command cannot be placed.

You can open or close the list in Command Sequence in the same manner described in Application Selections.
The copy button and the paste button are located under the area.

 : copies an application or a command.

 : pastes the copied application or command.

3.2.7 Command Details

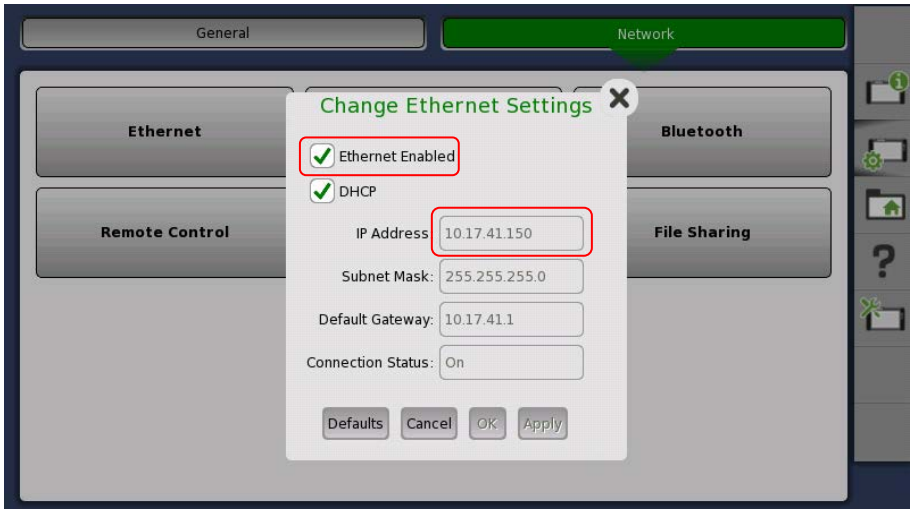
Clicking the command or the application placed in Command Sequence displays the setting items on this area.

For contents of displayed items, refer to 3.4.5.2 “Copy and Paste” and 3.4.7 “Editing the Command”.

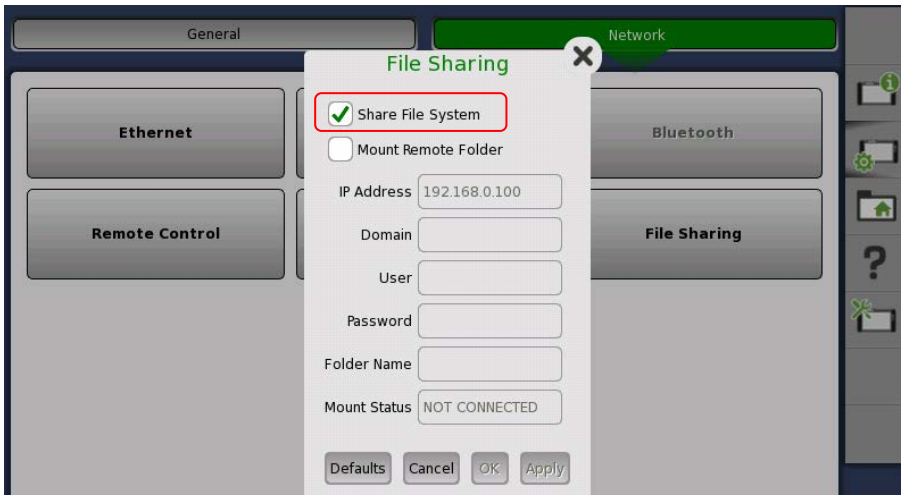
3.3 Connecting with the Network Master

MX100003A can access the folder of the Network Master by connecting to the Network Master via Ethernet.

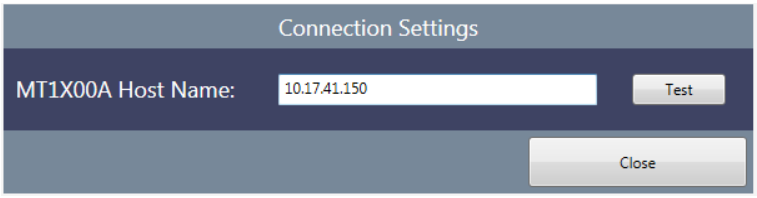
1. Select **Ethernet Enabled** on the Network Master, confirm the IP address.



2. Select **Share File System** on the Network Master.



3. Connect Network Master and PC using the Ethernet cable.
4. Click **MT1x00A Connection** on MX100003A.



5. Enter the Network Master IP address confirmed in step 1 and click **Test**. If MX100003A has connected to the Network Master, Success message appears.

3.4 Editing the Scenario

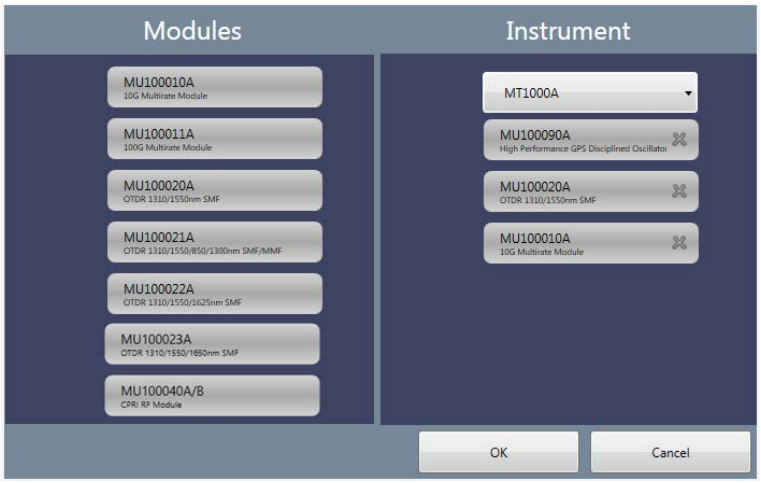
Click  on toolbar to load the scenario file or click  to start editing new scenario.

3.4.1 Instrument Configuration

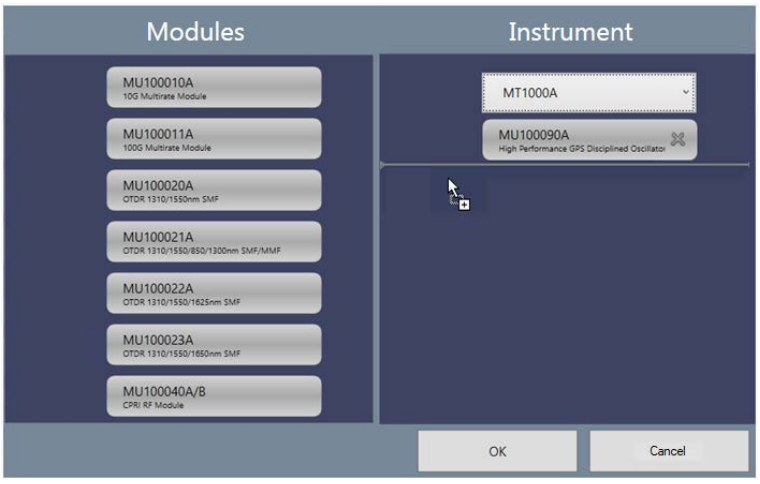
Set the hardware configuration of the scenario.

When **MT1X00A** is selected for **Select Platform**

1. Click **Instrument Configuration** on the toolbar. The following dialog box appears.

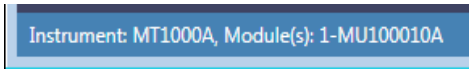


2. Select **MT1000A** or **MT1100A** by pull-down menu on Instrument. Available module(s) appear on Modules area.
3. Click the module button and drag it to Instrument.
   To delete the module on Instrument, click  on the button.



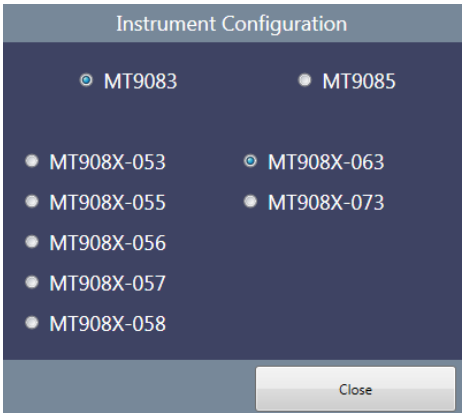
4. Click **OK**.
5. If the current scenario is not saved, the message confirming the edited scenario will be lost appears. Click **Yes** if change the

Instrument Configuration. Instrument Configuration is displayed at left bottom of the window.

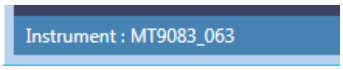


When **MT9083/MT9085** is selected for **Select Platform**

1. Click **Instrument Configuration** on the toolbar. The following dialog box appears.



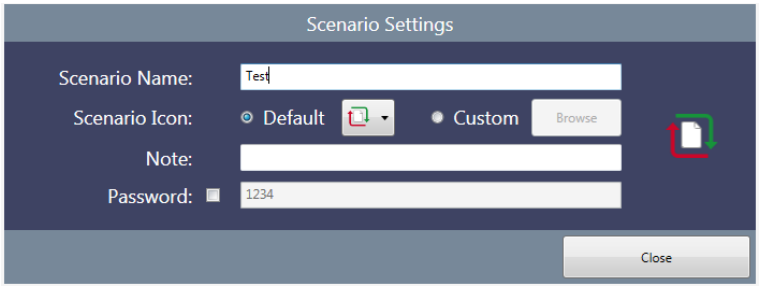
2. Select an MT9083/MT9085 option you want to use when running the scenario.
3. Click **Close**. Configuration is displayed at left bottom of the window.



3.4.2 Scenario Settings

Set the scenario information to display on the Network Master screen.

1. Click **Scenario Settings** on the toolbar. The following dialog box appears.



2. Enter the scenario name up to 12 characters in alphabetic-numeric. You cannot set Scenario Name as blank.
3. Click **Browse**. Specify the image file displaying as the icon by the dialog box.
4. Enter the comment.
5. Select the Password check box if requiring the password when running the scenario. Enter a 4 to 8-digit password.

6. Click **Close**.

Example of Scenario Settings and Displays on the Network Master are shown below.

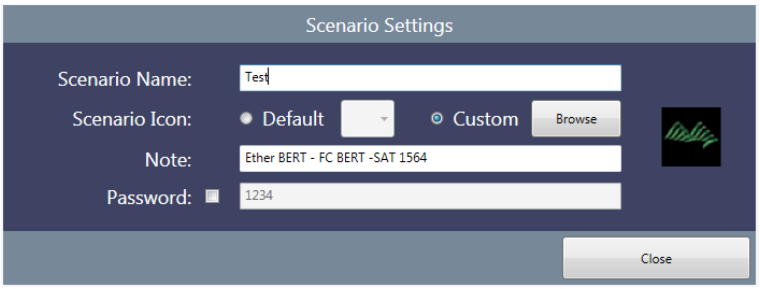


Figure 3.4.2-1 Example of Scenario Settings

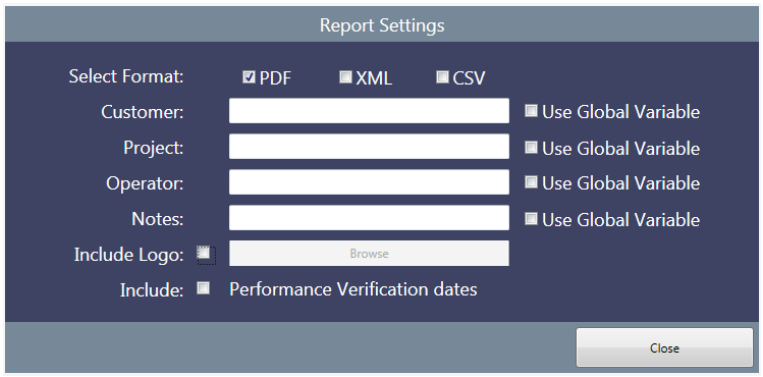
Figure 3.4.2-2 Display Example on Network Master (Scenario Manager)

Figure 3.4.2-3 Display Example on Network Master (Utilities)

3.4.3 Report Settings

Set the Report file header information of the application.

1. Click **Report Settings** on the toolbar. The following dialog box appears.



2. Select **Select Format** check box(es) to specify the report file format.
3. Enter Customer, Project, Operator and Note.
   If **Use Global Variable** check box has selected, the variable name appears in Global Variables dialog box and the variable value cannot be set in Report Settings dialog box.

Table 3.4.3-1 Variable Names of Report Settings

Items in Report Settings	Global Variable Name
Customer	REPORT_SETTING_CUSTOMER
Project	REPORT_SETTING_PROJECT
Operator	REPORT_SETTING_OPERATOR
Notes	REPORT_SETTING_NOTES

4. To include a logo in Report, select **Include Logo** check box and click **Browse**. Specify the image file of the log using the dialog box.
5. To include the Performance Verification Dates in Report, select **Include** check box.
6. Click **Close**.

The report file in PDF format will be saved in the designated folder after running the scenario.

Document Information

Report Name	PTP_unicast
Customer	
Project	
Operator	
Notes	

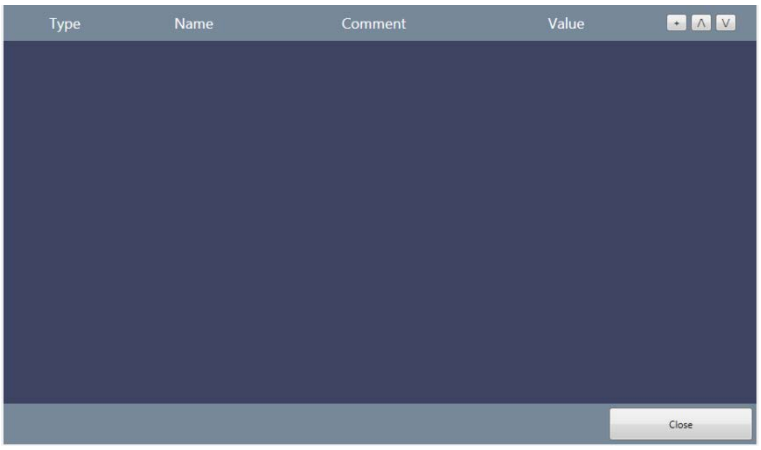
Module Type	Serial no	Software Version
MT1000A	708230006	5.03
MU100010A	N/A	

Figure 3.4.3-1 Report File Header Example

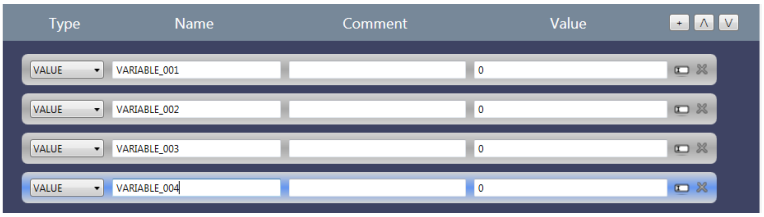
3.4.4 Global Variables

Define the variables used in the scenario. The variables defined in this dialog box can be referred from multiple applications executed in the scenario.
Also, they can be edited on the Network Master screen. For the operation of the Network Master, refer to 4.2.3 "Editing the Scenario".

1. Click **Global Variables** on the toolbar. The following dialog box appears.



2. To add a variable, click [+].



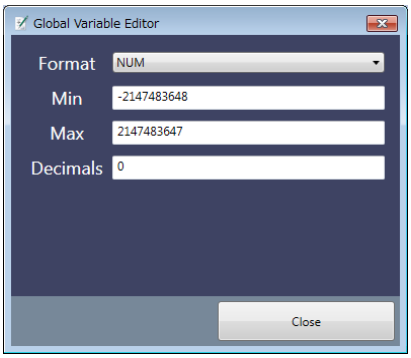
To change the order, click [Λ] or [V] .Clicking [⌘] deletes the variable.

3. Select Type from the following.

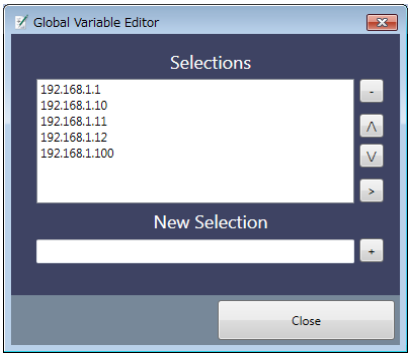
Table 3.4.4-1 Type of Global Variables

Type	Description
MAC	A hexadecimal number in MAC address format.
IPV4	A decimal number in IPv4 address format.
IPV6	A hexadecimal number in IPv6 address format.
STRING	A string.
VALUE	A numeric value. Following can be set. Format: BIN (Binary), NUM (Decimal), HEX (Hexadecimal) Min: Minimum value Max: Maximum value Decimals: Number of digits under the decimal point
LIST_STR	Enter string for appending the list item. Numeric value is processed as string.

4. Enter the variable name in Name column.
5. Enter description of the variable in Comment column.
6. Enter default value of the variable in Value column.
7. When Type is set to VALUE or LIST_STR, click  to open Global Variable Editor dialog box.
8. When Type is set to VALUE, enter Format, Min. , Max. and Decimals.



When Type is set to LIST_STR, enter the string in **New Selection** and click  . You can change order by clicking the item in **Selection** and click  or  . To delete a list item, click the item in **Selection** and click  .



9. Click **Close** or  .

3.4.5 Editing the Sequence

3.4.5.1 Placing Applications and Commands

Edit the sequence by dragging an application or a command to this area.
There are two positions they can be placed:
Outer application and Inner application.

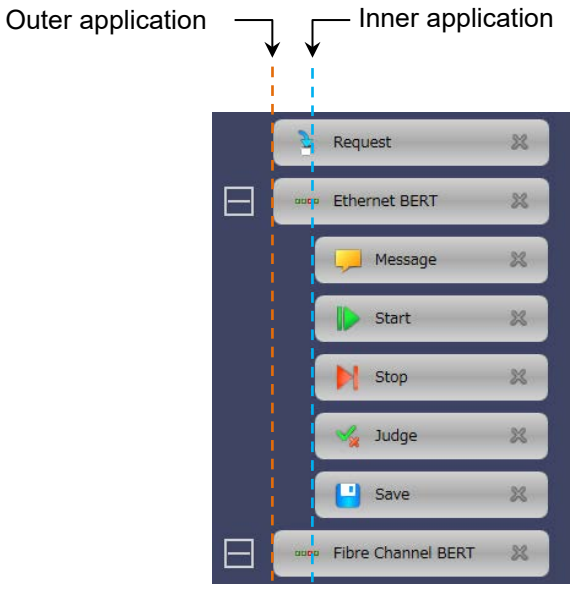


Figure 3.4.5-1 Positions to Place Applications and Commands

Place applications at Outer application. Place commands excluding Request at Inner application.
Only Request can be placed at both positions.
When Request is placed at Outer application, the inputted variable can be referred from the applications executed subsequently.
When Request is placed at Inner application, the inputted variable can be referred from the only applications executing.

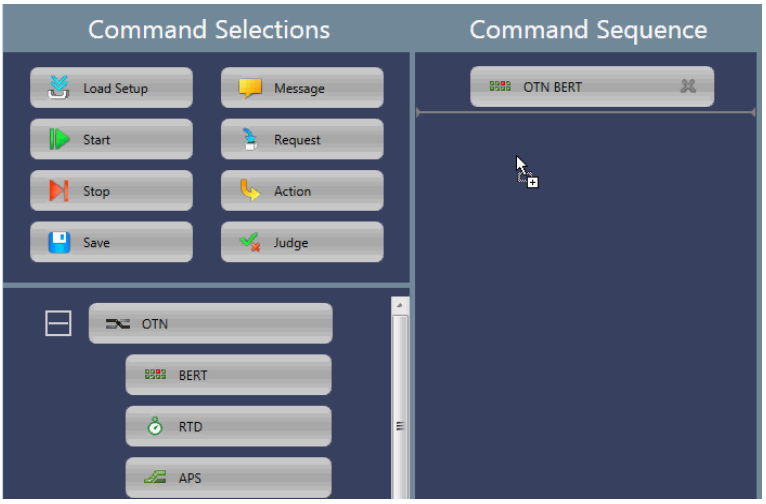


Figure 3.4.5-2 Position to Place Applications

Place commands at Inner application. Gray line appears if placing a command to Command Sequence area.
Command will be placed if dropping the command when length of the gray line is the same as application button width.

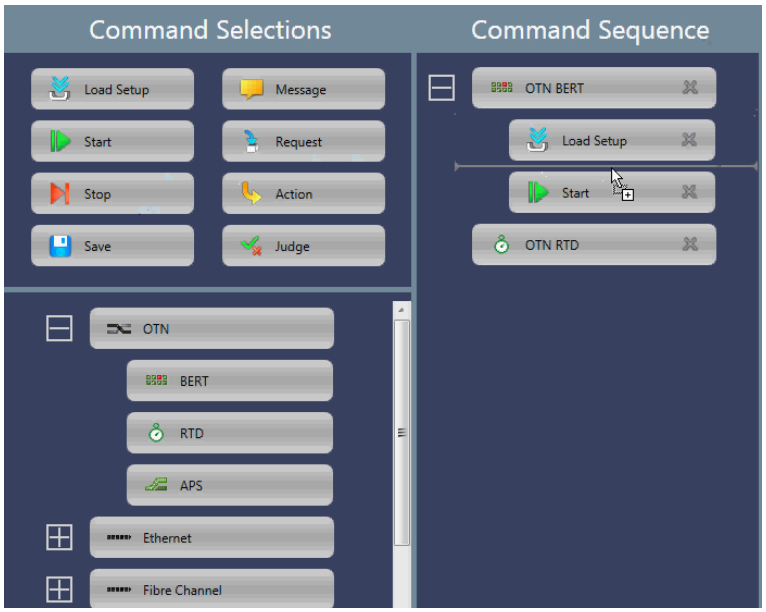


Figure 3.4.5-3 Position to Place Commands

Gray rectangle appears if dragging a command to the application button in the Command Sequence area. Dropping the command in this case will append to the last position of the application.

Figure 3.4.5-4 Appending a Command at Last Position of the Application

Note:

The command once placed in the application cannot be moved to other application. In the example of Figure 3.4.5-4, you cannot move Start command in OTN BERT to OTN RTD.

To delete the placed command:

- Click the application or command and press **Delete** or **Back Space** of the keyboard.
- Click on the application or command. Click on the displayed message.

Notes:

- If deleting the application or command, you can not restore it.
- If the application is deleted, all the commands placed in the application will be deleted.

The custom application of Other can write the whole process from starting to quitting the application in the SCPI command. Thus, the custom application can place only the following commands.

Message, Request, Action, Judge

3.4.5.2 Copy and Paste

You can copy and paste an application or a command in the sequence. You cannot cancel (Undo) or repeat (Redo) the operation.

Copying and Pasting an Application

1. Place applications and commands in the area.
2. Click the desired application (in the example, **OTN RTD**).
3. Click the copy button.
4. Click the application (in the example, **OTN APS**) where you want to paste the copied application.
5. Click the paste button. The copied application is inserted under the application selected at step 4.

If clicking the paste button without selecting the application, the copied application is inserted at the end of the sequence.

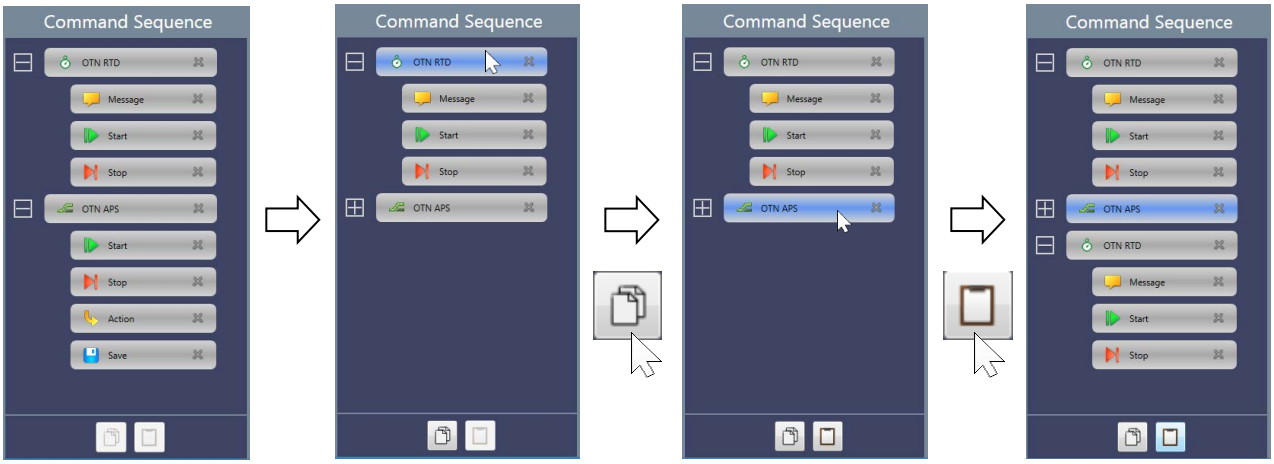


Figure 3.4.5.2-1 Copying and Pasting an Application

When a command has been clicked after copying an application, the paste button will be disabled. Right-clicking the selected command enables the paste button.

Copying and Pasting a Command

1. Place applications and commands in the area.
2. Click the desired command (in the example, **Message** in OTN RTD).
3. Click the copy button.
4. Click the command (in the example, **Stop** in OTN APS) where you want to paste the copied application.
5. Click the paste button. The copied command is inserted under the command selected at step 4.

If clicking the paste button after selecting the application, the copied command is inserted at the end of the sequence of the application.

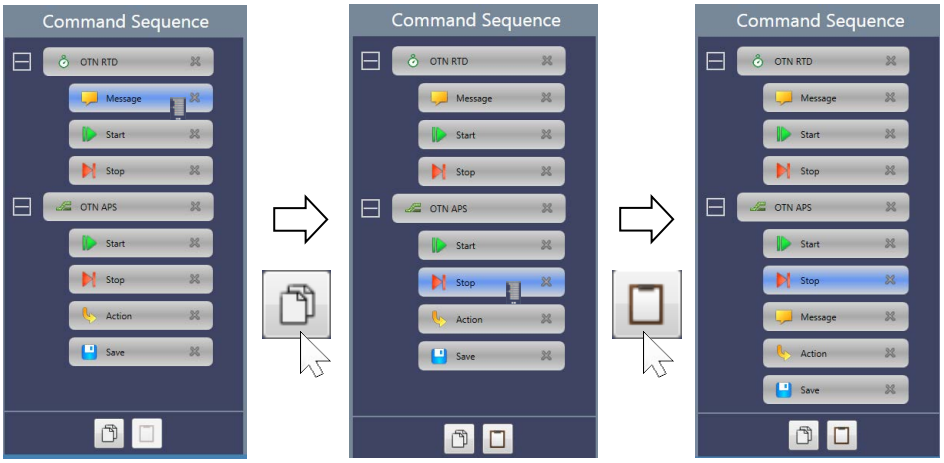


Figure 3.4.5.2-2 Copying and Pasting a Command

The following restrictions apply to copying a command

- Only one Load Setup command can be placed in the application sequence. Therefore, multiple Load Setup commands cannot be pasted into the same application.
- Some commands cannot be pasted between Optical Fiber Testing application and other applications.

3.4.6 Editing the Application

Clicking the application placed in the Command Sequence allows to select the port(s) to use in the Command Detail area.

Launch with + OTN:

Displayed for the application to which OTN layer can be added, select whether adding OTN layer or not.

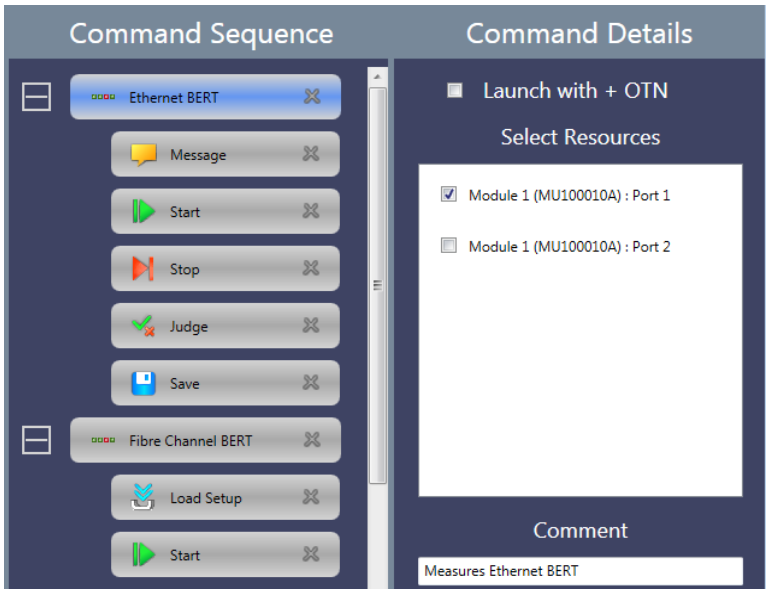


Figure 3.4.6-1 Application Details

3.4.7 Editing the Command

Clicking the command placed in the Command Sequence allows to edit the parameter(s) in the Command Detail area.

3.4.7.1 Load Setup

Set the setup filename (*.cfg) of the application to load.

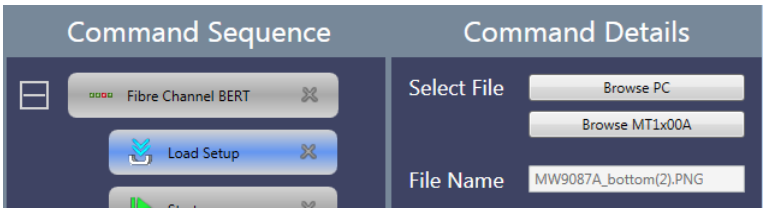


Figure 3.4.7.1-1 Load Setup Details

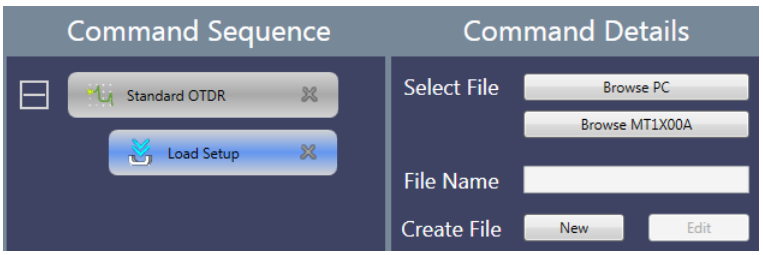


Figure 3.4.7.1-2 Load Setup Details (For Standard OTDR)

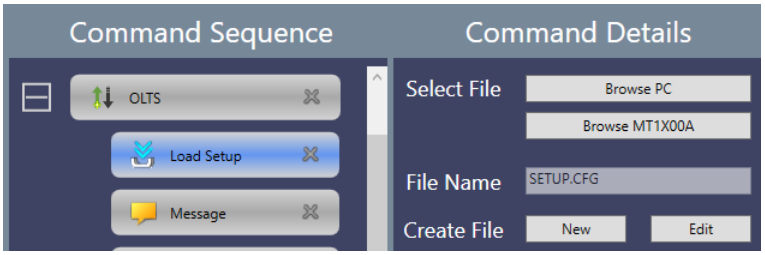


Figure 3.4.7.1-3 Load Setup Details (For OLTS)

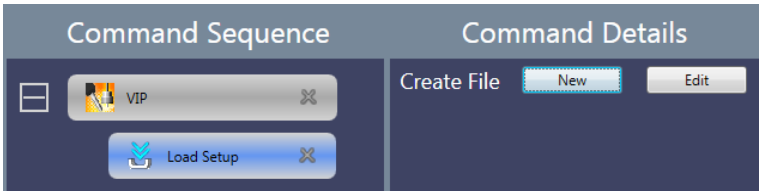


Figure 3.4.7.1-4 Load Setup Details (For VIP)

Select File

Load the application settings from the configuration file. The name of the loaded configuration file is displayed in the **File Name** field.

To select the file saved in the PC, click **Browse PC**.

To select the file saved in the folder of Network Master, click **Browse MT1X00A**.

Create File

For the Standard OTDR application, OLTS application, and the VIP application, **Create File** is displayed.
This allows you to newly create a configuration file or edit an existing configuration file.

New: Newly creates a configuration file, discarding any changes you have made.

Edit: Allows you to edit the current settings.

Clicking **New** or **Edit** opens the following dialog box.

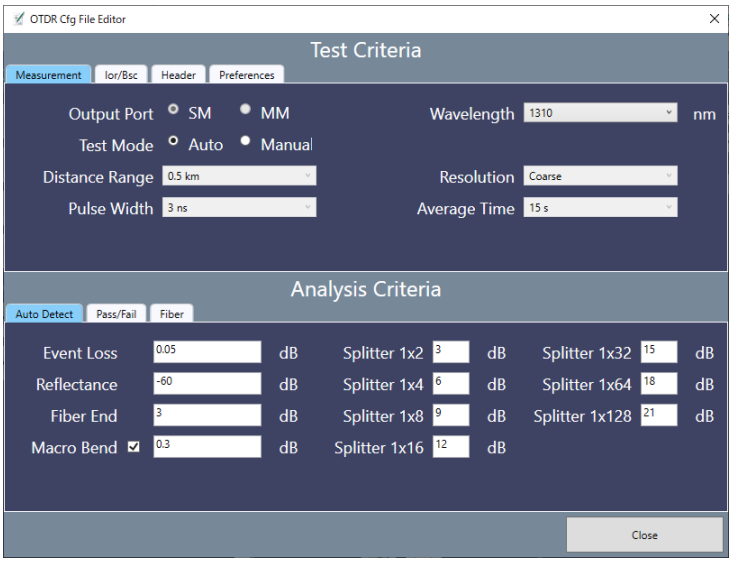


Figure 3.4.7.1-5 OTDR CFG File Editor (For MU100021A)

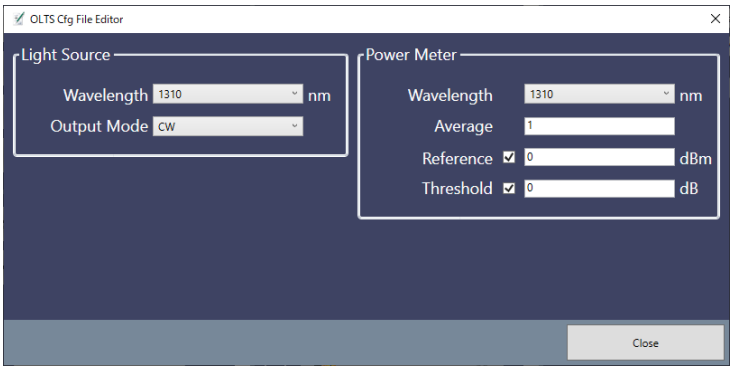


Figure 3.4.7.1-6 OLTS CFG File Editor

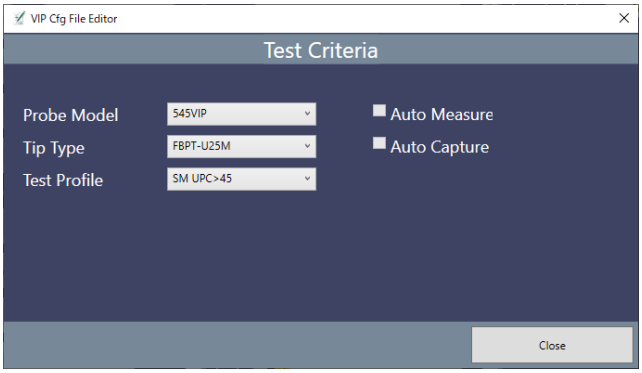


Figure 3.4.7.1-7 VIP CFG File Editor

For descriptions of the setting items, refer to the MT1000A Network Master OTDR Modules Operation Manual (M-W3810AE).

3.4.7.2 Start

Set how to start the test. Selectable options vary according to the application. The parameters for some applications cannot be edited.

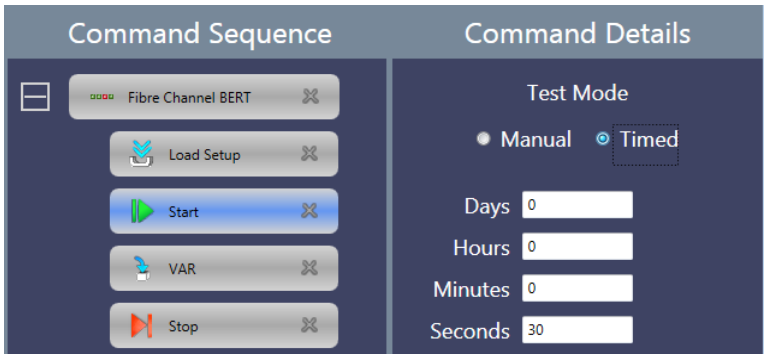


Figure 3.4.7.2-1 Start Details

Auto: Starts the test immediately and stops the test automatically when the measurement completes.

Timed: Starts the test immediately and stops the test when the specified time has elapsed.

Manual: Starts the test if or on the Network Master has touched.

3.4.7.3 Stop

There are no parameters to set for the Stop command.

Note:

Stop command cannot be placed in the VIP application.

3.4.7.4 Save

Set the filename to save. The file is saved in the Internal/Scenario_logs folder of Network Master.

Note:

Save command cannot be placed in the Discovery application.

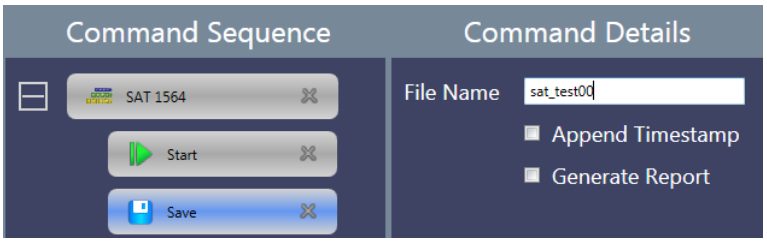


Figure 3.4.7.4-1 Save Details

Append Timestamp: Appends data and time to the filename.
Generate Report: Generates the Report file and saves.

When %1 is entered in **File Name**, **Select Variable** appears. The variables defined in 3.4.4 “Global Variables” can be selected. The string selected for **Select Variable** will be the file name automatically.

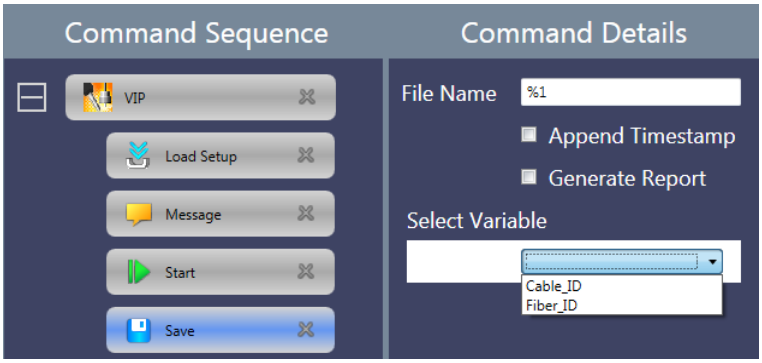


Figure 3.4.7.4-2 Select Variable

3.4.7.5 Message

Set the title, text, and image displayed in the message.

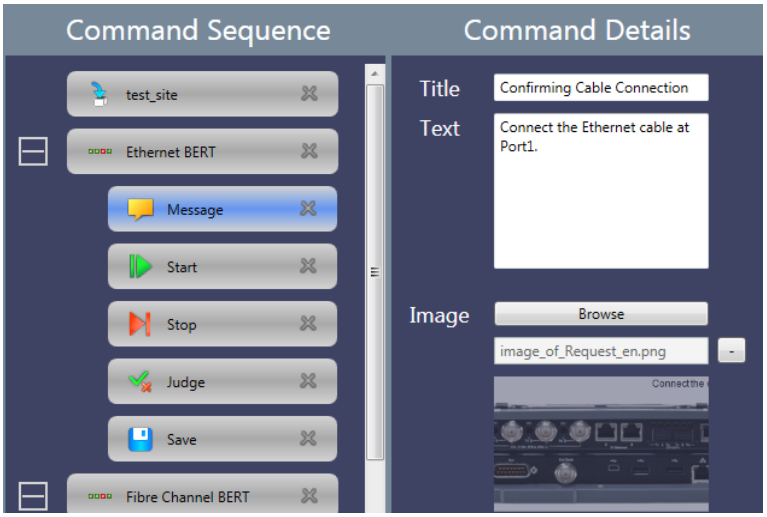


Figure 3.4.7.5-1 Example of Message Settings

To delete the image, click [-].

The following message appears if the scenario has run on the Network Master.

Figure 3.4.7.5-2 Example of Displayed Message

3.4.7.6 Request

Drop Request command in the Command Sequence area to display a variable name on the button. Define parameters to input for Request command. For setting items, refer to 3.4.4 “Global Variables”

When Request is placed at Inner application, the input variable can be referred only from the running applications.
When Request is placed at Outer application, the input variable can be referred from multiple applications.

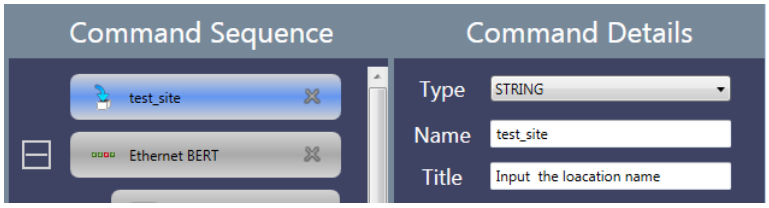


Figure 3.4.7.6-1 Example of Request Settings

In this settings, the following dialog box is displayed on the Network Master.

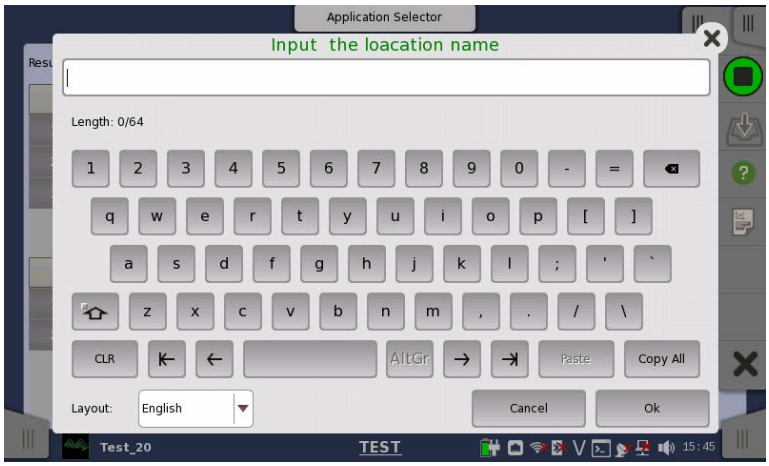
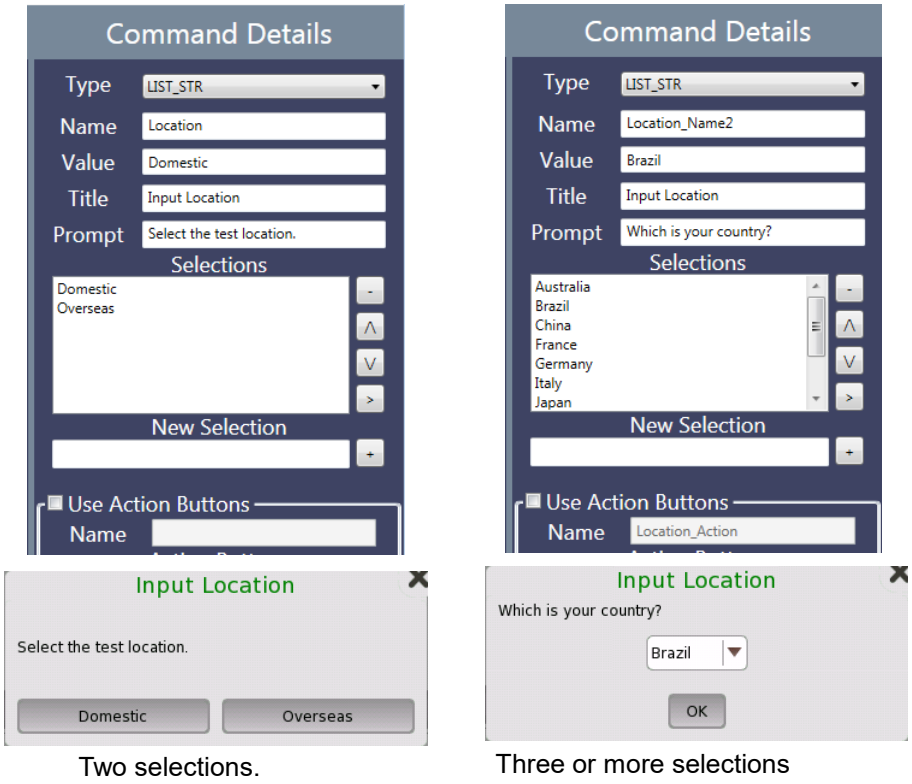


Figure 3.4.7.6-2 Example of Displayed Dialog Box

Selecting LIST_STR for **Type** allows users to edit selections to display in the dialog box.



Two selections. Three or more selections

Figure 3.4.7.6-3 Setting Example and Dialog Boxes When Type is LIST_STR

When there are two selections, two buttons are displayed in the dialog box. Touching one of them closes the dialog box. When there are three or more selections, a combo box is displayed in the dialog box. Touching **OK** closes the dialog box.

Selecting the **Use Action Buttons** check box enables to edit the Action buttons. The dialog box displayed on Network Master is closed by touching one of the Action buttons at the bottom.

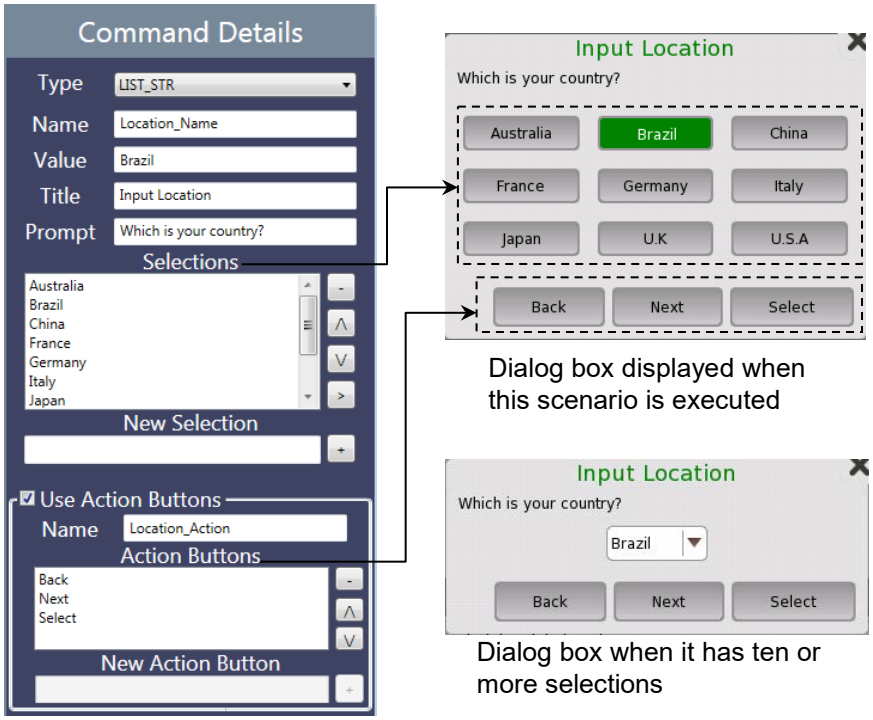


Figure 3.4.7.6-4 Setting Example and Dialog Boxes When Using Action Buttons

Note:

The Action buttons can be set up to three.

3.4.7.7 Action

Set the action against the application.

Set Source MAC: Sets the Source MAC address.
Set Destination MAC: Sets the Destination MAC address.
Set Source IPv4: Sets the Source IPv4 address.
Set Destination IPv4: Sets the Destination IPv4 address.
Custom: Performs the user defined action. Refer to "In case of Custom" on next page.

"Set Source MAC" to "Set Destination IPv4" appear on the specific Ethernet applications.

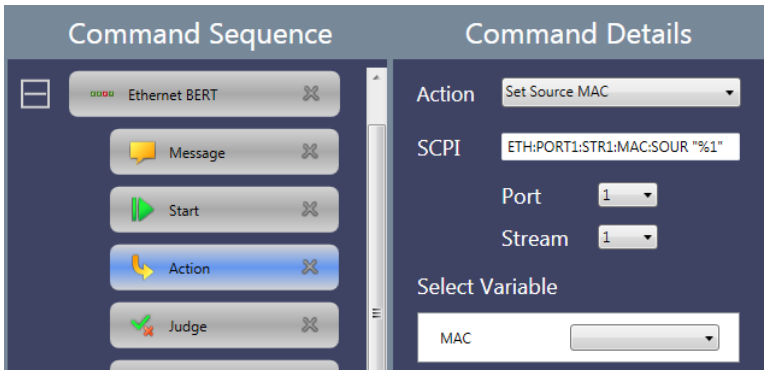


Figure 3.4.7.7-1 Action Details

In case of other than Custom

The SCPI command is displayed in SCPI field.

1. Specify Port and Stream numbers.
2. Select a variable of the address at **Select Variable**.
   If no variables are on the list, define the variable by using **Global Variable** or **Request** command.

Note:

The default timeout value of SCPI command is 30 seconds.

When sending the command which takes more than 30 seconds to receive the response, select **Custom** and then select **Script**. Change timeout value by using TIMEOUT command.

Example of script:

```
TIMEOUT, 60000
EQUAL, "SYST:WAIT:DUR 30"
EQUAL,,"*OPC?"
```

In case of Custom

There are two ways to define the action.

Script: Runs the script written with commands described in Appendix B "Command Reference".

SCPI: Sends a SCPI command written in *MT1000A Network Master Pro MT1100A Network Master Flex Remote Scripting Operation Manual* and *MT1000A Network Master Pro OTDR Modules Remote Scripting Operation Manual* to the Network Master.

Select Script or SCPI.

If selecting SCPI, enter the SCPI command in the SCPI field. If a query command is entered, "FAIL" is returned.

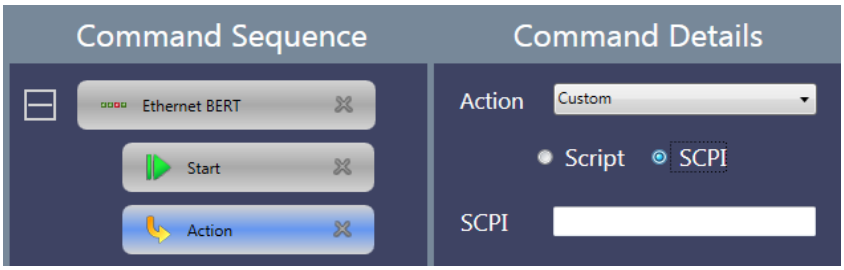


Figure 3.4.7.7-2 Action Details (SCPI)

Entering % and a number as a parameter displays **Select Variable**. Select a variable from defined name in 3.4.4 "Global Variables" or 3.4.7.6 "Request".

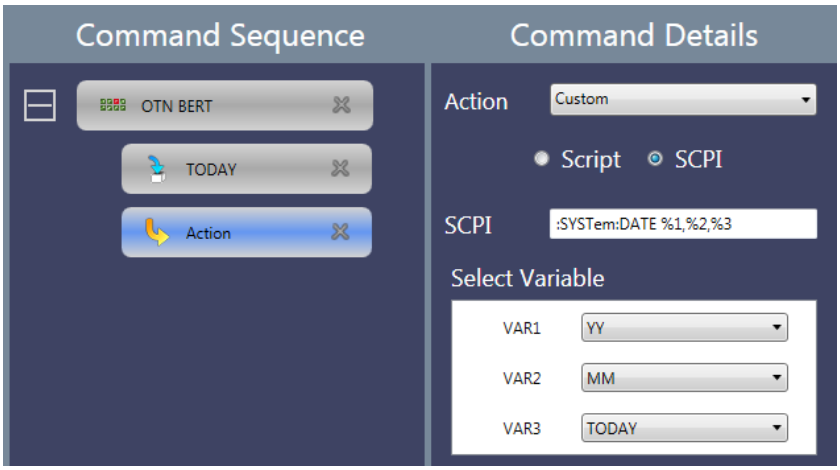


Figure 3.4.7.7-3 Select Variable of SCPI

Figure 3.4.7.7-4 Action Details (Script)

If selecting Script, write the script into the field.
To load the script from the file, click **Import from File**.
To check the script syntax, click **Validate**. For messages of the result, refer to Appendix C “Error Messages”.

3.4.7.8 Judge

Set the method to judge Pass/Fail of the test result.

Summary: Uses the judgement displayed in the status area.
Custom: The method specified by the script or the SCPI command

Note:

If judged as “Fail” by the Judge command, the running scenario stops.

If selecting Custom, select Script or SCPI.
If selecting SCPI, enter the SCPI command in the SCPI field.
If selecting Script, write the script into the field.
To load the script from the file, click **Import from File**.
To check the script syntax, click **Validate**. For messages of the result, refer to Appendix C “Error Messages”.

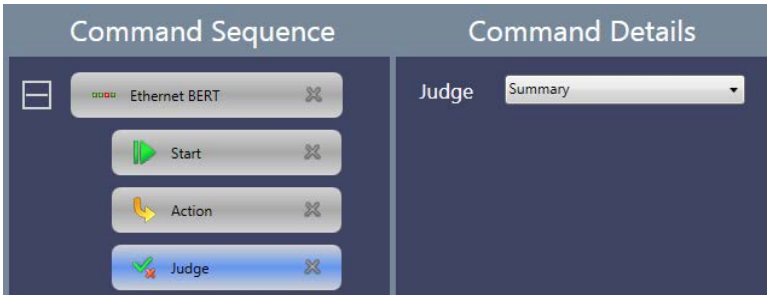


Figure 3.4.7.8-1 Judge Details (Summary)

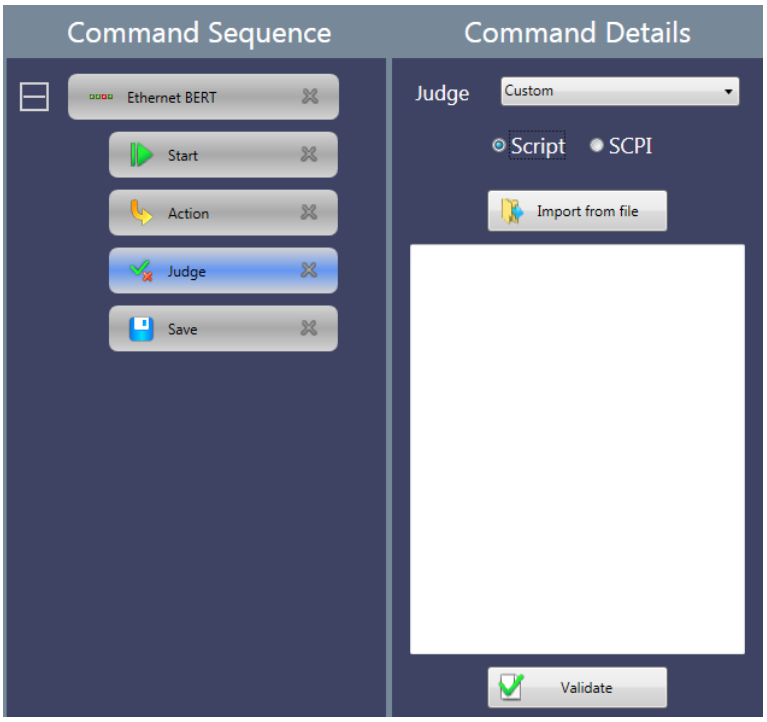


Figure 3.4.7.8-2 Judge Details (Script)

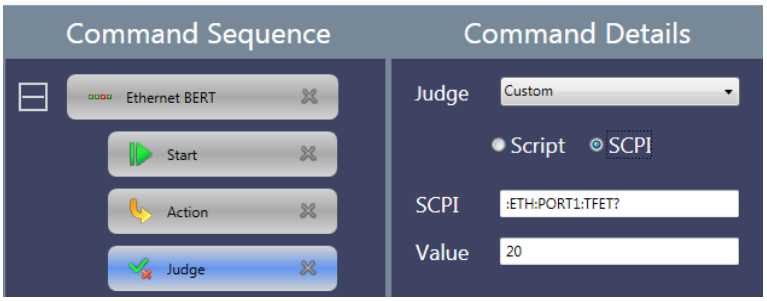


Figure 3.4.7.8-3 Judge Details (SCPI)

Note:

The default timeout of a SCPI command is 30 seconds. To send a SCPI command that takes more than 30 seconds to receive the response, write the SCPI command in the script.

3.4.8 Editing Scenario Settings for MT9083/MT9085

To edit MT9083 Series ACCESS Master settings, click the **Settings** menu, click **Select Platform**, and then select **MT9083/MT9085**.
For descriptions of the setting items, refer to the *MT9083 Series ACCESS Master Operation Manual (M-W3634AE)* or *MT9085 Series ACCESS Master Operation Manual (M-W3971AE)*.

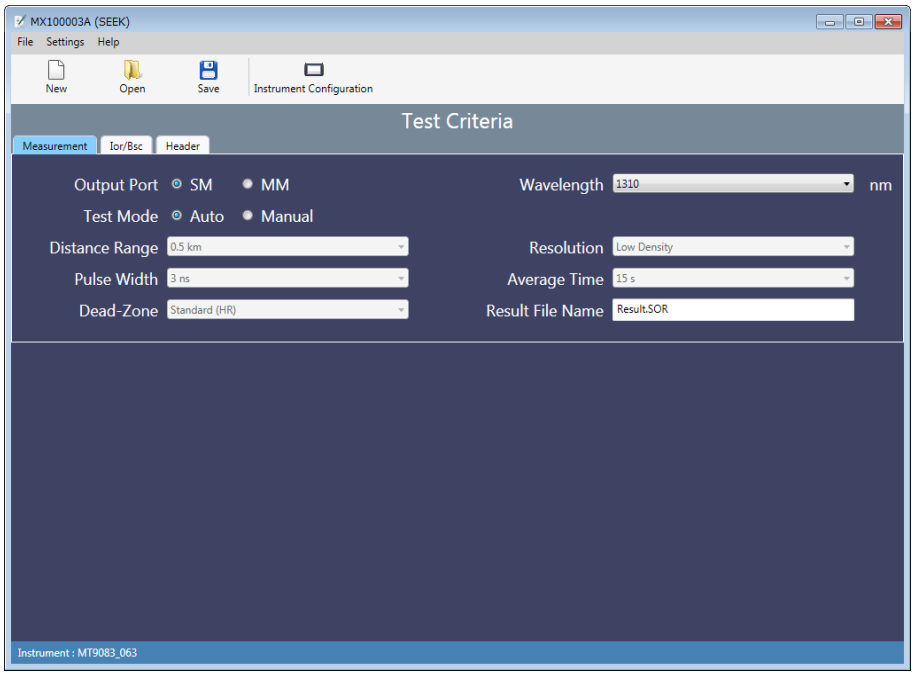


Figure 3.4.8-1 Appearance of the Window When MT9083/MT9085 is Selected for Select Platform

Edited settings are saved as a scenario file (*.acm) for the ACCESS Master. By running a scenario file in Scenario Manager Lite on the ACCESS Master, the ACCESS Master performs Standard OTDR measurement according to the edited settings. Measurement results are saved to a file named as specified in the **Result File Name** field.

3.5 Checking the Scenario Contents

Scenario contents are checked when you try to save the scenario. Clicking **Save** on the toolbar displays the error message if the scenario has error(s).

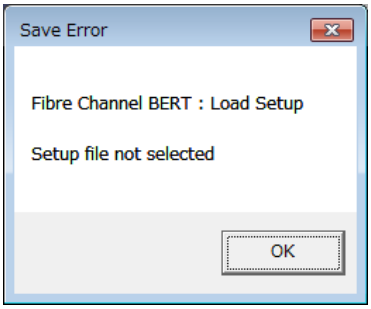


Figure 3.5-1 Error Message Example

Correct the error displayed on the message. The scenario cannot be saved until no more error is detected.

Chapter 4 Running the Scenario

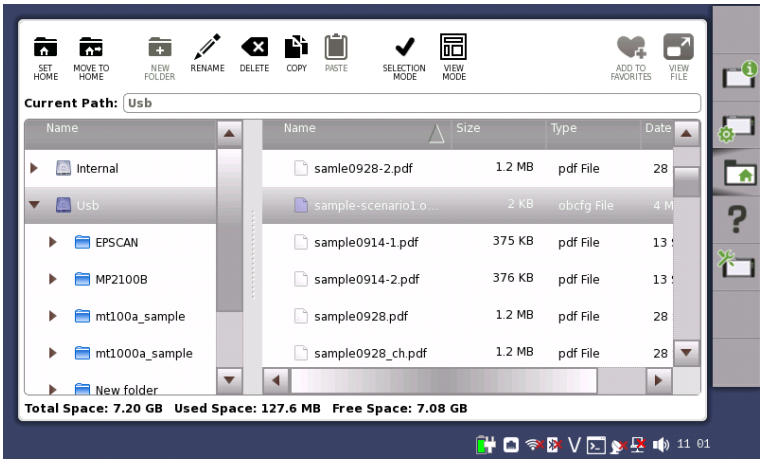
This chapter explains how to run the scenario created by MX100003A on the Network Master.
For how to run scenarios on the MT9083 Series ACCESS Master, refer to Chapter 16 “Scenario Manager Lite” in the *MT9083 Series ACCESS Master Operation Manual (M-W3634AE)*.
For how to run scenarios on the MT9085 Series ACCESS Master, refer to Chapter 13 “Scenario Manager Lite” in the *MT9085 Series ACCESS Master Operation Manual (M-W3971AE)*.

4.1 Copying the Scenario Files

Copy the scenario file(s) created by MX100003A to the storage in the Network Master.

Using USB flash drive

1. Connect a USB flash drive to the PC.
2. Copy the scenario file(s) to the USB flash drive.
3. Unplug the USB flash drive from the PC and connect it to the Network Master.
4. Touch  on the Instrument Toolbar of the Network Master.
5. Touch **Usb** folder and touch the scenario file name.



6. Touch **COPY**  .
7. Touch **Internal** folder, and touch **PASTE**  .

Saving via Ethernet

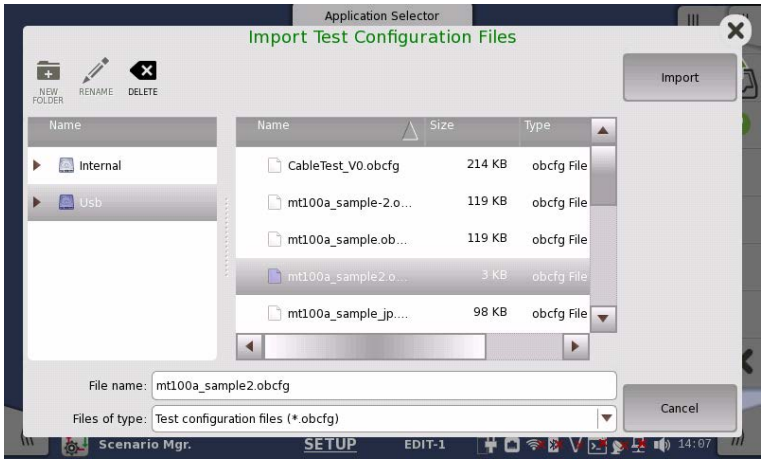
1. Click **File**, **Save To MT1X00A** in the menu when MX100003A is connected to the Network Master via Ethernet.
2. Enter the file name and click **Save**.

4.2 Registering and Running the Scenario

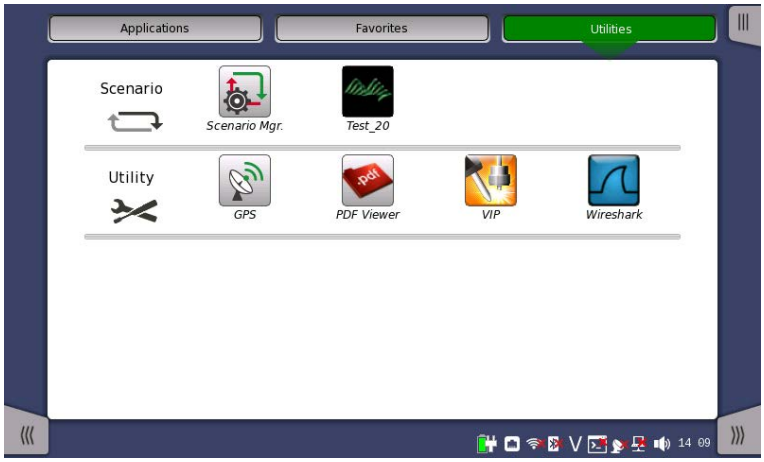
To run the scenario, Register the scenario using Scenario Manager.

4.2.1 Registering the Scenario

1. Display the Utilities screen of the Network Master.
2. Touch **Scenario Mgr.** 
3. Touch  .
4. Select a scenario file and touch **Import**. When the scenario file has been loaded, the content of the scenario is displayed in the list of Scenario Manager.



5. Touch ✕. Confirm that the icon of the loaded scenario appears in the Utilities screen.



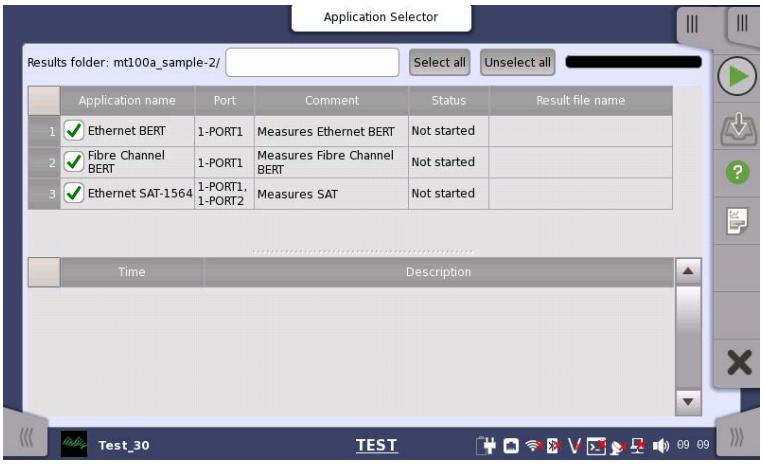
Touching **Hide** on the setup screen of Scenario Manager sets the scenario icon on the Utilities screen to invisible.

4.2.2 Running the Scenario

1. Touch the scenario icon you have registered on the Utilities screen of the Network Master. The destination folder of the result files of scenario appears at **Result folder**. If you wish to create the sub-folder, touch the field and enter the folder name.

   File will be saved in the following folder:
   Internal/Scenario_logs/(scenario name)/(string in the field_date and time_Pass/Fail)

   Example:
   Internal/Scenario_logs/mt100a_sample-2/2016-03-14@10-28-35_Fail



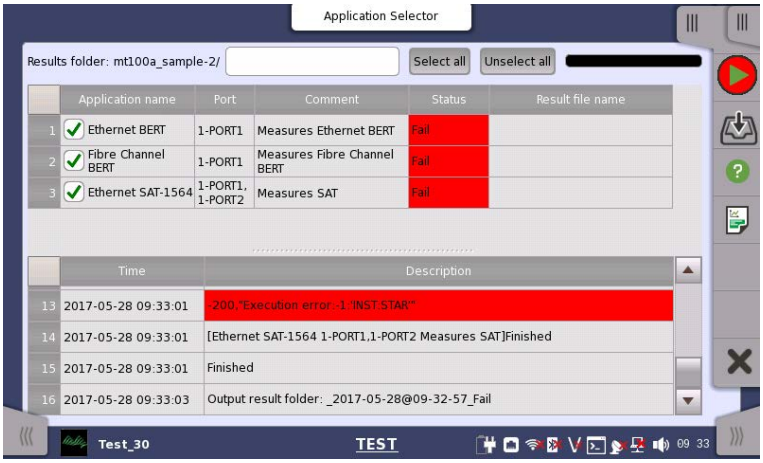
   The destination drive of the result folder can be set when touching an icon on Utilities screen after USB flash drive was connected to the Network Master.

   Even if **Result Folder :Usb** has been set, result files are stored in Internal memory temporary and moved to USB flash drive after the scenario execution has finished.

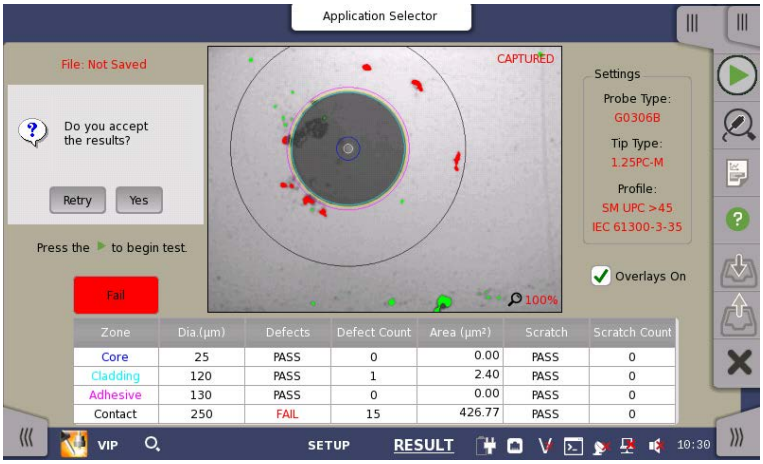
   In cases below, the warning message appears when Network Master has tried to save result files to USB flash drive. If Network Master failed to save result files to USB flash drive, they are stored in Internal memory

   - USB flash drive had been removed before the scenario execution finished.
   - The free space of the USB flash drive is shortage.

2. Select the check box for the application you want to run.
3. Touching  starts running the scenario.
4. Depending on the scenario contents, the panel operation (entering the variable etc.) is required.
   The scenario stops when the each progress of all applications has changed to PASS or FAIL.

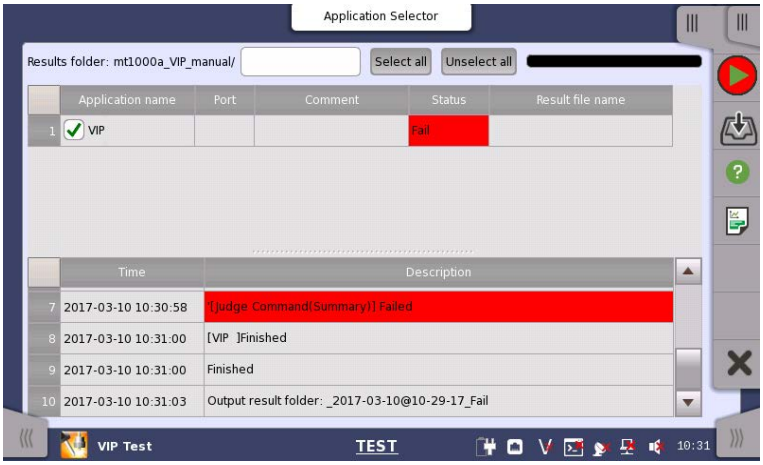


If the VIP scenario has started, the VIP screen appears. In this case, Report, File Save, and File Load icons on Application Toolbar are disabled.



Perform the following operation to close the VIP screen and return to the scenario screen.

- Save the results by touching **Yes** on the above screen.
- Quit the measurement by touching  .

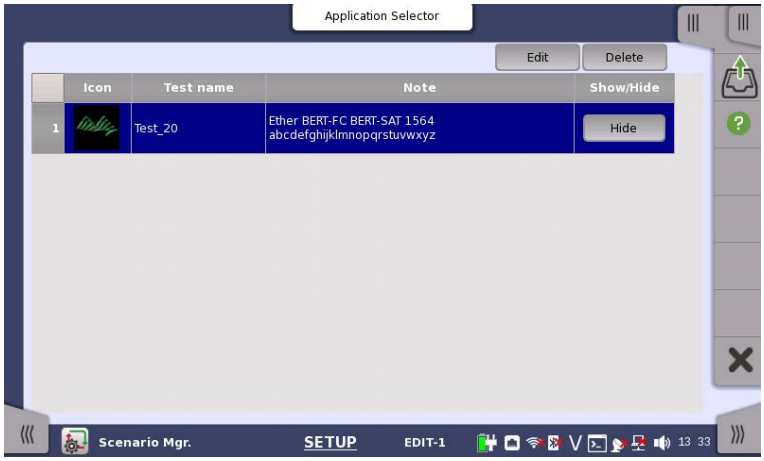


4.2.3 Editing the Scenario

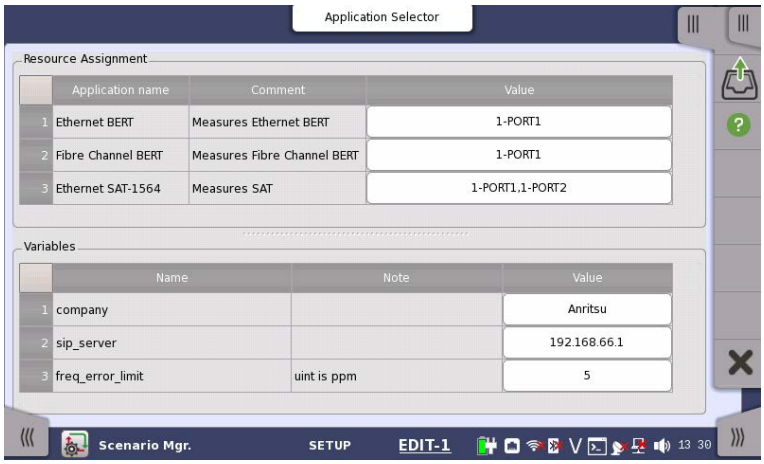
The Scenario Manager of the Network Master can edit the following items:

- Port(s) occupied by the application
- Global variables

1. Display the Utilities screen of the Network Master.
2. Touch **Scenario Mgr.** 
3. Touch the scenario icon to edit.



4. Touch **Edit** or right-bottom tab  .



5. To edit the port(s) the application occupies, touch the field in Value column of Resource Assignment.
6. To edit the variables, touch the field in Value column of Variables.
7. To back to Setup screen, touch the right-bottom tab  .

The warning icon appears at right of Resource Assignment when the port defined in the scenario does not exist. In this case, touch the **Value** field and set the available port(s).

4.2.4 Saving the Scenario

The Scenario can be saved in the following procedure:

1. Display the Utilities screen of the Network Master.
2. Touch **Scenario Mgr.** 
3. Touch the scenario icon to save.
4. Touch **Export**.
5. Enter the file name and touch **Export**.

4.3 Result Files

The result files of the scenario will be saved in the Internal/Scenario_logs folder of the Network Master.
The folder which has the same name as the scenario file will be created, sub-folder will be created according to the time scenario started.

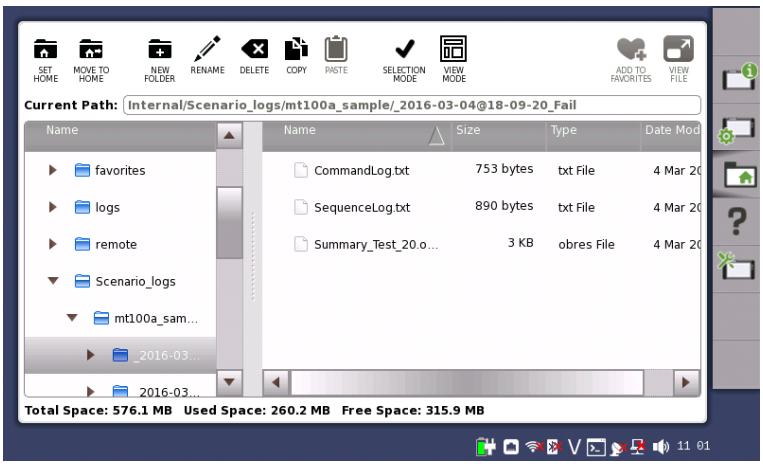


Figure 4.3-1 Result Files and Destination Folder

The following result files will be saved.

CommandLog.txt

The communication log between the Network Master and the scenario. The communication time, SCPI commands, and responses are recorded.

SequenceLog.txt

The log of the messages which were displayed on the Network Master screen.

To see the contents of the file, select the file and touch **VIEW FILE** .

Note:

VIEW FILE cannot display two-byte characters correctly. When two-byte characters are included in the result files, copy the result files to PC to confirm the contents.

Appendix A Syntax of Script

This section explains the syntax of script.

A.1 Elements of the Script

Script used in Action command and Judge command of the scenario is written in text format. The character code of script is UTF-8. To load the script created or edited by using general purpose text editor to MX100003A, be sure to save text file in UTF-8 code.

Elements of script are shown in the figure below.

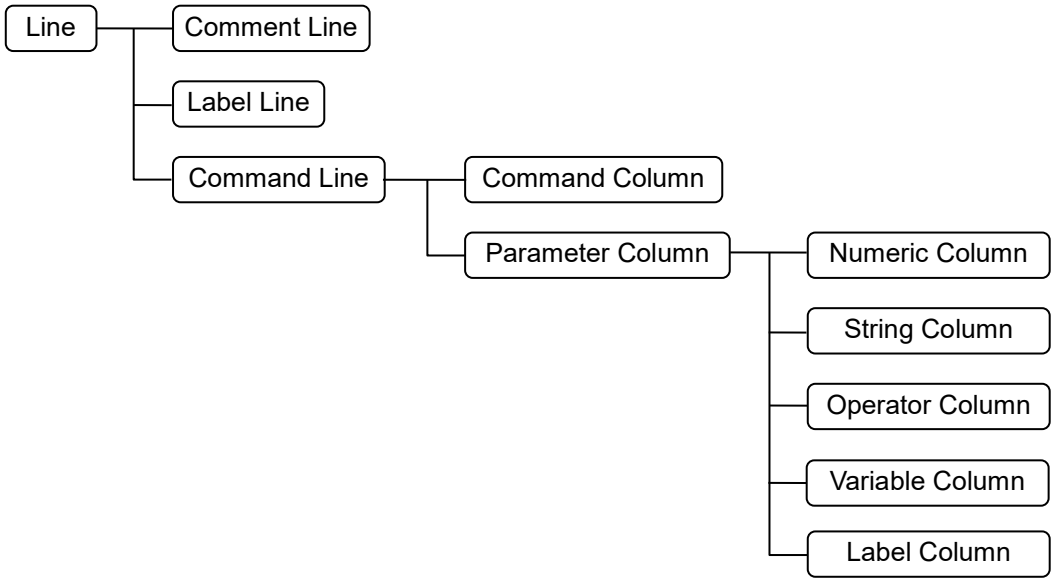


Figure A.1-1 Elements of the Script

Correspondence between example script and elements is shown in the figure below.

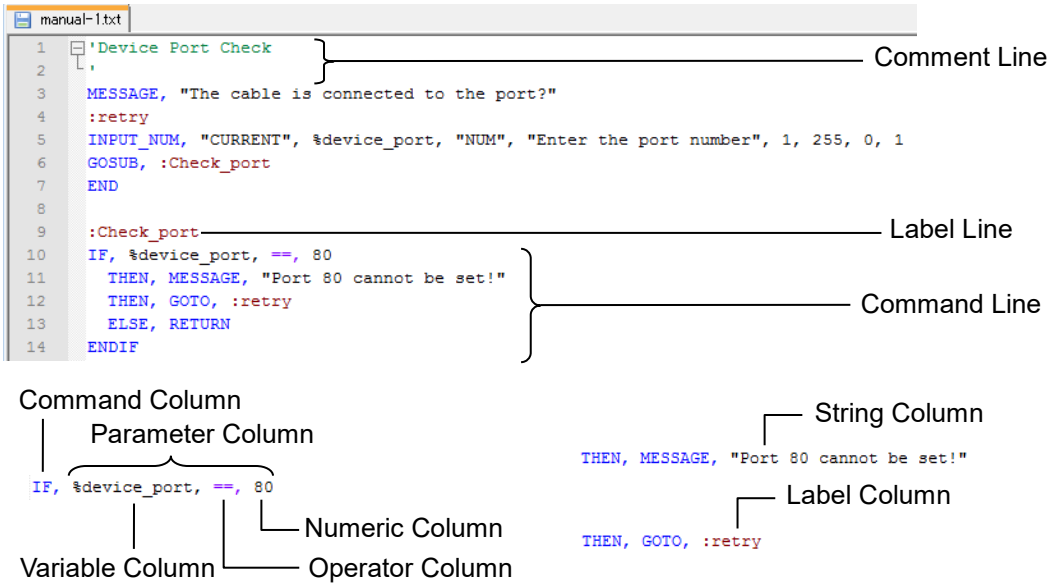


Figure A.1-2 Example Script

A.2 Line

The script syntax consists of one line. The comment, label, or command should be written in one line.
The blank line and the line which contains space only are ignored when running the script.
The line can be distinguished to three types in the following depending on the first character.

Comment Line

The first character of Comment line is single quotation (').
Alphanumeric characters, symbols, two-byte letters are allowed to use in the comment line.
Example:

```
'Script of Ping test to DNS server
```

Label Line

The first character of Label line is colon (:).
Label line is used as the target to jump the process.
Label line is used for the target when jumping the process during running the script.
Alphanumeric characters and under bar (_) are allowed to use in the label name.
Example:

```
:Setup_BERT1
```

Command Line

Command line is consisted of columns which are separated by comma.
The command listed in Appendix B should be written in the first column of the command line.

There are two kinds to Command lines depending on the command used.

- Command line performing the judgement
  Judges pass or fail of the application test result when processing the command line.
  Scenario execution stops when being judged as Fail.
- Command line without performing no judgement
  Does not judge the application test results when the command line has processed.

A.3 Column

The part between commas in the command line is called "Column". The comma at end of the line may be omitted.
The space and tab contained in the column are ignored. However they are not ignored if they are part of string enclosed double quotation (“ “).
There are the following kinds of columns used in the command line.

Command Column

The first column of the command line. Write the command listed in Appendix B.
Example:

```
COPY,CALC,IF,THEN,ELSE,ENDIF,
GOTO,GOSUB,RETURN,END
```

Parameter Column

The second or later column of the command line. Write the parameter(s) of command line.

A.3.1 String Column

In String Column, write the string parameter for the command.
String constant, variable, or both of them are allowed to write in the String Column. Connecting the strings in the String Column is also allowed.

String Constant

String constant is described by string enclosed with double quotation (“ ”).
Example:

```
"*CLS"
"*ESR?"
```

“\” can be used as the escape character. For example, “\n” starts a new line.
The following is the example describing the string double quotations are contained.
Example:

```
"XXXX:YYY \"abcd\"\n Second Line"
```

is regarded as

```
 XXXX:YYY "abcd"
 Second Line
```

String Variable

Variable can be used in the String Column.
The variable is indicated by appending “%” at top of the variable name.
The character in the variable name is case sensitive. For example, %Port and %port are not the same variables.
Alphanumeric characters and under bar (_) are allowed to use in the variable name.
Example:

```
%String,
%Counter_Value,
```

The variables does not have types such as integer or string. The variable can store a numeric value and a string.
The statement of the variable is not required. A variable discovered for the first time during the script execution will be initialized immediately.
The default setting of string is "". If the variable is initialized once, the memory area for the variable is reserved until the script execution ends.
For the variable in the String Column, even if the numeric value is stored in the variable, it is converted to the string and processed.
The converting format from numeric value to string can be specified by using colon. Number of significant figures can be also specified.

```
%variable namd:degit number.degit number under decimal point
```

Example:

```
%Value:6.3
```

This setting is compliant to specifying real number (same as printf function of C language).

If string is stored in the variable, the specified format is ignored.

The following names are reserved for string variable name.

Table A.3.1-1 Reserved Variable Name

Variable Name	Description
%NM_JUDGE_RESULT	Result of Judge command (Summary) *1
%RESULT_FOLDER	File path and folder name which the result file are saved. *1
%RESULT_FOLDER_BASE	Basic part of the folder name which the result file are saved. *1 Date, time, and pass/fail result of the folder name are not included.
%REPORT_SETTING_COMBINE	Setting output of multiple reports.*2 If set to 0, a report is created for each application.
%REPORT_SETTING_CUSTOMER	Customer information in Report Settings*3,*4
%REPORT_SETTING_PROJECT	Project name in Report Settings*3,*4
%REPORT_SETTING_OPERATOR	Operator name in Report Settings*3,*4
%REPORT_SETTING_OPERATOR	Notes in Report Settings*3,*4 When nothing has not been set to %TEST_REPORT_SETTING_NOTES, this parameter value will be output to the test report.
%SUMMARY_REPORT_FILE_NAME	Portion of the report file name*4
%SUMMARY_REPORT_FILE_NAME_HEADER	Header of the report file name*4
%TEST_REPORT_SETTING_NOTES	Notes to be output to the test report *4,*5

*1: Do not set a value to these variables.

*2: Available when the module is MU100020A, MU100021A, MU100022A, or MU100023A.

*3: Applied to reports for all application. When **Use Global Variable** is selected in 3.4.3 “Report Settings”, the variable is displayed in Global Variable dialog box.

*4: These variables are available for the Network Master which software version is 8.01 or later. If Network Master’s version is 8.00 or earlier, values set in 3.4.3 “Report Settings” are output to the report file.

*5: This variable is applied to only application which the variable value has been set.

In case of Figure A.3.1-1, the value stored in the variable are shown below.

%RESULT_FOLDER_BASE ABCD

%RESULT_FOLDER
Internal/Scenario_logs/sample-scenario1/ABCD_2016-03-16@18-01-15_Fail

Figure A.3.1-1 Example of Result Folder Setting

To use the variables defined in 3.4.4 “Global Variable” and 3.4.7.6 “Request”, append the “%” before the variable name. In case of the setting example in Figure A.3.1-2, the variable name will be “%Site_A” in the script for using “Site_A”.

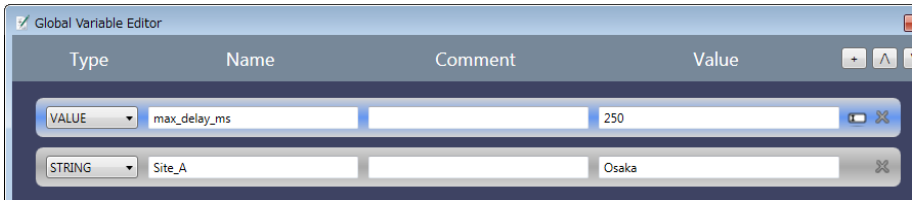


Figure A.3.1-2 Example of Global Variable Setting

If the scenario execution has completed, the following PDF file will be created.
Summary_***.pdf
Using %SUMMARY_REPORT_FILE_NAME enables to set a name to the *** portion.
Example:

```
COPY, %SUMMARY_REPORT_FILE_NAME, "Anritsu"
```

(The name of a created file will be Summary_Anritsu.pdf.)

String Concatenation

Concatenation of string constant, variable, or both of them in String Column is available.

Constants and variables written straight in the column are regarded as a concatenated string.
Example:

```
 "*C" "LS",
```

(Constants are concatenated and treated as “*CLS”)

```
"MKP_CDP" %A,
```

(When the value of variable %A is 10, the concatenated string will be “MKP_CDP 10”)

A.3.2 Numeric Column

In Numeric Column, write the numeric parameter for the command. Numeric constant or variable are allowed to write in the Numeric Column. It is allowed to write only one value in a column. Calculation in Numeric Column is not available.
The range of the numeric constant and variable are shown below.
For integer: –2147483648 to 2147483647
For decimal: Up to 15 digits under the decimal point

Numeric Constant

Numeric value can be written in real number. Digits under the decimal point can be omitted.
Negative value is described by using minus sign at top of digits. If value is less than 1, do not omit a zero (0) before decimal point.
Write (0.25), not .25
Writing 0.25 is correct, but .25 is not correct.
Example:

```
10
-30.0
```

Numeric Variable

Variable can be used in the Numeric Column.
The variable is indicated by appending "%" at top of the variable name.
The character in the variable name is case sensitive. For example, %Port and %port are not same variable.
Alphanumeric characters and under bar (_) are allowed to use in the variable name.
Example:

```
%String,
%Counter_Value,
```

The statement of the variable is not required. A variable discovered for the first time during the script execution will be initialized immediately. The default setting of string is 0. If the variable is initialized once, the memory area for the variable is reserved until the script execution ends. For the variable in the Numeric Column, even if the string is stored in the variable, it is converted to the numeric value automatically at processing the variable. The value is set to 0 for the string which is not able to convert to numeric value.
To use the variables defined in 3.4.4 "Global Variable" and 3.4.7.6 "Request", append the "%" before the variable name. In case of the setting example in Figure A.3.1-2, the variable name will be "%max_delay_ms" in the script for using "max_delay_ms".

A.3.3 Operator Column

The column which one of operator in the following table is written.

Table A.3.3-1 Operator List

Operator	Process	Operator	Process
+	Adding	!=	Not equal
–	Subtraction	<	Greater
*	Multiplication	>	Lesser
/	Division	<=	Equal or greater
&	Bitwise AND	>=	Equal or lesser
|	Bitwise OR	&&	Logical AND
==	Equal	||	Logical OR

A.3.4 Variable Column

Variable Column is used to store a value from the command output.
Usually the processed value or the calculated value is stored to a variable.
Only one variable can be written in the Variable Column.
Writing a numeric constant, a string constant, or an operator etc. occurs the error. Calculation in Variable Column is not available.
Example:

```
%Calc_Value,
```

A.3.5 Label Column

The column which the string of Label line is written. This column is used to write the label of jump target in GOTO command or GOSUB command.
Example:

```
GOTO, :Label
```

The following label name is reserved.

Table A.3.5-1 Reserved Label Name

Label Name	Description
TEST_START_APP	Top position of command sequence in the application

Appendix B Command Reference

This section explains the function, parameter, and example of the commands.

B.1 Command Description Method

Symbols used for command description and how to use them are shown in table below.

Table B.1-1 Symbols Used for Command Description

Symbol	Description
<> []	The string enclosed by angle bracket is a parameter name. The parameter enclosed by square bracket can be omitted.

Commands should be written in capital letters. Using small letters in the command causes a syntax error.

Parameters are described in the following format.

Parameter Name	Type of Column	Description
<result>	Variable Column	Variable used to store the calculation result
<value1>	Numeric/String Column	Value for calculation
<operator>	Operator Column	Operator describing the calculation type
<value2>	Numeric/String Column	Value for calculation

B.2 Command List

Commands are listed in the following tables.

Table B.2-1 List of Commands not Performing the Judgement

Command	Function
CALC	Calculates two values.
COPY	Stores a value to a variable.
ELSE	Performs the process if the latest condition determination result is false.
END	Ends the running application.
ENDIF	Removes the latest result of condition decision.
GOSUB	Jumps to the specified label, recording the line to return.
GOTO	Jumps to the specified label.
IF	Performs a condition decision.
IF_EX	Performs a condition decision from the multiple conditions.
INPUT_LIST	Displays the selection dialog box on the screen and stores the selected value to the variable.
INPUT_LIST_RET	Displays the selection dialog box on the screen and stores the selected button value and string to the variables.
INPUT_NUM	Displays the numeric entry dialog box on the screen and stores the entered value to the variable.
INPUT_STR	Displays the string entry dialog box on the screen and stores the entered value to the variable.
LOG	Outputs string in the message table on the Network Master screen.
MESSAGE	Displays the message box on the screen.
MID	Retrieves characters in a specified range from the origin string.
REMOVE	Removes the specified string from the original string.
RETURN	Jumps process to the line for return which is recorded in most recently.
SPLIT	Takes out the string of the position specified by the number from the comma separated string.
START_APP	Can be used in a script of Custom Application and starts another application.
STR_LEN	Takes out the string length.
STR2VAL	Converts character strings into numbers.
THEN	Performs the process if the latest condition determination result is true.
TIMEOUT	Sets the timeout of the communication with the Network Master.
WAIT	Waits for specified time.

Table B.2-2 List of Commands Performing the Judgement

Command	Function
DLG_OK	Displays the received response on the message box which has OK button.
DLG_YESNO	Displays the received response on the message box which has YES and NO buttons.
EQUAL	Tests whether the response matches the expected value.
IM_COPY	Loads characters from the specified file and assigns them to variables.
JUDGE_FAIL	Judges the test result as Fail intentionally.
NOT_EQUAL	Tests whether the response does not match the undesirable value.
VAR_STORE	Stores the response to a variable.

B.3 Commands not Performing the Judgement

This section explains the commands which does not perform the Pass or Fail judgement of the test result.

CALC

Function

Calculates two values.

Syntax

```
CALC, <result>, <value1>, <operator>, <value2>
```

Parameters

<result>	Variable Column	Variable used to store the calculation result
<value1>	Numeric/String Column	Value for calculation
<operator>	Operator Column	An operator which indicates calculation type
<value2>	Numeric/String Column	Value for calculation

Description

Performs the specified calculation of <value1> and <value2> by <operator>, stores the result to <result>.
The calculation is performed according to the format of <value1> and <value2>.
Available operators are:

Four arithmetic operations:
Available for the calculation of numeric values. Not available for the string.
Adding (+), Subtraction (–), Multiplication (*), Division (/)
Bit operation:
Available for the calculation of integers. Not available for the real number and string.
Bitwise AND (&), Bitwise OR (|)
Comparison operation:
Equal (==), Not equal (!=), Greater (<), Lesser (>), Equal or greater (<=), Equal or lesser (>=)
Returns 1 if comparison result is true. Returns 0 if the result is false.
Logical operation:
Available for the calculation of integers. Not available for the real number and string.
Logical AND (&&), Logical OR (||)
Returns 1 if comparison result is true. Returns 0 if the result is false.

Example of Use

```
CALC, %Value, 10, +, 20
CALC, %Value, %Value, -, 1
CALC, %Value, %A, &, %B
```

COPY

Function

Stores a value to a variable.

Syntax

```
COPY, <variable>, <value>
```

Parameters

<variable>	Variable Column	The Variable for storing the value
<value>	Numeric/String Column	Value to be stored to the variable

Example of Use

```
COPY, %Value,  0
COPY, %A,  %B
```

ELSE

Function

Performs the process if the latest condition determination result is false.

Syntax

```
ELSE, <command>
```

Parameter

<command>	Command	Command to perform when determination result is false.

Description

Processes the command in the next column of ELSE if the latest condition decision result in the stack is false.
It is not necessary to write ELSE command in next line of IF or THEN command.
Performing ELSE command depends on the result in the stack, not where IF command is written.
The commands excluding IF, IF_EX, THEN, ELSE, and ENDIF are available in the next column of THEN command.
When the stack of the condition decision result is empty, performing ELSE command causes an execution error.

Example of Use

```
IF, %Value, ==, 100
ELSE, LOG, "%Value is not 100"
ELSE, JUDGE_FAIL
ENDIF
```

END

Function

Ends the running application.

Syntax

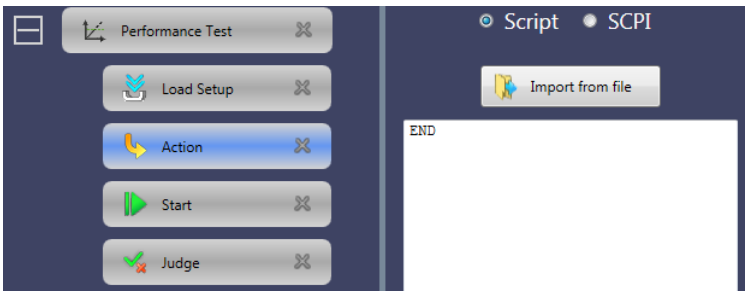
```
END
```

Parameter

None

Description

Application process ends automatically if the process of the final command in a command sequence has been performed. This command is used to end the process of the application expressly.
In the following scenario, if END in the script of Action command has been performed, the Performance Test application ends without performing Start and Judge commands.



Example of Use

```
END
```

ENDIF

Function

Removes the latest result of condition decision.

Syntax

```
ENDIF
```

Parameter

None

Description

Pops the result of the latest condition decision which has performed by IF command from the stack.
This command should be written after THEN and ELSE command which are continued from IF command.
It is not necessary to write THEN and ELSE command in next line of IF or ELSE command.
Performing THEN and ELSE command depends on the result in the stack, not where IF command is written.
Only the result in the stack decides whether perform THEN and ELSE command. The position of IF command does not effect performing THEN and ELSE command.
When the stack of the condition decision result is empty, performing ENDIF command causes an execution error.

Example of Use

```
IF, %Value, ==, 100
THEN, LOG, "Value is 100"
ELSE, LOG, "Value is not 100"
ENDIF
```

GOSUB

Function

Jumps to the specified label, recording the line to return.

Syntax

```
GOSUB,  <label>
```

Parameter

<label>	Label Column	Label of jump destination

Description

This command jumps the line to process to the line specified by <label>.
Using this command is able to return to the line next to GOSUB command by using RETURN command.
The line number for return is pushed to the stack by using the GOSUB command. Use GOTO command if the process does not need to return.

Example of Use

```
GOSUB, :Sub
```

GOTO

Function

Jumps to the specified label.

Syntax

```
GOTO, <label>
```

Parameter

<label>	Label Column	Label of jump destination

Description

This command jumps the line to process to the line specified by <label>.
It is not possible to return the process by using RETURN command because GOTO command does not record the line for return.
Use GOSUB and GOTO command properly according to your needs.

Example of Use

```
GOTO, :Next
```

IF

Function

Performs a condition decision.

Syntax

```
IF, <value1>, <operator>, <value2>
```

Parameters

<value1>	Numeric/String Column	Value used for the decision
<operator>	Operator Column	An operator which indicates calculation type
<value2>	Numeric/String Column	Value used for the decision

Description

Performs the specified calculation of <value1> and <value2> by <operator>, pushes the result to the decision result stack.
After this command, the process can be switched by using THEN and ELSE command.
This command is usually used with pair to ENDIF command.

The calculation is performed according to the format of <value1> and <value2>.
Available operators are:

Four arithmetic operations:
Available for the calculation of numeric values. Not available for the string.
Adding (+), Subtraction (–), Multiplication (*), Division (/)
Bit operation:
Available for the calculation of integers. Not available for the real number and string.
Bitwise AND (&), Bitwise OR (|)
Comparison operation:
Equal (==), Not equal (!=), Greater (<), Lesser (>), Equal or greater (<=), Equal or lesser (>=)
Returns 1 if comparison result is true. Returns 0 if the result is false.
Logical operation:
Available for the calculation of integers. Not available for the real number and string.
Logical AND (&&), Logical OR (||)
Returns 1 if comparison result is true. Returns 0 if the result is false.

Example of Use

```
IF, %Value, ==, 100
THEN, LOG, "Value is 100"
ELSE, LOG, "Value is not 100"
ENDIF
```

IF_EX

Function

Performs a condition decision from the multiple conditions.

Syntax

```
IF_EX, <condition>
```

Parameter

<condition>	String Column	String which conditions are described

Description

Use this command if deciding the multiple conditions. Use IF command to decide the single condition.
Variables can be written into the string.
After this command, the process can be switched by using THEN and ELSE command.
This command is usually used with pair to ENDIF command.
Available operators are the same as those of IF command basically. For this command, "%" is also available as remainder calculation in Four arithmetic operations.

Example of Use

```
IF_EX, "( %ABC >= 10 ) && ( %ABC <= 20 )"
IF_EX, "( %ABC % 2 ) == 0"
```

INPUT_LIST

Function

Displays the selection dialog box on the screen and stores the selected value to the variable.
This command performs the same process as Request command in 3.4.7.6 "Request". Variable is treated as string.

Syntax

```
INPUT_LIST, <scope>, <result>, <title>, <option>[, <text>][,
<default>]
```

Parameters

<scope>	String Column	Scope of the variable CURRENT Local variable which is available only in the current script Global Global variable which is accessible from all of scripts
<result>	Variable Column	Variable which stores the selected value
<title>	String Column	String to display on the title bar of the dialog box
<option>	String Column	String of options separated by comma
<text>	String Column	String to display on the dialog box (can be omitted)
<default>	String Column	Default value of option (can be omitted) If omitted, the first option is selected.

Note:

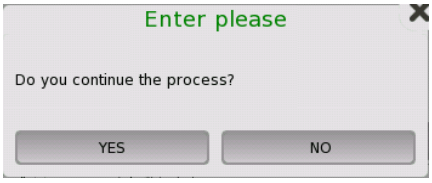
When number of the options is two or less, default value is not displayed on the dialog box.

Description

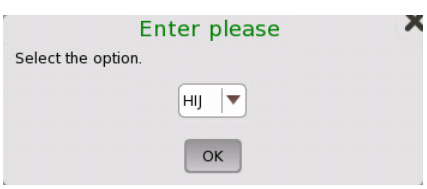
Touching the button closes the selection dialog box.
Script process pauses until the selection dialog box is closed.

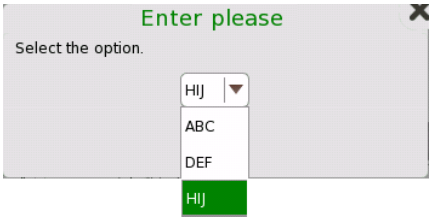
Example of Use

```
INPUT_LIST, "CURRENT", %ABC, "Enter please", "YES, NO", "Do
you continue the process?"
```




```
INPUT_LIST, "CURRENT", %ABC, "Enter please ", "ABC, DEF, HIJ",
"Select the option.", "HIJ"
```





INPUT_LIST_RET

Function

Displays the selection dialog box on the screen and stores the selected value to the variable.
This command performs the same process as when the **Use Action Buttons** checkbox is selected on the Command Details window in 3.4.7.6 “Request”. The variable is treated as a string.

Syntax

```
INPUT_LIST_RET, <scope>, <actionName>, <actionList>,
<result>, <title>, <option>[, <text>][, <default>]
```

Parameters

<scope>	String Column	Scope of the variable CURRENT Local variable which is available only in the current script GLOBAL Global variable which is accessible from all of scripts
<actionName>	Variable Column	Variable which stores the selected decision button value
<actionList>	String Column	String of Action buttons separated by commas
<result>	Variable Column	Variable which stores the selected option value
<title>	String Column	String to display on the title bar of the dialog box
<option>	String Column	String of options separated by commas
<text>	String Column	String to display on the dialog box (can be omitted)
<default>	String Column	Default value of option (can be omitted) If omitted, the first option is selected.

Notes:

When the number of options is ten or more, the options are displayed in list format.

The Action buttons can be set up to three. When four or more Action button names are set, the first three buttons appear in the dialog box.

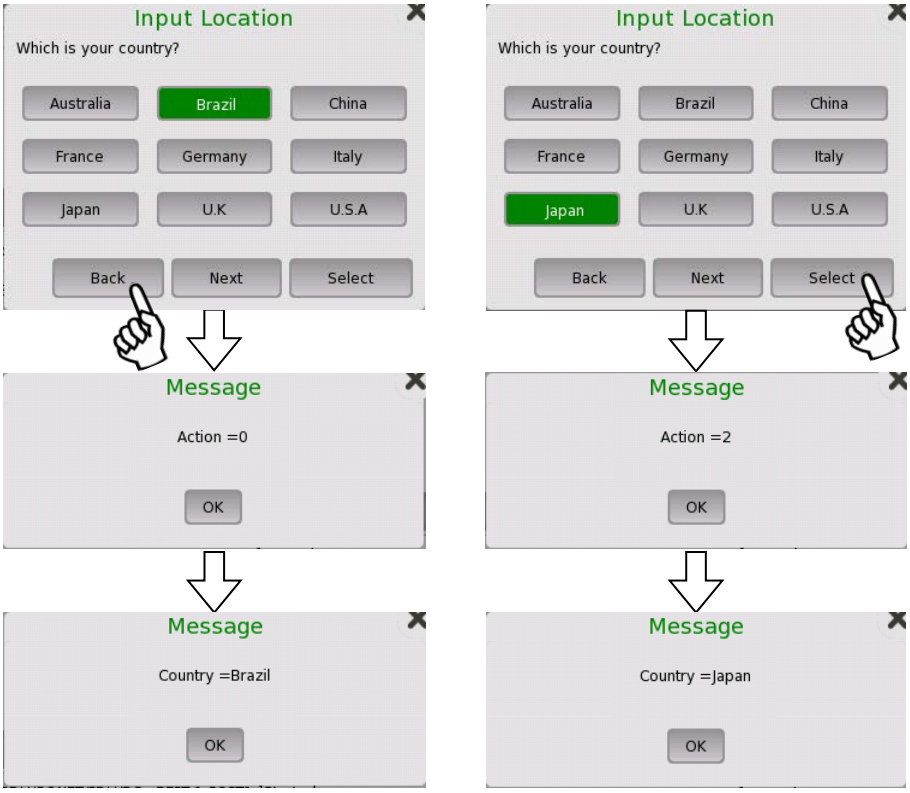
Description

Touching an Action button closes the selection dialog box.

Script process pauses until the selection dialog box is closed.

Example of Use

```
INPUT_LIST_RET, "CURRENT", %action,
"Back,Next,Select", %country, "Input Location",
"Australia,Brazil,China,France,Germany,Italy,Japan,U.K,U
.S.A", "Which is your country?", "Brazil"
MESSAGE, "Action =" %action
MESSAGE, "Country =" %country
```



INPUT_NUM

Function

Displays the numeric entry dialog box on the screen and stores the entered value to the variable.
This command perform the same process as Request command in 3.4.7.6 “Request”. Variable is treated as numeric value.

Syntax

```
INPUT_NUM, <scope>, <result>, <type>, <title>, <min>, <max>,
<dec>[, <default>]
```

Parameters

<scope>	String Column	Scope of the variable CURRENT Local variable which is available only in the current script GLOBAL Global variable which is accessible from all of scripts
<result>	Variable Column	Variable which stores the entered value
<type>	String Column	Display type of the numeric entry dialog box NUM Decimal HEX Hexadecimal BIN Binary
<title>	String Column	Text to display on the title bar of the dialog box
<min>	Numeric Column	Minimum value to display on the dialog box
<max>	Numeric Column	Maximum value to display on the dialog box
<dec>	Numeric Column	Digits number under the decimal point 0 to 15
<default>	Numeric Column	Default value to display on the dialog box (can be omitted) If omitted, minimum value is selected.

Note:

Even if hexadecimal or binary is specified, the value is managed in decimal format in inside. Due to this reason, for example, comparison in IF command is performed in decimal format when using the variable.

Description

Displays the string specified by <message> on the message box. Touching the OK button closes the numeric entry dialog box.
Script process pauses until the numeric entry dialog box is closed.

Example of Use

```
INPUT_NUM, "CURRENT", %ABC, "HEX", "Enter please", 0, 255,
0, 4
```

INPUT_STR

Function

Displays the string entry dialog box on the screen and stores the entered value to the variable.
This command perform the same process as Request command in 3.4.7.6 “Request”. Variable is treated as string.

Syntax

```
INPUT_STR, <scope>, <result>, <type>, <title>[, <default>]
```

Parameters

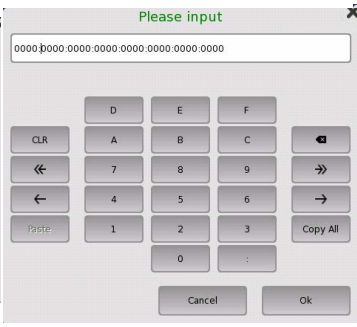
<scope>	String Column	Scope of the variable CURRENT Local variable which is available only in the current script GLOBAL Global variable which is accessible from all of scripts
<result>	Variable Column	Variable which stores the entered value
<type>	String Column	Display type of the string entry dialog box IPV4 IPv4 address format IPV6 IPv6 address format MAC MAC address format STRING string
<title>	String Column	Text to display on the title bar of the dialog box
<default>	String Column	Default string to display on the dialog box (can be omitted) If omitted, blank or 0 is selected.

Description

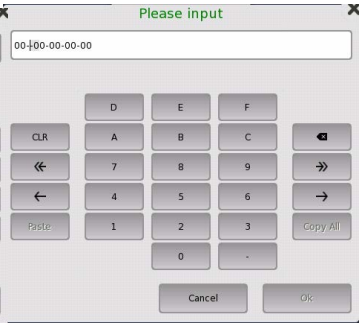
Displays the string specified by <message> on the message box.
Touching the OK button closes the dialog box.
Script process pauses until the dialog box is closed.
<type>

IPV4

IPV6



MAC



STRING

Example of Use

```
INPUT_STR, "CURRENT", %ABC, "IPv4", "Please Input"
INPUT_STR, "CURRENT", %ABC, "IPv6", "Please Input"
INPUT_STR, "CURRENT", %ABC, "MAC", "Please Input"
INPUT_STR, "CURRENT", %ABC, "STRING", "Please Input", "ABC"
```

LOG

Function

Outputs string in the message table on the Network Master screen.

Syntax

```
LOG, <message>[, <color>]
```

Parameters

<message>	String Column	String to output into log
<color>	String Column	Color of the output string on the log “GREEN”, ”RED”, or ”YELLOW”are available. If omitted, normal color (gray) is applied.

Description

Outputs string specified by <message> into message table as log.

This command does not process the remote control to Network Master.

Example of Use

```
LOG, "ABC"
LOG, "ABC", "GREEN"
COPY, %Value, -20
LOG, "ABC= " %Value " [dB] "
```

MESSAGE

Function

Displays the message box on the screen.

This command performs the same process as Message command in 3.4.7.5 “Message”

Syntax

```
MESSAGE,  <message>
```

Parameter

<message>	String Column	String to display on the message box

Description

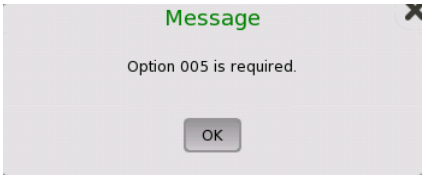
Displays the string specified by <message> on the message box.

Touching the OK button closes the message box.

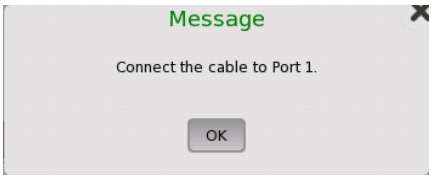
Script process pauses until the message box is closed.

Example of Use

```
MESSAGE, "Option 005 is required."
```




```
MESSAGE, "Connect the cable to Port 1."
```



MID

Function

Returns a specified number of characters from a string.

Syntax

```
MID, <result>, <source>, <match>[, <count>]
```

Parameter

<result>	Variable Column	Variable which stores the result after the specified string is obtained
<source>	String / Variable Column	Original string
<position>	Numeric Column	Value specifying the starting string position (begins from 1)
<count>	Numeric Column	The number of characters to return

Description

Obtains the number of characters specified by <count> from the <position> -th position string in <source> and stores it in <result>.
If <count> is omitted, characters from <position> -th to end are stored.

Example of Use

```
MID, %power, "12.34dBm", 1, 5
MID, %unit, "12.34dBm", 6
LOG, "Power= " %power
LOG, "Unit=" %unit
'"-12.34" is stored into %power.
'"dBm" is stored into %unit.
```

REMOVE

Function

Removes the specified string from the original string.

Syntax

```
REMOVE, <type>, <result>, <source>, <match>
```

Parameter

<type>	String Column	String format Write “SIMPLE” in this column.
<result>	Variable Column	Variable which stores the result after the specified string is removed
<source>	String / Variable Column	Original string
<match>	String / Variable Column	String to be removed

Description

Removes a string specified by <match> from the <source> string and stores into <result>.

Example of Use

```
COPY, %power, "-12.34dBm"
REMOVE, "SIMPLE", %power, %power, "dBm"
LOG, "Power= " %power
'"-12.34" is stored into %power.
```

RETURN

Function

Jumps process to the line for return which is recorded in most recently.

Syntax

```
RETURN
```

Parameter

None

Description

Pops the line for return from the stack recorded by GOSUB command and jumps to the next line of GOSUB command.
When the stack of the line for return is empty, performing RETURN command causes an execution error.

Example of Use

```
:Error
   IF, %code, !=, 0
   THEN, MESSAGE, "Error code ="%code
   ENDIF
RETURN
```

SPLIT

Function

Splits a given string to multiple strings with comma and takes out one string among them.
Brackets “(” and “)” contained in the given string are eliminated in this process.

This command is useful when storing a measurement value to a numeric variable from the string obtained by a query which returns measurement result, for example, a SCPI command "ETH:PORT1:IFET? (A,B,C)".

Syntax

```
SPLIT, <result>, <position>, <source>
```

Parameter

<result>	Variable Column	Variable which stores the entered value
<position>	Numeric Column	Value specifying the string position
<source>	String Column	Comma separated string

Description

Stores the <position> -th position string in the multiple strings obtained by splitting string <source> with comma to <result>.
Specifying 0 to <position> stores the number of strings separated by comma to <result>.

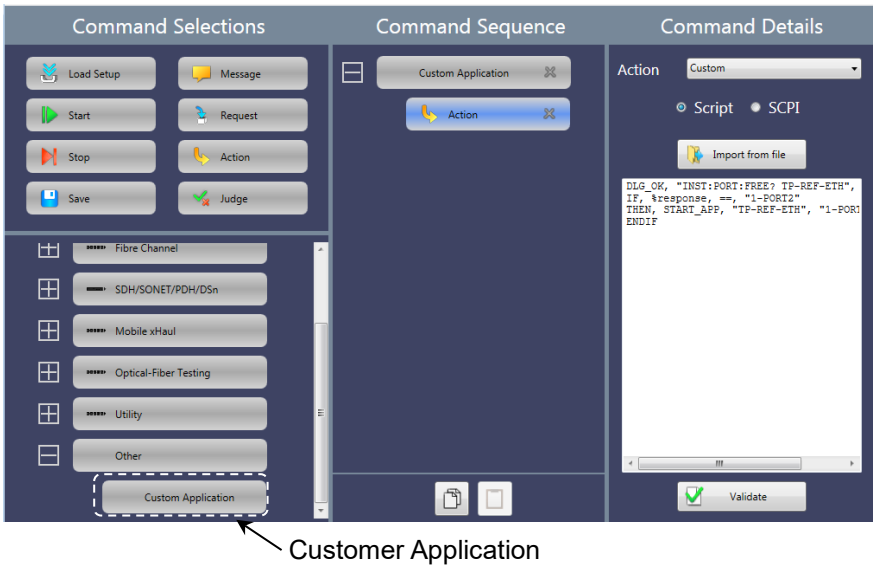
Example of Use

```
SPLIT, %A, 1, "1.00, 2.00, 3.00"
' "1.00" is stored to %A
SPLIT, %A, 2, "ABC, DEF, HIJ, KLM"
' "DEF" is stored to %A
SPLIT, %A, 1, "(1.00, 2.00, 3.00)"
' "1.00" is stored to %A
SPLIT, %A, 0, "(1.00, 2.00, 3.00)"
' 3 (numeric value) is stored to %A
```

START_APP

Function

Starts another application by a script of Custom Application.
The scripts of other applications cannot be used.



Syntax

```
START_APP, <application>, <port>
```

Parameter

<application>	String Column	Name of application to start. Specifies the words in the <application> column of Table B.3-1.
<port>	String Column	Number of port that the application uses. If multiple ports are used, separate their numbers by commas.

For the details of port numbers, refer to 1.6.4 “Port Number (Logical Port)” in the *MT1000A Network Master Pro MT1100A Network Master Flex Remote Scripting Operation Manual*.

Note:

When setting a port number for <port>, make sure to select a port unused by application.

Table B.3-1 Application Parameters

<application>	Application
TP-APS-OTN	OTN, APS
TP-APS-SDHPDH	SDH/SONET/PDH/DSn, APS
TP-APS-SDHPDH-OTN	SDH/SONET/PDH/DSn, APS, Launch with + OTN

Table B.3-1 Application Parameters (Cont'd)

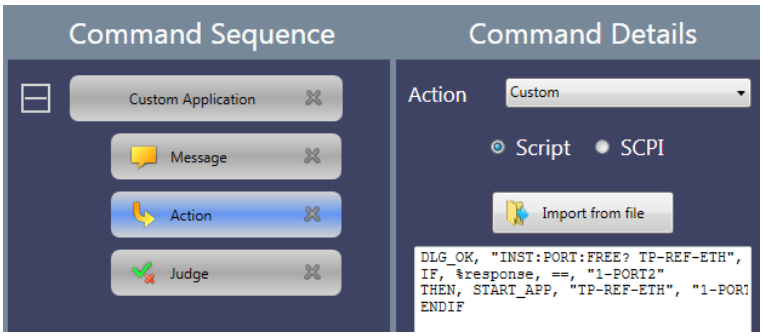
<application>	Application
TP-BERT-CPRI	Mobile xHaul, CPRI/OBSAI BERT
TP-BERT-CPRI-OTN	Mobile xHaul, CPRI/OBSAI BERT, Launch with + OTN
TP-BERT-ETH	Ethernet, BERT
TP-BERT-ETH-OTN	Ethernet, BERT, Launch with + OTN
TP-BERT-FC	Fibre Channel, BERT
TP-BERT-FC-OTN	Fibre Channel, BERT, Launch with + OTN
TP-BERT-OTN	OTN, BERT
TP-BERT-ROE	Mobile xHaul, eCPRI/RoE BERT
TP-BERT-SDHPDH	SDH/SONET/PDH/DSn, BERT
TP-BERT-SDHPDH-OTN	SDH/SONET/PDH/DSn, BERT, Launch with + OTN
TP-CABLE-ETH	Ethernet, Cable Test
TP-CHSTAT-ETH	Ethernet, Chanel Stats
TP-DISC-ETH	Ethernet, Discovery
TP-MONGEN-ETH	Ethernet, Monitor/Generator
TP-MONGEN-ETH-OTN	Ethernet, Monitor/Generator, Launch with + OTN
TP-NOFRAME-DEVICE	Device Test, No Frame
TP-PASS-CPRI	Mobile xHaul, CPRI P.Thru
TP-PASS-ETH	Ethernet, Pass Through
TP-PERF-FC	Fibre Channel, Performance Test
TP-PERF-FC-OTN	Fibre Channel, Performance Test, Launch with + OTN
TP-PING-ETH	Ethernet, Ping
TP-REFL-ETH	Ethernet, Reflector
TP-REFL-ETH-OTN	Ethernet, Reflector, Launch with + OTN
TP-REFL-FC	Fibre Channel, Reflector
TP-REFL-FC-OTN	Fibre Channel, Reflector, Launch with + OTN
TP-RFC-ETH	Ethernet, RFC 2544
TP-RFC-ETH-OTN	Ethernet, RFC 2544, Launch with + OTN
TP-RFC6349-ETH	Ethernet, RFC 6349
TP-RTD-OTN	OTN, RTD
TP-RTD-SDHPDH	SDH/SONET/PDH/DSn, RTD
TP-RTD-SDHPDH-OTN	SDH/SONET/PDH/DSn, RTD, Launch with + OTN
TP-SAT-ETH	Ethernet, SAT 1564
TP-SAT-ETH-OTN	Ethernet, SAT 1564, Launch with + OTN
TP-SYNCTEST-ETH	Ethernet, Sync Test
TP-TRACE-ETH	Ethernet, Trace Route
OTDR-OTDR	OTDR, Standard OTDR
OTDR-OLTS	OTDR, OLTS

Description

Assign the port number specified for <port> and start the application. This command can be used only once in a script.

When the script ends, the application started by the START_APP command quits automatically. In the following figure, the Action

command starts the Reflector application of Ethernet. When the Action command ends, the Reflector application also ends.
For the Judge command which follows the Action command, the START_APP command can be used as a script.



Example of Use

```
START_APP, "TP-BERT-OTN", "2-PORT2"
START_APP, "TP-APS-OTN", "1-PORT2,1-PORT1"
```

STR_LEN

Function

Takes out the string length.

Syntax

```
STR_LEN, <result>, <source>
```

Parameter

<result>	Variable Column	Variable which stores the string length
<source>	String/ Variable Column	String

Description

Stores the string length of <source> to <result>.

Example of Use

```
STR_LEN, %count, "12.34dBm"
LOG, "Count=" %count
' 8 is stored to %count
```

STR2VAL

Function

Converts character strings into numbers.
Character strings in octal, decimal, hexadecimal, and binary formats can be converted.

Syntax

```
STR2VAL, <variable>, <value>
```

Parameter

<variable>	Variable Column	Variable to store converted values.
<value>	String Column	Character strings to be converted into numbers.

Description

Convert a character string of <value> into numbers and store the data in <variable>.

Example of Use

For decimal numbers:

```
STR2VAL, %val, "165"
LOG, "Value=" %val
```

For hexadecimal numbers:

```
STR2VAL, %val, "#HA5"
LOG, "Value=" %val
```

For octal numbers:

```
STR2VAL, %val, "#Q245"
LOG, "Value=" %val
```

For binary numbers:

```
STR2VAL, %val, "#B101000101"
LOG, "Value=" %val
```

In all the examples above, `165` is stored to `'%val`.

THEN

Function

Performs the process if the latest condition determination result is true.

Syntax

```
THEN, <command>
```

Parameter

<command>	Command	Command to perform when determination result is true.

Description

Processes the command in the next column of THEN if the latest condition decision result in the stack is true.
It is not necessary to write THEN command in next line of IF command.
Performing THEN command depends on the result in the stack, not where IF command is written.
The commands excluding IF, IF_EX, THEN, ELSE, and ENDIF are available in the next column of THEN command.
When the stack of the condition decision result is empty, performing THEN command causes an execution error.

Example of Use

```
IF, %Value, ==, 100
  THEN, LOG, "%Value is 100"
  THEN, GOSUB, :Next
ENDIF
```

TIMEOUT

Function

Sets the timeout of the communication with the Network Master.

Syntax

```
TIMEOUT, <time>
```

Parameter

<time>	Numeric Column	Time of the timeout (ms unit) 0 to 999999999

Description

Changes the timeout of the waiting time to receive response from the Network Master.
After performing this command, script waits the response from Network Master up to the specified time.

The default timeout is 30 seconds. Specifying negative value to timeout causes a syntax error.

Example of Use

```
TIMEOUT, 3000
TIMEOUT, %MaxTime
```

WAIT

Function

Waits for specified time.

Syntax

```
WAIT, <time>
```

Parameter

<time>	Numeric Column	Waiting time (ms unit) 0 to 60000

Description

Waits for specified time before performing next process.
Specifying negative value to waiting time causes a syntax error.
If waiting more than 60 seconds is required, use WAIT command in multiple times.

Example of Use

```
WAIT,  200
```

Script which waits one hour

```
COPY,  %i,  0
:Wait_loop
IF,  %i, <, 360
  THEN, WAIT, 10000
  THEN, CALC, %i, %i, +, 1
  THEN, GOTO, :Wait_loop
  ELSE, RETURN
ENDIF
```

B.4 Commands Performing the Judgement

This section explains the commands which perform the Pass or Fail judgement of the test result.

DLG_OK

Function

Displays the received response on the message box which has OK button.

Syntax

```
DLG_OK, [<scpi_command>, ][<scpi_query>, ][<response>]
```

Parameters

<scpi_command>	String Column	String of SCPI command
<scpi_query>	String Column	String of SCPI query
<response>	Variable Column	Variable which stores the response from Network Master

Description

Displays the received response from Network Master on the message box which has only OK button.
If <scpi_command> and <scpi_query> are omitted, the response from Network Master is not displayed on the message box.
This command does not perform the judgement against the response. The test result is always judged as Pass. This command is useful to confirm the received response.

Example of Use

```
DLG_OK, "*IDN?", %response
```

DLG_YESNO

Function

Displays the received response on the message box which has YES and NO buttons.

Syntax

```
DLG_YESNO, [<scpi_command>,][ <scpi_query>,]<response>
```

Parameters

<scpi_command>	String Column	String of SCPI command
<scpi_query>	String Column	String of SCPI query
<response>	Variable Column	Variable which stores the response from Network Master

Description

Displays the received response from Network Master on the message box which has only **YES** and **NO** buttons.
If <scpi_command> and <scpi_query> are omitted, the response from Network Master is not displayed on the message box.
If touching **Yes** on the dialog box, the test result is always judged as Pass.
If touching **No** on the dialog box, the test result is always judged as Fail.

Example of Use

```
DLG_YESNO, "ETH:PORT1:IFET? THR, BPE", %response
```

EQUAL

Function

Tests whether the response matches the expected value.

Syntax

```
EQUAL, [<scpi_command>, ][<scpi_query>, ]<expected>
```

Parameters

<scpi_command>	String Column	String of SCPI command (can be omitted)
<scpi_query>	String Column	String of SCPI query (can be omitted)
<expected>	String Column	String used for the judgement

Description

Compares the response received from Network Master and <expected>, judges the test result as Fail if they are not matched. If all parameters are omitted, the test result is judged as Pass.
"K" command is available as substitute of "EQUAL".

Example of Use

```
EQUAL, "*ESE "%Value, "*ESE?", "100"
K, "*ESE "%Value, "*ESE?", "100"
```

IM_COPY

Function

Loads characters from the specified file and assigns them to variables.

Syntax

```
IM_COPY, <scope>, <result>, <type>, <path>, <delimiter>,
<line>, <row>
```

Parameters

<scope>	String Column	Scope of the variable CURRENT Local variable which is available only in the current script GLOBAL Global variable which is accessible from all of scripts
<result>	Variable Column	Variable which stores the entered value
<type>	String Column	File type CSV Comma separated file (Comma Separated Values)
<path>	String Column	Path of a file to be loaded. Starts the path with the following string: /Internal or Internal /Usb or Usb
<delimiter>	String Column	Specifies the delimiter of CSV file. Only commas (",") are supported.
<line>	Numeric/String Column	Line number in the file where a parameter(s) to be loaded exists. Specify a value of 1 or above.
<row>	Numeric/String Column	Number from the top where the parameter(s) to be loaded exists in the specified line. Specifies a value of 0 or above. If 0 is specified, all parameters in the line will be loaded.

Description

If CSV is specified for <type>, the file specified for <path> is regarded as a matrix separated by <delimiter> characters. The character strings at the positions specified by <line> and <row> in that file are stored in <result>. The character format of the reference file is UTF-8, and the line feed character is CR+LF.

If characters cannot be loaded from the file and stored as variables, the application test result is Fail.

Example of Use

```
IM_COPY,"GLOBAL", %A, "CSV", "Internal/a.txt", ",", 1, 1
LOG, %A
IM_COPY,"GLOBAL", %A, "CSV", "Internal/a.txt", ",", 4, 2
LOG, %A
IM_COPY,"GLOBAL", %A, "CSV", "Internal/a.txt", ",", 4, 0
LOG, %A
```

If the contents of "/Internal/a.txt" are as follows,

```
Abc, 3940, telite, 192.168.10.10
Mtqoei, 94308, mk, 10.10.10.10
Aboat, 1902, tmao, 10.12.10.3
Zpoute, 109, xlite, 1.1.1.2
```

the log is displayed as follows.

```
ABC
109
Mtqoei, 94308, mk, 10.10.10.10
```

JUDGE_FAIL

Function

Judges the test result as Fail intentionally.

Syntax

```
JUDGE_FAIL
```

Parameter

None

Description

Judges the test result as Fail. This command is usually used in combination with IF command.

Example of Use

```
IF, %RESP, ==, 1
  THEN, LOG, "OK . RESP=1"
  ELSE, LOG, "Fail RESP!=1"
  ELSE, JUDGE_FAIL
ENDIF
```

NOT_EQUAL

Function

Tests whether the response does not match the undesirable value.

Syntax

```
NOT_EQUAL, [<scpi_command>, ][<scpi_query>, ]<unexpected>
```

Parameters

<scpi_command>	String Column	String of SCPI command (can be omitted)
<scpi_query>	String Column	String of SCPI query (can be omitted)
<expected>	String Column	String used for the judgement (can be omitted)

Description

Compares the response received from Network Master and <unexpected>, judges the test result as Fail if they are matched. If all parameters are omitted, the test result t of application is judged as Pass.

Example of Use

```
NOT_EQUAL, "*STB?", "4"
NOT_EQUAL, "INST:MOD:CAT", "MU100010A"
```

VAR_STORE

Function

Stores the response to a variable.

Syntax

```
VAR_STORE, <scpi_command>, <scpi_query>, <response>
```

Parameters

<scpi_command>	String Column	String of SCPI command
<scpi_query>	String Column	String of SCPI query
<response>	Variable Column	Variable which stores the response from Network Master

Description

Stores the received response to a variable.
If storing response to a variable failed, the test result of application is judged as Fail.
"V" command is available as substitute of "VAR_STORE".
To send only a query without sending an SCPI command, leave <scpi_command> blank as shown in **Example of Use**.

Example of Use

```
VAR_STORE, "*CLS", "*IDN?", %Info
V, "*CLS", "*IDN?", %Info

V,, "OTDR:TRACe:EELOss?", %Loss
LOG, "Total Loss(dB):" %Loss
IF, %Loss, <, 10
THEN, LOG, "Pass"
ELSE, LOG, "Fail"
ELSE, JUDGE_FAIL
ENDIF
```

Appendix C Error Messages

This appendix describes error messages which appear when the script is validated.

Table C-1 Error Messages List

Message	Action
File not found	Check the file path in the fifth column of the IM_COPY command.
Invalid command	Check the command of the line number displayed in the error message.
Invalid parameter (Constant or Variable)	Check the following items and correct if there is any mistake. • Non-numeric character is used in the numeric column. • Two-byte character is used in the numeric value or variable. • Comma is forgotten after a numeric value. • 0 before decimal point is omitted when the value is less than 1. • Double-quotation (") is not used before and after the string. • The value is the out of range for numeric constant and numeric variable.
Invalid parameter (Variable)	Check the following items and correct if there is any mistake. • There are the character(s) other than alphanumeric and under bar (_) in the variable. • The top character of variable is not %. • The constant is written in the variable column.
Invalid parameter (Constant)	Check the command of the line number displayed in the error message.
Invalid parameter (String)	Check the following items and correct if there is any mistake. • Double-quotation (") is not used before and after the string. • Escape sequence (\) is not used when using double quotation (") in the string.
Invalid parameter (Operator)	Check the following items and correct if there is any mistake. • The character which is not defined as an operator is used. • Two-byte character is used in the operator. • Comma is forgotten after an operator. • An operator is not written in the third parameter of CALC command.
Invalid parameter (Numeric)	Check the command of the line number displayed in the error message.

Table C-1 Error Messages List (Cont'd)

Message	Action
Invalid parameter (Label)	Check the following items and correct if there is any mistake. • There are the character(s) other than alphanumeric and under bar (_) in the label column. • The top character of label column is not colon (:).
Undefined command	Check the following items and correct if there is any mistake. • Small character is used in the command. • There are misspelling in the command. • Two-byte character is used in the command. • Comma is forgotten after a command. • The top character of comment line is not apostrophe ('). • The top character of label line is not comma (:).
Invalid line	Check the line displayed in the error message.
Invalid character in label	There are the character(s) other than alphanumeric and under bar (_) in the label.
Duplicate label	Two or more duplicate labels exist. Change the labels so that the same labels do not exist.
Label not found	The label written as a parameter of GOTO or GOSUB command does not exist. Correct the description in the label column. Or write the label line which is described in the label column.
The 2nd column must be GLOBAL or CURRENT	Write "CURRENT" in the second column of INPUT_LIST, INPUT_NUM, INPUT_STR, and IM_COPY command.
The 4th column must be MAC, IPV4, IPV6 or STRING	Check the fourth column of INPUT_STR command.
The 4th column must be NUM, HEX or BIN	Check the fourth column of INPUT_NUM command.
Message is too long (max length=160)	Change the string length of MESSAGE command to 160 or less.
Invalid conditional expression format	Check the condition equation of IF_EX command.
Out of range value	The parameter of WAIT or TIMEOUT command is out of range. Change the value of the parameter.
Mismatch value	Set parameters of INPUT_NUM command as they satisfy the following equation. <min> ≤ <default> ≤ <max>

Appendix D Loop Processing

This appendix describes how to describe loop processing in the command sequence.

D.1 Loop Processing in Command Sequence

To do loop processing within a command sequence, use GOTO command in the script. Figure D.1-1 shows an example script description. Note that this example is available for applications excluding VIP.

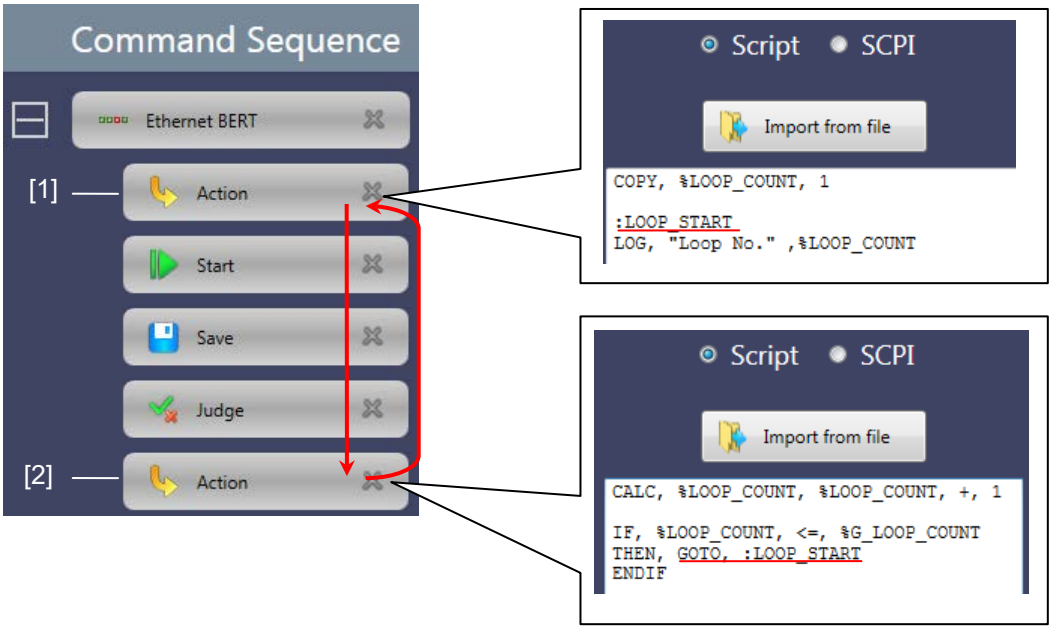


Figure D.1-1 Example Script of Loop Processing

In the example, loop processing will be performed between Action [1] and Action [2].
Write a label column in script [1].command
In script [2], write the GOTO command which jumps to the label column of script [1].

Loop processing of commands is available only for the sequence in the same application.
It is not able to jump to the command in other applications ([2] to [3]) as shown in the example in next page.

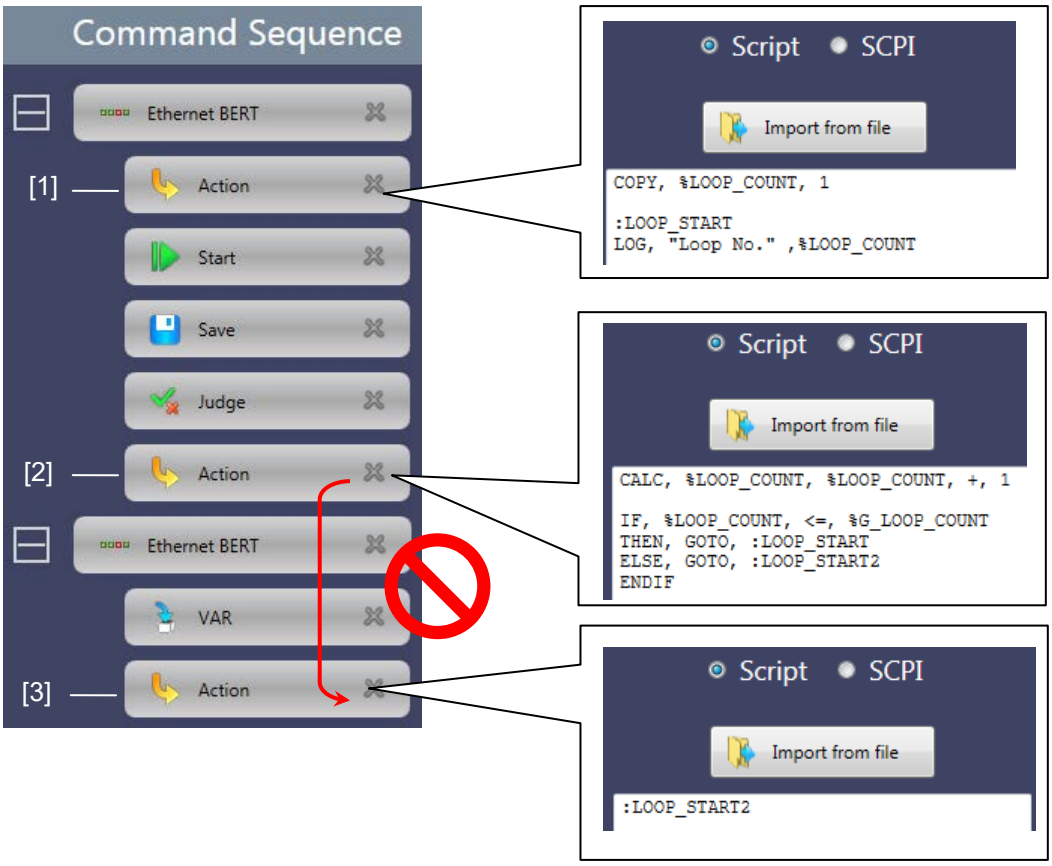


Figure D.1-2 Example Script Which Raises Error

Syntax is checked when saving the scenario and compile error shown in next page appears.

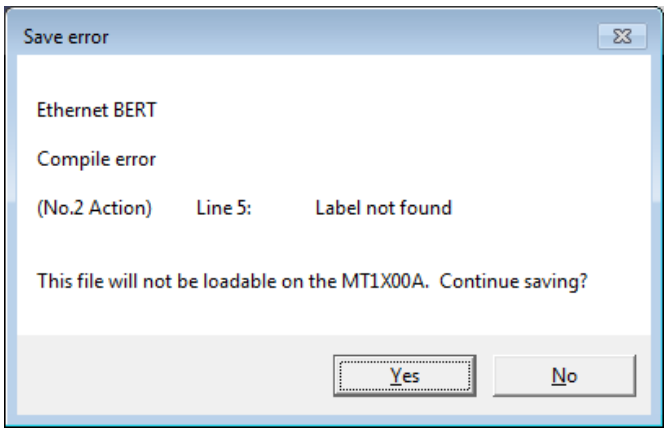


Figure D.1-3 Save Error Example

This error shows that the label column :LOOP_START2 in the script [2] was not found. Click **No** and modify the scenario.

D.2 Loop Processing in VIP Application

In VIP application, it is not able to jump to the label column in other commands. Jump to top of the application is just available.

Specify :TEST_START_APP reserved by the software as destination label to jump.

Also exit VIP application by executing EQUAL, “INST:TERM” before jump.

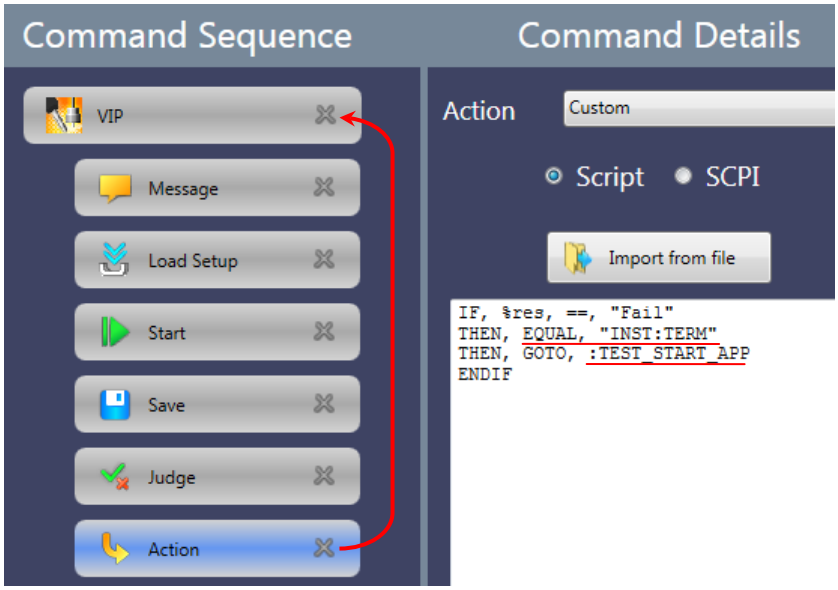


Figure D.2-1 Example Script of VIP Application

For other than VIP application, jump to top of the application is available by the same manner too.