www.tek.com/keithley

Automated Characterization Suite (ACS) Fundamentals

Reference Manual

ACS-914-01 Rev. C November 2023

ACS-914-01C

KEITHLEY
A Tektronix Company

ACS

Automated Characterization Suite

Fundamentals Reference Manual

Document number: ACS-914-01 Rev. C November 2023

KEITHLEY
A Tektronix Company

Safety precautions

The following safety precautions should be observed before using this product and any associated instrumentation. Although some instruments and accessories would normally be used with nonhazardous voltages, there are situations where hazardous conditions may be present.

This product is intended for use by personnel who recognize shock hazards and are familiar with the safety precautions required to avoid possible injury. Read and follow all installation, operation, and maintenance information carefully before using the product. Refer to the user documentation for complete product specifications.

If the product is used in a manner not specified, the protection provided by the product warranty may be impaired.

The types of product users are:

Responsible body is the individual or group responsible for the use and maintenance of equipment, for ensuring that the equipment is operated within its specifications and operating limits, and for ensuring that operators are adequately trained.

Operators use the product for its intended function. They must be trained in electrical safety procedures and proper use of the instrument. They must be protected from electric shock and contact with hazardous live circuits.

Maintenance personnel perform routine procedures on the product to keep it operating properly, for example, setting the line voltage or replacing consumable materials. Maintenance procedures are described in the user documentation. The procedures explicitly state if the operator may perform them. Otherwise, they should be performed only by service personnel.

Service personnel are trained to work on live circuits, perform safe installations, and repair products. Only properly trained service personnel may perform installation and service procedures.

Keithley products are designed for use with electrical signals that are measurement, control, and data I/O connections, with low transient overvoltages, and must not be directly connected to mains voltage or to voltage sources with high transient overvoltages. Measurement Category II (as referenced in IEC 60664) connections require protection for high transient overvoltages often associated with local AC mains connections. Certain Keithley measuring instruments may be connected to mains. These instruments will be marked as category II or higher.

Unless explicitly allowed in the specifications, operating manual, and instrument labels, do not connect any instrument to mains.

Exercise extreme caution when a shock hazard is present. Lethal voltage may be present on cable connector jacks or test fixtures. The American National Standards Institute (ANSI) states that a shock hazard exists when voltage levels greater than 30 V RMS, 42.4 V peak, or 60 VDC are present. A good safety practice is to expect that hazardous voltage is present in any unknown circuit before measuring.

Operators of this product must be protected from electric shock at all times. The responsible body must ensure that operators are prevented access and/or insulated from every connection point. In some cases, connections must be exposed to potential human contact. Product operators in these circumstances must be trained to protect themselves from the risk of electric shock. If the circuit is capable of operating at or above 1000 V, no conductive part of the circuit may be exposed.

Do not connect switching cards directly to unlimited power circuits. They are intended to be used with impedance-limited sources. NEVER connect switching cards directly to AC mains. When connecting sources to switching cards, install protective devices to limit fault current and voltage to the card.

Before operating an instrument, ensure that the line cord is connected to a properly-grounded power receptacle. Inspect the connecting cables, test leads, and jumpers for possible wear, cracks, or breaks before each use.

When installing equipment where access to the main power cord is restricted, such as rack mounting, a separate main input power disconnect device must be provided in close proximity to the equipment and within easy reach of the operator.

For maximum safety, do not touch the product, test cables, or any other instruments while power is applied to the circuit under test. ALWAYS remove power from the entire test system and discharge any capacitors before connecting or disconnecting cables or jumpers, installing or removing switching cards, or making internal changes, such as installing or removing jumpers.

Do not touch any object that could provide a current path to the common side of the circuit under test or power line (earth) ground. Always make measurements with dry hands while standing on a dry, insulated surface capable of withstanding the voltage being measured.

For safety, instruments and accessories must be used in accordance with the operating instructions. If the instruments or accessories are used in a manner not specified in the operating instructions, the protection provided by the equipment may be impaired.

Do not exceed the maximum signal levels of the instruments and accessories. Maximum signal levels are defined in the specifications and operating information and shown on the instrument panels, test fixture panels, and switching cards.

When fuses are used in a product, replace with the same type and rating for continued protection against fire hazard.

Chassis connections must only be used as shield connections for measuring circuits, NOT as protective earth (safety ground) connections.

If you are using a test fixture, keep the lid closed while power is applied to the device under test. Safe operation requires the use of a lid interlock.

If a screw is present, connect it to protective earth (safety ground) using the wire recommended in the user documentation.

The symbol on an instrument means caution, risk of hazard. The user must refer to the operating instructions located in the user documentation in all cases where the symbol is marked on the instrument.

The symbol on an instrument means warning, risk of electric shock. Use standard safety precautions to avoid personal contact with these voltages.

The symbol on an instrument shows that the surface may be hot. Avoid personal contact to prevent burns.

The symbol indicates a connection terminal to the equipment frame.

If this symbol is on a product, it indicates that mercury is present in the display lamp. Please note that the lamp must be properly disposed of according to federal, state, and local laws.

The **WARNING** heading in the user documentation explains hazards that might result in personal injury or death. Always read the associated information very carefully before performing the indicated procedure.

The **CAUTION** heading in the user documentation explains hazards that could damage the instrument. Such damage may invalidate the warranty.

The **CAUTION** heading with the symbol in the user documentation explains hazards that could result in moderate or minor injury or damage the instrument. Always read the associated information very carefully before performing the indicated procedure. Damage to the instrument may invalidate the warranty.

Instrumentation and accessories shall not be connected to humans.

Before performing any maintenance, disconnect the line cord and all test cables.

To maintain protection from electric shock and fire, replacement components in mains circuits — including the power transformer, test leads, and input jacks — must be purchased from Keithley. Standard fuses with applicable national safety approvals may be used if the rating and type are the same. The detachable mains power cord provided with the instrument may only be replaced with a similarly rated power cord. Other components that are not safety-related may be purchased from other suppliers as long as they are equivalent to the original component (note that selected parts should be purchased only through Keithley to maintain accuracy and functionality of the product). If you are unsure about the applicability of a replacement component, call a Keithley office for information.

Unless otherwise noted in product-specific literature, Keithley instruments are designed to operate indoors only, in the following environment: Altitude at or below 2,000 m (6,562 ft); temperature 0 °C to 50 °C (32 °F to 122 °F); and pollution degree 1 or 2.

To clean an instrument, use a cloth dampened with deionized water or mild, water-based cleaner. Clean the exterior of the instrument only. Do not apply cleaner directly to the instrument or allow liquids to enter or spill on the instrument. Products that consist of a circuit board with no case or chassis (e.g., a data acquisition board for installation into a computer) should never require cleaning if handled according to instructions. If the board becomes contaminated and operation is affected, the board should be returned to the factory for proper cleaning/servicing.

Safety precaution revision as of June 2018.

Table of contents

Section 1

Introduction

In this section:

Introduction to this documentation

If you have any questions after reviewing this information, please contact your local Keithley Instruments office, sales partner, or distributor. You can also call the Tektronix corporate headquarters (toll-free inside the U.S. and Canada only) at 1-800-833-9200. For worldwide contact numbers, visit tek.com/contact-tek.

NOTE

When the Series 2600B System SourceMeter® instruments are referenced, it also includes the Series 2600A System SourceMeter instruments.

Reference manual content

This manual contains information about the ACS software. It includes information about how to install, configure, and operate the ACS software and full details of software features such as prober control, wafer setup, summary reports, and building, loading, and executing test projects.

For more information about instruments that can be used with the ACS software, refer to the documentation for each specific Keithley Instruments model:

- Model 4200-SCS Semiconductor Characterization Suite
- Model 4200A-SCS Parameter Analyzer
- Series 2600B System SourceMeter
- Series 2650A High Power System SourceMeter
- Series 2400 SourceMeter
- Series 3700A System Switch/Multimeter
- Model 707A or 707B Switching Matrix
- Model 708A or 708B Switching Matrix
- S500 Integrated Test System

NOTE

When the Model 4200A-SCS is indicated in this manual, it includes both Model 4200A-SCS Parameter Analyzer and the Model 4200-SCS Semiconductor Characterization Suite unless otherwise specified.

You can visit the Keithley Instruments website to search for updated information by model number.

ACS software overview

The Keithley Instruments Automated Characterization Suite (ACS) software supports semiconductor characterization at the cassette, wafer, and device level, using multiple test instruments and automatic parametric testing with semiautomatic and automatic probe stations.

ACS software controls system instruments and hardware connected through the GPIB, LAN, USB, and RS-232 serial port. This includes:

- Models 4200-SCS and 4200A-SCS Parameter Analyzers
- Series 2600B System SourceMeter® instruments
- Series 2400 SourceMeter® instruments
- Model 707A and 707B Switch Matrices
- Series 3700A System Switch/Multimeter instruments

ACS can perform multisite parallel I-V testing using a Series 2600B as the primary instrument. You can have as many as 16 groups of source-measure units (SMUs) on a single GPIB card, for a total of 16 parallel test sites. Each group can support up to 64 Series 2600B instruments, or 128 channels.

NOTE

Contact Keithley Instruments if you need more than 16 parallel sites.

ACS software is supported on the following probers:

- Tokyo Electron Limited (TEL): P8/P12/19S Prober and T78S/80S Prober

NOTE

To make sure that ACS software will properly control the TEL P8 prober, use the following method to configure the settings: On the TEL P8 prober, go to the **Main Menu** > **Diagnostics** > **Adjustments** (enter a password, if necessary) > **HARDWARE OPTIONS** > **GPIB Parameter Settings** > **Output SRQ 47H at the Off Wafer With X, Y, I, J, b**. Then select **Yes**.

- Tokyo Semitsu: Semitsu (TSK) UF200/UF3000/APM60/70/80/90 Prober
- MPI Corporation: SENTIO Prober
- Cascade Microtech: Tesla prober
- Cascade Microtech: 12000/S300/R19S/CM300 Prober

NOTE

If you want to make the ACS prober ignore or execute the Z/D command, open the `.ini` file and make the following changes:: Go to the `C:\ACS\KATS\ACS_setting.ini` file > `[common]` > `chuck_for_move=1 (1 = ignore Z/D; 0 = execute Z/D).`

- Electroglas: EG2X/EG4X Prober

NOTE

Only the EG4X prober supports post-inking in ACS.

- Wentworth Laboratories: Pegasus 300S
- Signatone: CM500 Prober (WL250)
- Suss Micro Tec: A200/PA300 Prober
- Yang: Sagi3 Prober with SRQ check
- Micromanipulator: 8860/P300A Prober
- Semiprobe SPFA Prober
- MJC AP-80 Prober
- Apolloware AP200/AP300 Prober
- Vector Semiconductor AX/VX Series Prober
- HiSOL HSP-300 Prober

ACS testing hierarchy

ACS provides a combined test-execution engine that supports Series 2600B SourceMeter® instruments, Series 2400 SourceMeter® instruments, and Model 4200A-SMU source-measure instruments through interactive test modules (ITMs). An ITM allows you to define a test interactively using the user interface.

A script test module (STM) supports Test Script Processor (TSP®) files. A TSP file customizes tests for a Series 2600B, Series 2400, or Model 3706A Switch/Multimeter instrument. A Python programming language test module (PTM) supports control of any GPIB, LAN, or USB instrument. A script editor is included with ACS for editing STMs and PTMs.

ACS supports semiautomatic and automatic testing, multipattern wafer maps, and Series 2600B multigroup tests and Series 2400 mixed group tests. Common transistor parameter testing (for example, Vds-Id, Vgs-Id, Vg-Ig) and life testing (TDDB, NBTI, HCI) are supported.

NOTE

Another option for test modules is the C Test Module (CTM). The CTM is created using Microsoft Visual C++. This method of test creation is most useful when interfacing with external DLL libraries or when ACS is running directly on a Keithley 4200A-SCS or 4200-SCS. There is no access to LPTLib when this method is used. As a result, this is the most difficult method of test library creation and should be reserved for use in creating drivers for unsupported instruments or incorporating third-party libraries into the ACS environment. Refer to the *ACS Programmer's Manual* (ACS-907-01) for more detail.

ACS organizes tests in the hierarchy below (from the cassette-level down), consistent with semiconductor-wafer organization, allowing for a one-time setup when managing cassette-level tests:

- **Cassette:** Automate multiple wafers tests
- **Wafer:** Define test patterns
- **Pattern:** Multiple sites (nonexclusive) for each pattern
- **Device:** Perform one or more tests (single or multigroup)
- **Subsite:** Perform one or more tests on devices
- **Test:** Four types = ITM, STM, PTM, and CTM

Section 2

Software installation

In this section:

ACS software introduction

ACS can be installed on any computer that meets the minimum configurations provided in the ACS datasheet, available at tek.com/keithley.

For help with installing ACS software, refer to the *ACS Quick Start Guide*.

ACS software supports these system configuration arrangements (see Hardware configuration (on page 10-1) for a complete list).

- ACS software on a system computer or laptop for the Series 2600B System SourceMeter® units.
- ACS software on Model 4200A-SCS or 4200-SCS (includes all instruments and equipment that are supported by KITE).
- ACS software on Model 4200A-SCS and 4200-SCS for the Series 2600B System SourceMeter units.

For detailed configuration information, refer to Test and control connections (on page 11-1).

NOTE

ACS software starts in one of three modes, as described in ACS modes. If you want to start in online mode, a license file is needed. For information about installing a license and using a floating license, refer to the *ACS Quick Start Guide*.

NOTE

It is recommended that you back up your system before installing new software.

NOTE

If you install ACS directly onto your computer without KITE, the elements related to KITE will not be available for you to use with ACS.

ACS post-installation

NOTE

After you have installed ACS software and before you set up any projects or run any tests, it is recommended that you review the ACS Release Notes. The Release Notes contain valuable information about known issues and provide guidance on how to handle issues that you may encounter. The Release Notes are available from the Windows Start menu and from the Help menu in ACS.

If ACS software is controlling a Series 2600B, Model 3706A, or other instruments with GPIB, you must run the Hardware Management Tool (on page 10-4) to configure the instruments. You may also add a prober station to the configuration using the Hardware Management Tool.

ACS license options

Refer to the ACS datasheet, available at tek.com/keithley, for information on the license options.

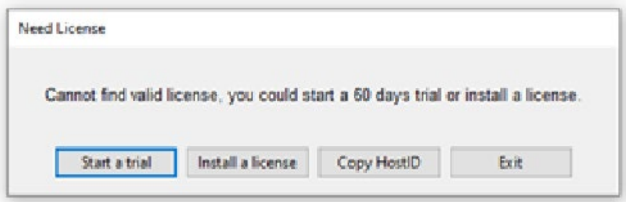
Trial license

If you do not have a license, when you first open the software, you will receive the following message. You can start a 60-day trial at this time, as shown in the following figure. Once the trial license ends, you can obtain a license.

NOTE

If you choose to start a 60-day trial, you will not have access to the reliability test modules (RTM).

Figure 1: Need License



A maintenance license is required after a 12-month support period to receive extended support. For additional information regarding license options, see tek.com/tektronix-maintenance-for-software.

For help with obtaining and installing a software license, refer to the *Quick Start Guide*.

S500 equipment and system

Refer to the *S500 Integrated Test System Administrative Guide* (PA-939) for specific test system information.

Section 3

Software overview

In this section:

Before starting ACS software

NOTE

Always start the subordinate instruments first before powering on the master instrument. If you power on the master instrument first, you may encounter errors on the subordinate instruments.

CAUTION

To avoid fatal errors to instruments, never start ACS software until all the instruments in the test configuration have completed all self-testing.

Before you start using ACS software:

- Make sure all the instruments in your test configuration are connected with a GPIB, LAN, USB, or a serial cable (for instance, an instrument to a system computer or a 4200A-SCS).
- Make sure any Series 2600B instruments have a TSP-Link® connection between them. For more information about setting up the hardware, refer to Test and control connections (on page 11-1).
- Assign GPIB addresses and node numbers (see Test and control connections (on page 11-1)).
- Make sure there is power on all the instruments.
- Allow self-testing to complete.

Start ACS software introduction

Before you begin using the software, make sure you have your instrument configuration completed. The Hardware Management Tool (on page 10-4) is how you manage the instrument configuration and the Keithley configuration file (.kcf) of the hardware.

ACS software starts in one of two modes: Demo or Online (see Demo mode (on page 3-2) and Online mode (on page 3-3) for more detail).

Demo mode

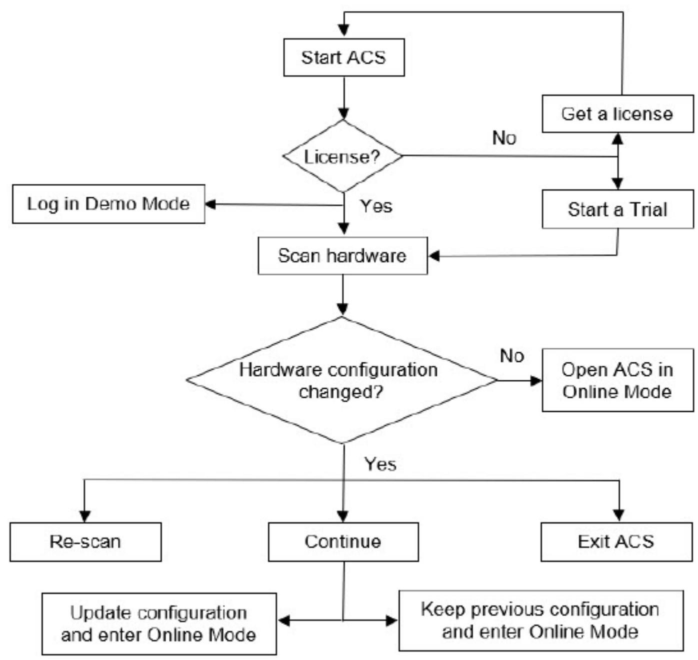
Demo mode is used to manage and simulate a virtual hardware system without actual instruments connected to the system computer. It is used for developing test projects and to see the results. While in Demo mode, ACS software cannot communicate with the test and measurement hardware. See the Configure Demo mode topic for more detail.

Online mode

In Online mode, ACS communicates with the test and measurement hardware and you can set up your testing process to control the prober station. When in online mode you can set up your testing process to control a prober station.

Note that the software may indicate a default project requires hardware that is not available. In this case, you will need to open the Hardware Management Tool to configure your test. See the following figure for a process flow.

Figure 2: ACS Software startup



Start ACS software

NOTE

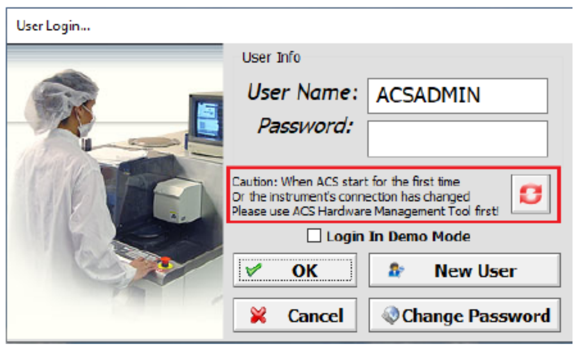
Before using the software, you must scan the hardware of your test configuration. If you do not scan the hardware, you see a warning indicating the hardware connection has changed. Make sure you have reviewed Hardware Management Tool (on page 10-4).

Before opening the software, make sure all the instruments are turned on and all self-tests are complete. You must purchase an ACS software license to start the software as a licensed user. If you do not have a license, you must select Login in Demo Mode. For more detail, refer to ACS license options (on page 2-2).

To start the ACS software:

1. Select the ACS software icon on the desktop of your computer. The User Login dialog opens, as shown in the following figure.

Figure 3: User login



2. Type a user name in the User Name field. The default user name is ACSADMIN. There is no default password.

NOTE

You can create more than one new user account. Once the first user account is created, you can select **New User**.

3. Select **OK**. The software opens.
4. To run ACS without a license, or without instruments connected, select Demo Mode.

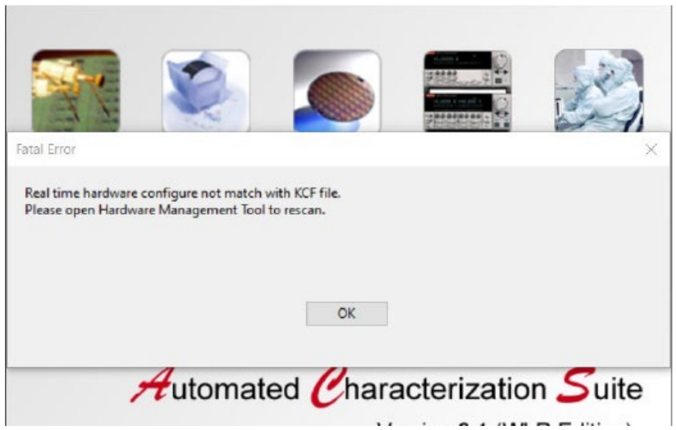
NOTE

Selecting Demo Mode before selecting **OK** enables ACS to immediately open in demo mode.

Depending on the license file you have (refer to ACS license options (on page 2-2) for more information), you may not be able to use all the ACS software features. However, with the 60-day trial version, you have full access to the ACS Standard Edition (ACS-STANDARDFL) software.

If you encounter a fatal error message when opening ACS, the hardware configuration may have changed (see the following figure for an example). You need to run the Hardware Management Tool (on page 10-4) to establish the hardware configuration.

Figure 4: Hardware configuration fatal error

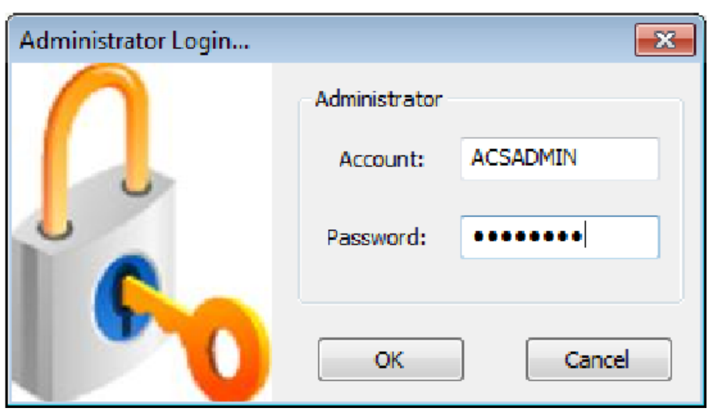


Create new user accounts

To create new user accounts:

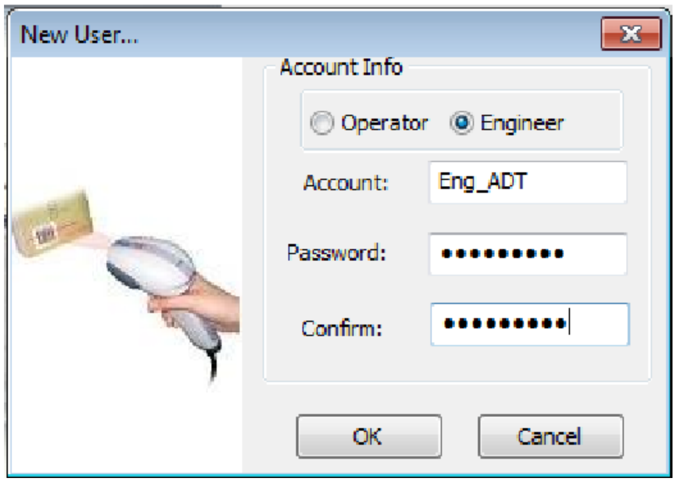
1. Select **New User** in the User Login dialog. The Administrator Login dialog is displayed.

Figure 5: ACS Administrator login



2. Enter a password for the administrator (default account usermane is `ACSADMIN` with no password). Once you enter your password, the password is created.
3. Select **OK**.

Figure 6: ACS new user account



4. Select the type of account: **Operator** or **Engineer**.
   - **Operator:** Allows the user to load and run test projects.
   - **Engineer:** Allows the user to access to all ACS software functions.
5. Enter an account name.
6. Enter a password.
7. Enter the password again in the Confirm field.
8. Select **OK**.

Initialize multiple user accounts

CAUTION

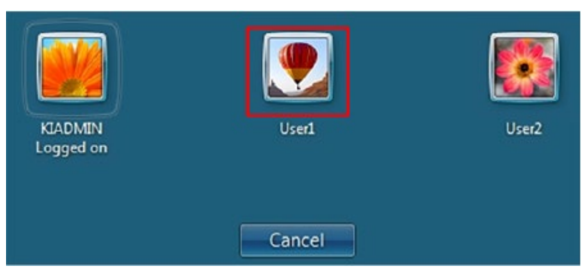
If you try to run ACS on a computer that has more than one Windows® user account and you have not initialized all the accounts to run ACS, you receive an error and you cannot use ACS software with all the user accounts. Refer to the following information to establish multiple user accounts in a Windows® environment.

NOTE

If you have other user accounts in your Windows® environment who need to access and use ACS software, you need to initialize the account for each user.

This information is for those who have more than one user account when logging in to a Windows® account. An example is shown in the following figure. If you have more than one user account on your computer, make sure that you have logged off the previous user account and have logged on using the correct user account.

Figure 7: Windows User Accounts

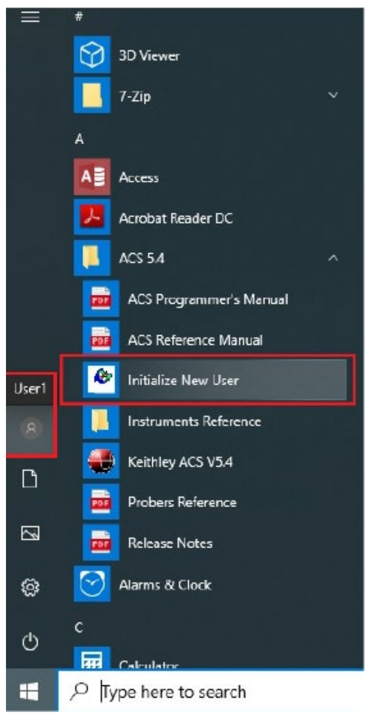


If one of the user accounts (Admin1, User1, or User2) has already installed ACS software, none of the other users need to install the software. However, you do need to initialize the user account to access and use ACS software. Additionally, if the Admin1 account is the user who installed ACS software on the computer, then Admin1 does not need to follow this procedure to access and use ACS.

To initialize a user account:

1. After selecting a user account (in this example, User1) and your desktop is loaded, select Windows® **Start** and find ACS in All Programs.
2. Select **Initialize New User**. A prompt to restart your computer is displayed.

Figure 8: Windows Start ACS Initialize New User



3. Restart your computer.
4. Log in to your account and run ACS software.

User accounts security

ACS software provides the following types of user accounts (or operating modes) and each has different rights and security features.

- **Engineer administrator:** There is only one engineer administrator. The engineer administrator account has unrestricted rights to create and edit test projects, including test parameters within ACS. It can also edit or delete operator accounts. The default administrator account usermane is ACSADMIN.
- **Engineer:** The engineer account has unrestricted rights to create and edit test projects, including test parameters within ACS. This user account cannot edit or delete operator accounts.
- **Operator:** The operator account can open and run available test projects that were created by either the engineer or administrator.

Account management

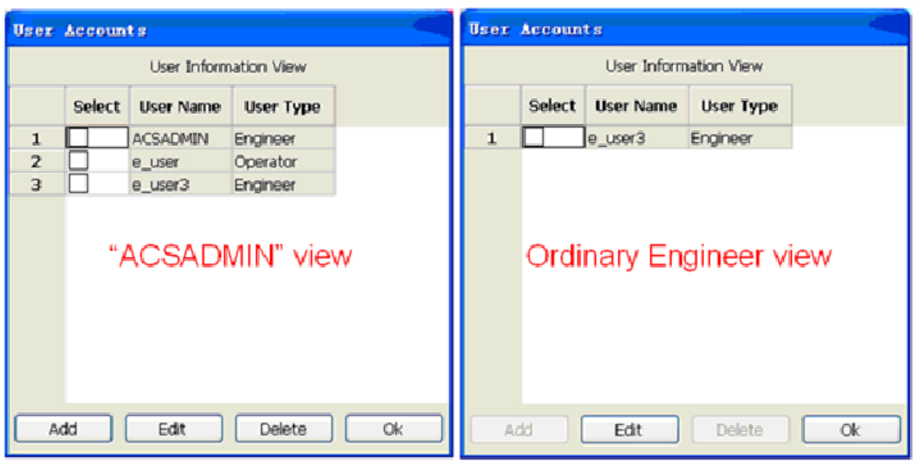
Only engineer accounts can access the User Accounts in the Tools menu and can only see their own accounts. Only the engineer administrator account ACSADMIN has access to the complete list of user accounts. Operators do not have access to other accounts.

To manage user accounts:

1. On the standard toolbar, select **Tools**.
2. On the menu, select **User Accounts**.

The dialog opens for the default engineer administrator account ACSADMIN or the default administrator engineer account, as shown in the following figure. The usermane indicates if the user is set as an Engineer or Operator. Notice that the ACSADMIN account shows all the user accounts.

Figure 9: User Accounts information



For more information about administrator accounts, see Add new user (on page 3-10).

Add new user

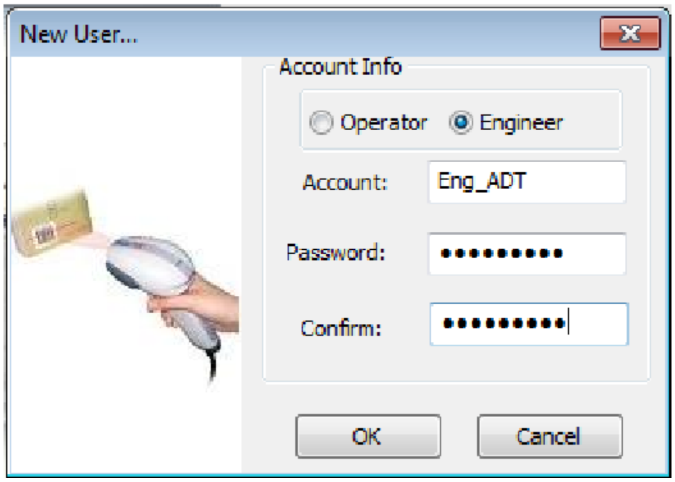
NOTE

Only the default engineer administrator account (ACSADMIN) has access to add a new operator or engineer account.

To add a new user:

1. Log into ACS as the ACSADMIN.
2. Select **Add** in the User Accounts dialog to create a new account. You see the **New User** dialog.

Figure 10: New User account



3. Select the **Operator** or **Engineer**.
4. Type an account name and password in the **Account** name and **Password** fields.
5. Type the password again in the **Confirm** field.
6. Select **OK**.

Change password

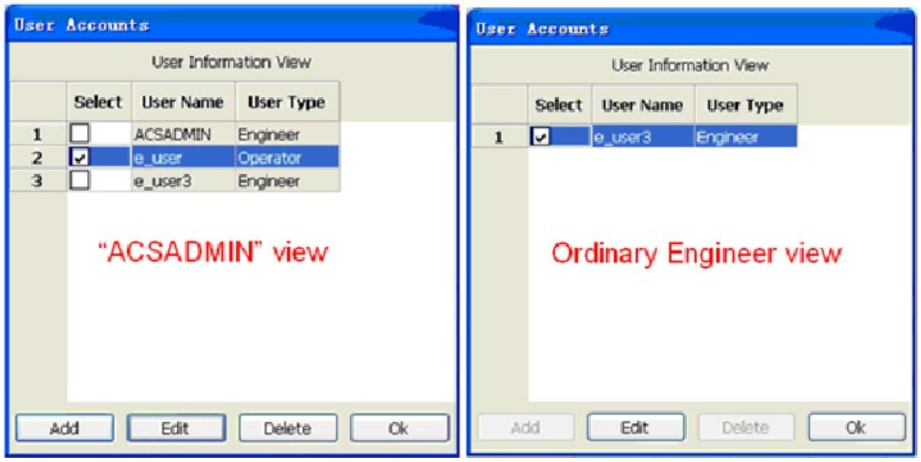
NOTE

Only the default engineer administrator account (ACSADMIN) can change a password for all the user accounts. An engineer account can change its own password, but not the password for other accounts. An operator cannot change the password.

To change the password:

1. Log into ACS as the ACSADMIN account or as an engineer account.
2. Select the account to change (see the following figure).

Figure 11: Select a user account



3. Select **Edit**. The **Change password** dialog opens.
4. Enter the present password of the account.
5. Enter a new password and confirm.

Figure 12: Change user account password

6. Select **OK**.

Delete user account

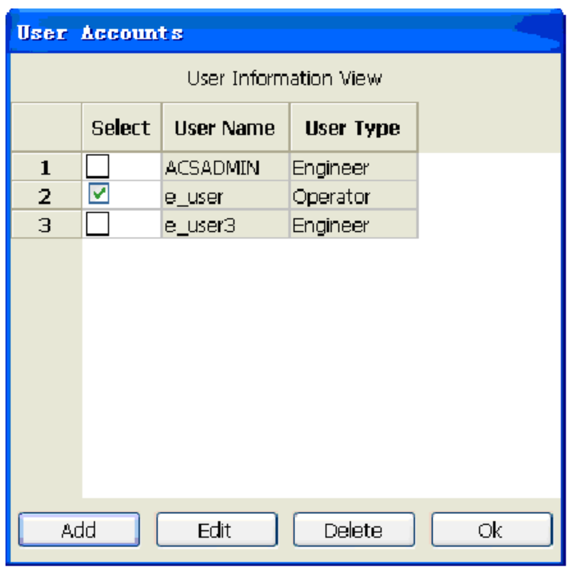
NOTE

Only the engineer default administrator account ACSADMIN has the necessary rights to delete user accounts. The engineer and operator account do not have access to this function.

To delete a user account:

1. On the User Accounts dialog shown in the following figure, select the account you want to delete.
2. Select **Delete**. A prompt is displayed.
3. Select **Yes** to delete the account.

Figure 13: Select user account to delete



ACS software GUI

Once ACS software has started, the ACS software graphical user interface (GUI) opens in either engineer mode or operator mode, as shown in the following figures.

Figure 14: ACS software in engineer mode

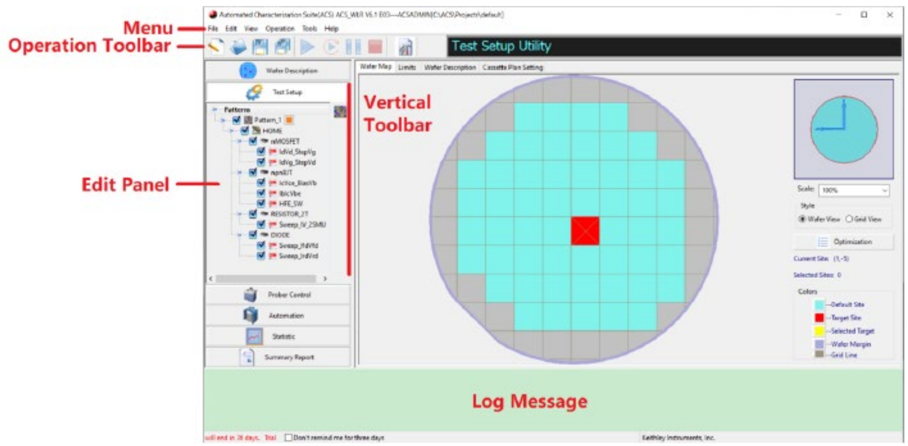
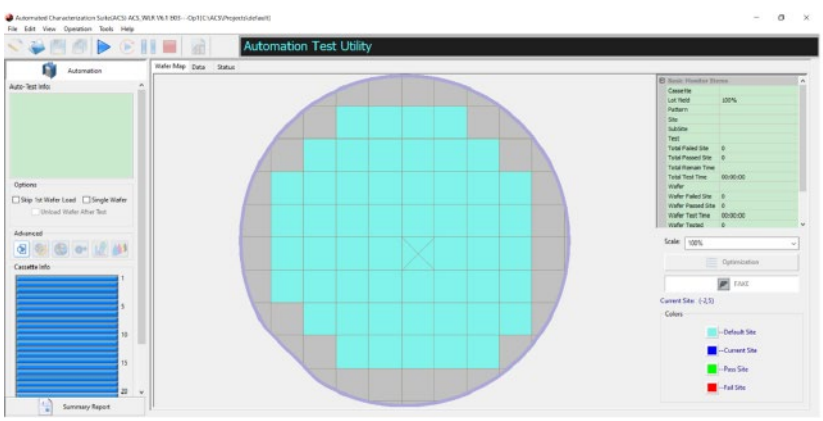


Figure 15: ACS software in operator mode



Operation toolbar

The Operation toolbar, shown in the following figure, is at the top of the ACS software Main dialog and contains a set of project file management functions that allow you to perform tasks.

Figure 16: Operation toolbar

Project file management functions

New, Open, Save, and Save All, shown in the following figure, allow you to create and manage project files. The options are:

- **New:** Opens a new project
- **Open:** Opens an existing project.
- **Save:** Saves an operation to the Test Tree.
- **Save All:** Saves all project settings and all test results data.

Figure 17: File management functions

NOTE

You must use the Save All function to be sure that all test results are saved and available in the data file.

Test control functions

Run, Repeat, Pause, and Stop, shown in the following figure, allow you to actively work with tests. The options are:

- **Run:** Initiates the test procedure.
- **Repeat:** Continues test execution until Stop is selected.
- **Pause:** Temporarily stops the test flow. The test pauses after the present operation finishes and the next test does not execute.
- **Stop:** Immediately ends testing.

Figure 18: Run, Repeat, Pause, and Stop

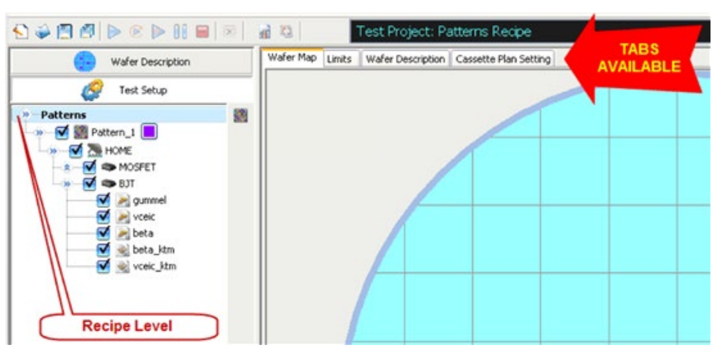
Vertical toolbar

On the vertical edit toolbar, select in the Test Setup pane, and the tabs available in the right pane change based on the level of your selection (recipe level, pattern level, subsite level, device level, and test level). See the following figure for an example of the tabs available with the recipe level selected.

NOTE

To view the vertical toolbar, fully expand the selections in the Test Setup dialog.

Figure 19: Recipe Level with tabs available



Test setup functions

Not all the test setup and project management function components are available for all node levels (such as Pattern (Site), Subsite, Device, and Test) in the test tree. They are added or removed as you select the different items.

The following table lists the project management functions.

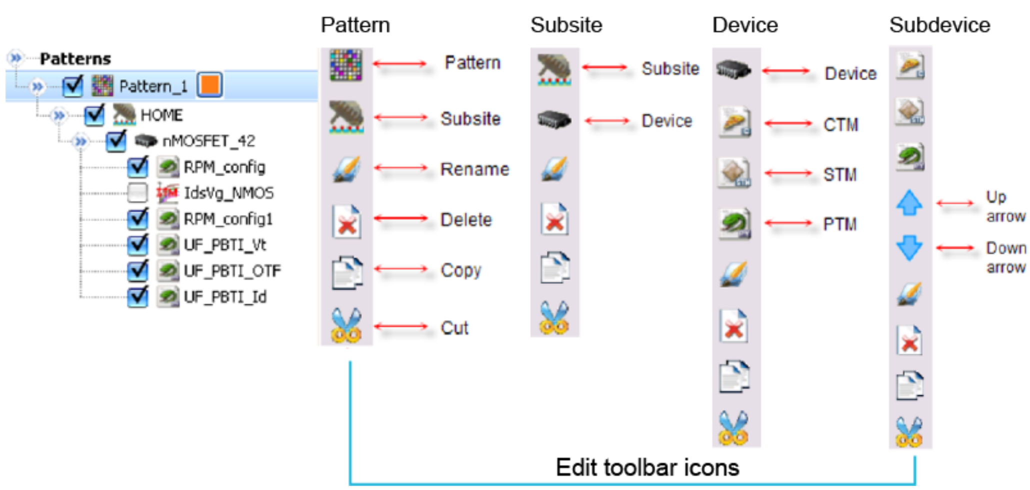
Rename icon	Rename your pattern, subsite, device, or test modules.
Delete icon	Delete a selected pattern, subsite, device, or test module.
Copy icon	Copy a pattern, subsite, device, or test module.
Cut icon	Cut a pattern, subsite, device, or test module.
Paste icon	Paste a pattern, subsite, device, or test module.
Up and Down icon	Move a pattern, subsite, device, or test module up or down, which also changes the test and device node sequence. These two options may be displayed or hidden depending on the present node location.

For examples of the Project management functions, see the figures in PATTERN, SUBSITE, DEVICE, ITM, STM, CTM, and PTM (on page 3-17).

PATTERN, SUBSITE, DEVICE, ITM, STM, CTM, and PTM

These functions are enabled or disabled depending on the active test functions. The following figure shows the Edit menus for the functions. A description of the functions follows the figure.

Figure 20: Icons on the Edit toolbars



Pattern: Pattern function allows wafer test set up site patterns.

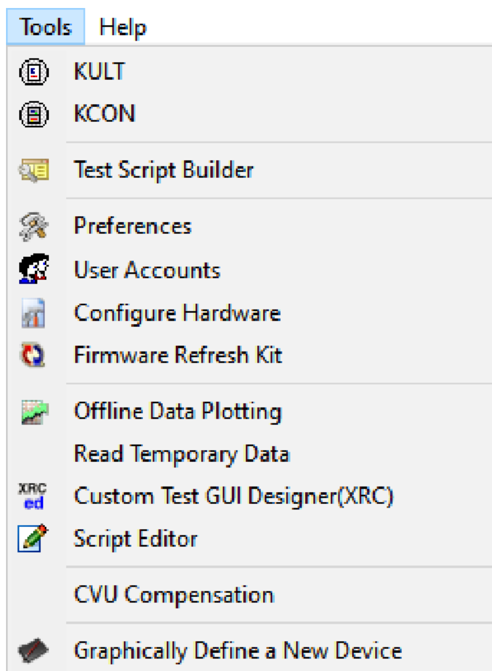
Subsite: Subsite function allows wafer subsites definition.

Device: Device function allows a choice of devices to be tested.

ITM, CTM, STM, and PTM (in this order): Device (ITM), CTM, STM, and PTM functions allow the classes of tests or test modules definition.

Tools menu

Figure 21: Tools menu



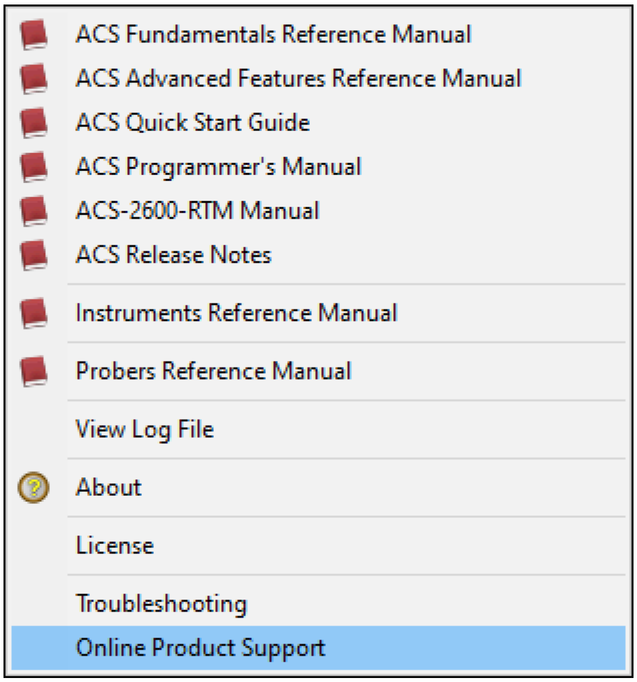
You can use the Tools menu to access functions that are related to the ACS software. The options are:

- **KULT:** Integrates custom algorithms (user modules) into KITE.
- **KCON:** Defines the configuration of external GPIB instruments, switch matrices, and analytical probers connected to the 4200A-SCS. KCON also provides basic diagnostic and troubleshooting functions.
- **Test Script Builder:** Orchestrates a sequence of events and programs the language for scripting.
- **Preferences:** Opens the advanced Preferences dialog. Refer to Log message window (on page 3-28) for more information.
- **User Accounts:** Views the dialog that contains all the configured user information.
- **Configure Hardware:** Accesses the Hardware Management Tool confirm mode.
- **Firmware Refresh Kit:** Refreshes the 2600B SourceMeter instrument firmware. This utility verifies the firmware on the Series 2600B instruments connected to ACS software and allows you to update the firmware with the latest files from tek.com/keithley.
- **Offline Data Plotting:** Accesses the data plot tool. Refer to Data plot (on page 5-233) for more information.
- **Read Temporary Data:** Browse to the path where the present temporary data is stored to read the data.
- **Custom Test GUI Designer (.xrc):** Accesses the graphical user interface (GUI) builder. Refer to Create a TSP GUI file (on page 5-129) for more information.
- **Script Editor:** Accesses the script editor tool. Refer to Script editor tutorials (on page 5-213) for more information.
- **CVU Compensation:** Accesses a CVU connection compensation by reading an open, short, load, or cable connection.
- **Graphically Define a New Device:** Accesses the device editor tool.

Help menu

The Help menu contains the ACS documentation, access to the log file, About information, licensing options, troubleshooting, and a link to online product support, as shown in the following figure.

Figure 22: Help menu



Edit panel

The left-side of the ACS Main dialog contains the edit panel. The following functions are on the edit panel:

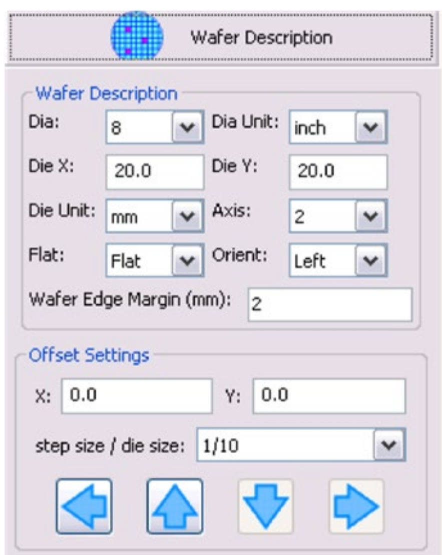
- Wafer description
- Test setup
- Prober control
- Automation
- Summary report
- Statistics

The test setup flow and details of each step are discussed in the following sections of this manual.

Wafer description

Select **Wafer Description** to display the wafer options (see the following figure).

Figure 23: Wafer Description panel



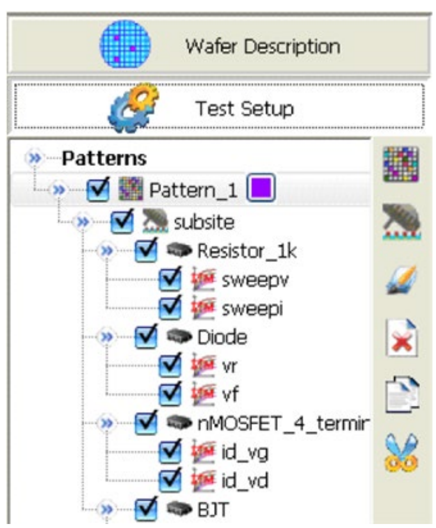
The Wafer Description panel is where you edit wafer information. The results immediately display in the work area to the right of the ACS Main dialog. Refer to the following table for a description of editable wafer characteristics.

Description of wafer characteristics	
Wafer description elements	**Description**
Dia	Select the diameter of the wafer (inches or millimeters) from a list. When Dia is changed, the wafer map is redrawn and all dies previously selected are cleared.
Dia Unit	Sets the unit of diameter (inches or millimeters). When Dia Unit is changed, the content of Dia is automatically converted.
Die X	Sets the X die size (millimeters or mils). When Die X is changed, the wafer map is automatically redrawn and all dies selected previously are cleared.
Die Y	Sets the Y die size (millimeters or mils). When Die Y is changed, the wafer map is redrawn automatically and all dies selected previously are cleared.
Die Unit	Sets the unit of die (millimeters or mils). When the Die Unit is changed, the X and Y die size values and X and Y of the subsite are converted automatically.
Axis	Sets the direction in which the X and Y coordinates increase.
Flat	Sets whether the wafer has a flat or notched edge.
Orient	Specifies the flat or notch position to be 0, 45, 90, 135, 180, 225, 270, or 315 degrees on the wafer.
Margin Size (mm)	Enters the width of the wafer margin (millimeter).
Offset Setting	There are four arrows for the offset setting: Left, up, down, right. When an offset setting is selected, the wafer moves to the corresponding direction.

Test setup

Select Test Setup on the Edit Panel to display the test tree in the edit panel (see the following figure).

Figure 24: Test Tree panel



The test tree panel displays your test flow characteristics. For example, it shows what kind of device will be tested and what test is to be performed.

General editing

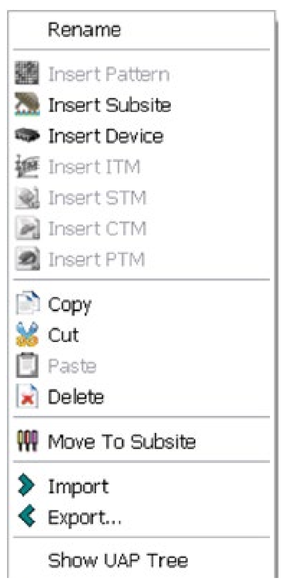
With the Test Setup panel active, you can make general edits to test patterns, subsites, devices, and test modules using the Copy, Paste, Cut, and Rename commands. Descriptions of these options are provided in the following table.

To make changes, select the node in the Test Setup panel, and then select the node to choose the editing command. For example, to make edits to a Subsite, select the Subsite node, then select the editing command.

NOTE

You must select a node to display the editing commands.

Figure 25: Subsite right-click menu

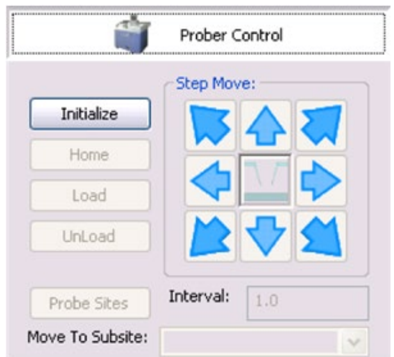


Test setup panel: Available editing commands	
Edit command	**Description**
Copy	Select and copy the pattern, subsite, device, or test module.
Cut	Select and cut the pattern, subsite, device, or test module.
Paste	After you copy or cut the pattern, subsite, device, or test module, select the place where you want to paste it. Select the location, then select Paste.
Rename	Select the pattern, subsite, device, or test module, then select Rename to change the name of the selected node. Available characters for naming are A to Z, a to z, 0 to 9, and "_". The first character must be a letter (A to Z or a to z).

Prober control

Select Prober Control on the edit panel. The Prober Control panel in the edit panel appears in the edit panel.

Figure 26: Prober Control panel



Prober Control is used for semiautomatic tests where you control the probers manually. The following table provides a description of the commands available to control probers.

Commands available	
Prober command	**Description**
Initialize	Establishes communication with the chosen prober.
Home	Moves to the target site.
Load	Loads a new wafer after unloading the presently loaded wafer.
Unload	Unloads a wafer. If no wafer is loaded, no action is taken.
Step Move	Moves to the next site in the corresponding direction. The function in the middle controls the chuck to be moved up and down.
Probe Sites	Moves to and touches all sites in every pattern to be tested and verifies that the wafer is correctly loaded but does not run the tests.
Interval	Time delay between every prober movement.
Move To Subsite	Moves to a selected subsite.

Automation panel

Select Automation on the edit panel to display the Automation panel in the edit panel.

The Automation panel allows you to run automation tests on an entire cassette. To run an automation test, select **Automation**, then select **Run**. The progress of the test appears and is refreshed in real time in the Auto-Test information table.

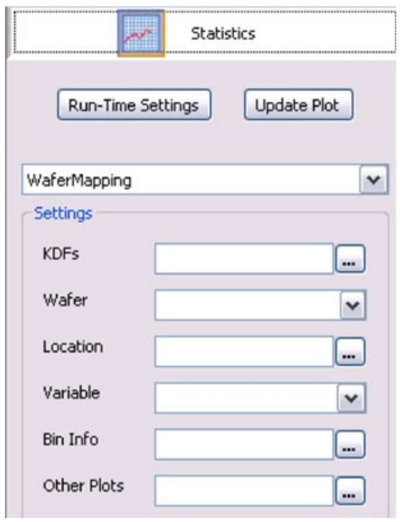
Figure 27: Automation panel

Statistics panel

ACS software provides statistical features so you can view data trends and plot distribution curves. You can do a simple analysis and modify the test conditions, if needed.

You can create an output file with a Keithley data file (`.kdf`), which also supports multiple KDF for analysis. You can also view the online data plot result while ACS testing is in progress.

Figure 28: Statistics panel

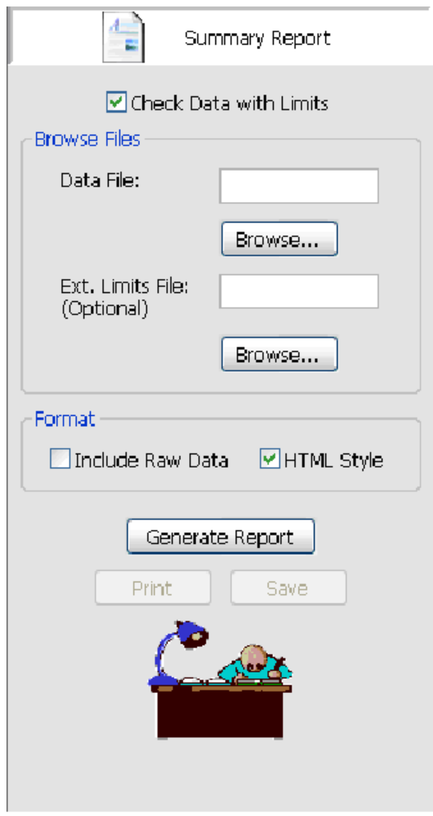


Summary report

To display a Summary Report in the edit panel, select **Summary Report** in the edit panel.

When the data file and limits file (optional) are imported, you can generate a report of the entire test by selecting the Generate Report function. The report appears in the ACS work area. If sorting by limits, your limit inputs are compared to the results. The results that exceed your limits are indicated in the report.

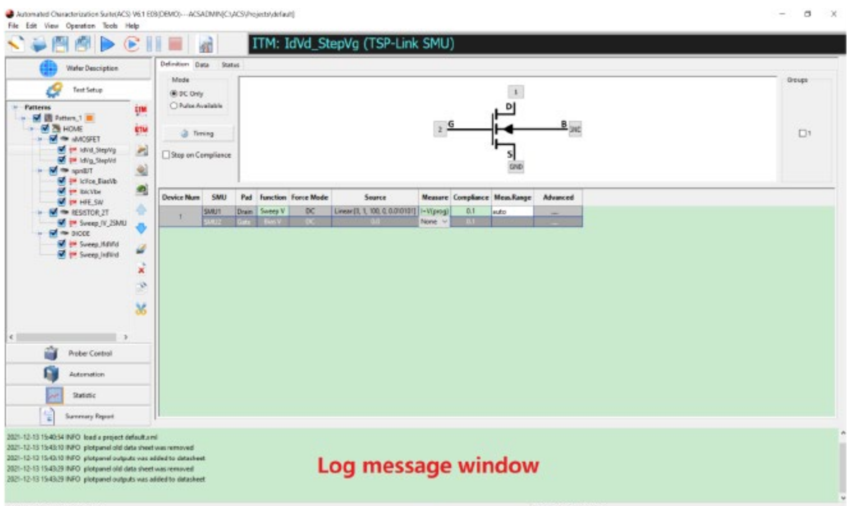
Figure 29: Summary Report panel



Log message window

The log message window is at the bottom of the main ACS user interface, as shown in the following figure. Log messages are displayed in real time. You can set the type of messages that are displayed. You can also hide the message window.

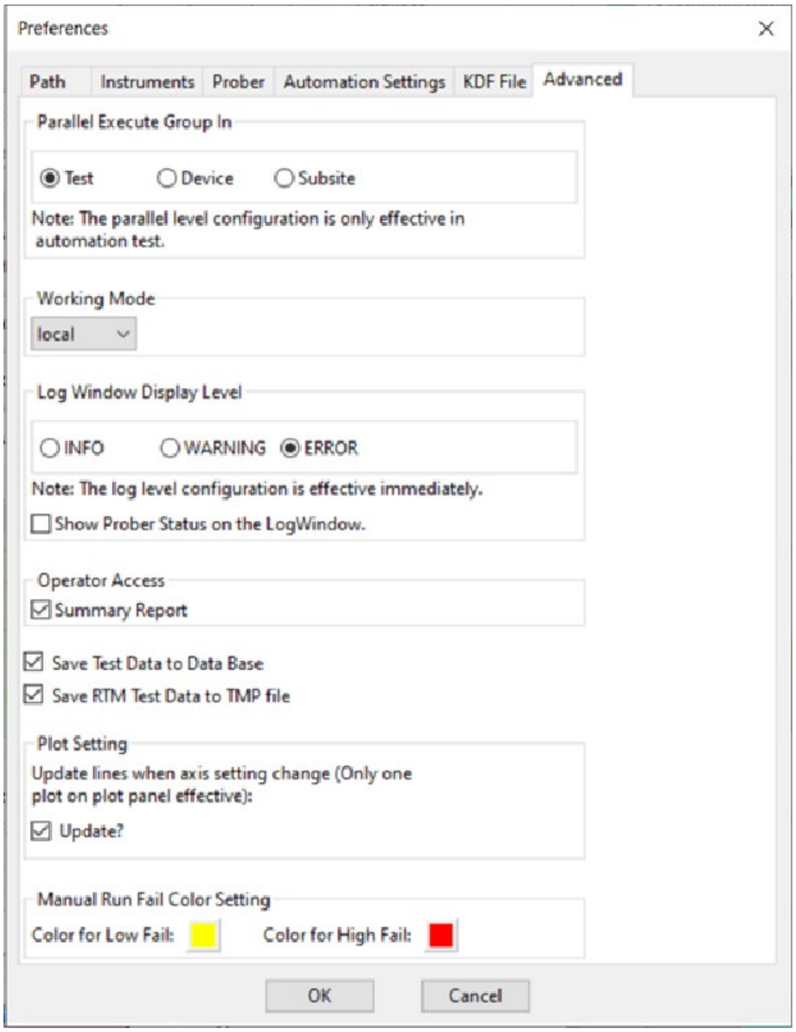
Figure 30: Log message window



To set the types of messages that are displayed in the log message window:

1. From the menu bar, select **Tools**.
2. Select **Preferences**. The dialog shown in the following figure is displayed.

Figure 31: Preferences Advanced tab

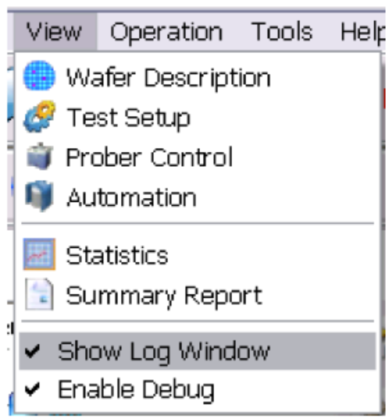


3. Select **Advanced**.
4. Select the type of log messages you to display: **INFO**, **WARNING**, or **ERROR**.

To hide or show the Log message window:

1. In the menu bar, select **View**.
2. Clear **Show Log Window** to hide the window. To redisplay the log, select **Show Log Window**.

Figure 32: View menu



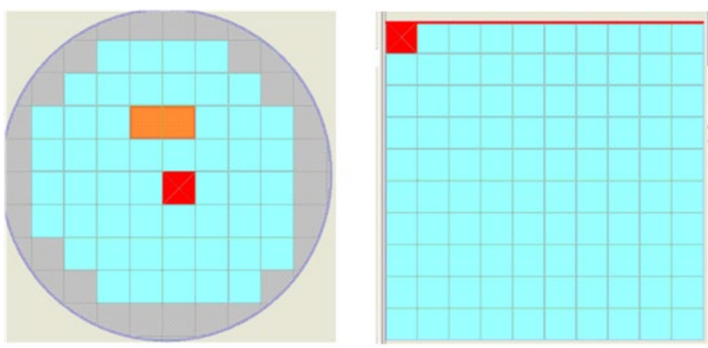
Wafer description

In this section:

Wafer introduction

ACS software supports circle and square shaped wafers. The circle-shaped wafer is the most common. The square-shaped wafer is less commonly applied. Examples of the circle and square-shaped wafers are shown in the following figure.

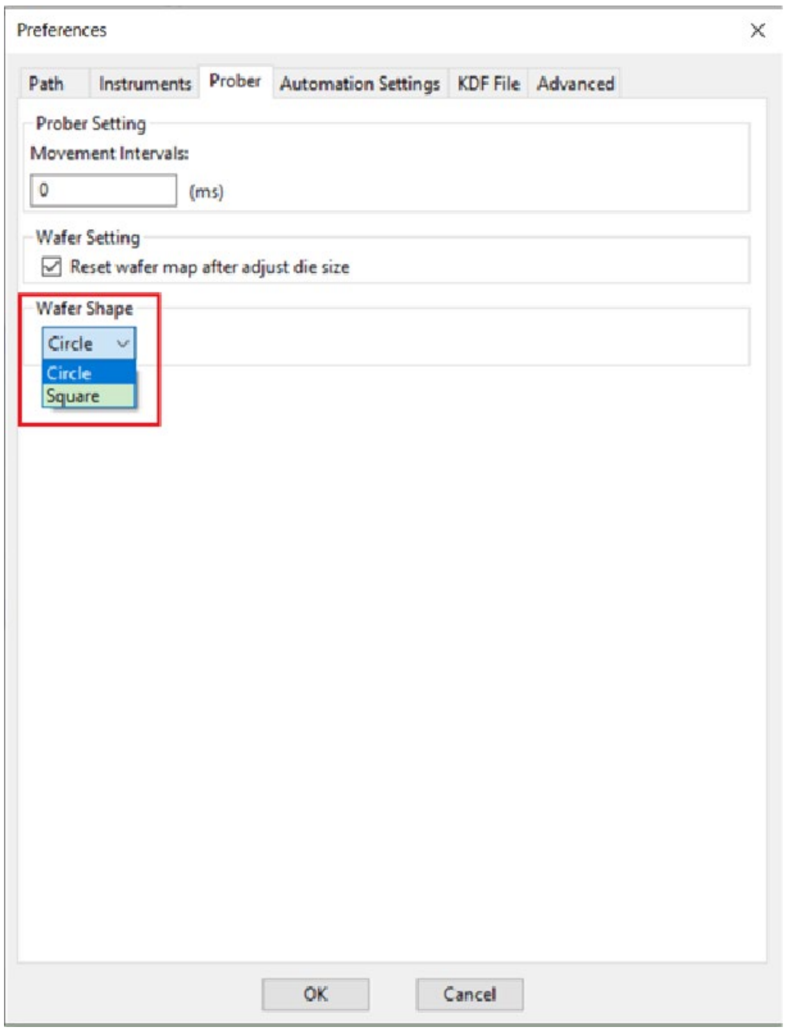
Figure 33: Square and circle wafer shapes



To select a wafer shape for a new project:

1. Select **Tools > Preferences**. The Preferences dialog opens.
2. Select the **Prober** tab.

Figure 34: Preferences Misc tab



3. Select the Wafer Shape.
4. Select the Instruments tab.
5. Select **Clear old TSP scripts when a project is opened and automation is run**. This eliminates previous scripts from the system.
6. Select **OK**. The shape takes effect when you create a new project.

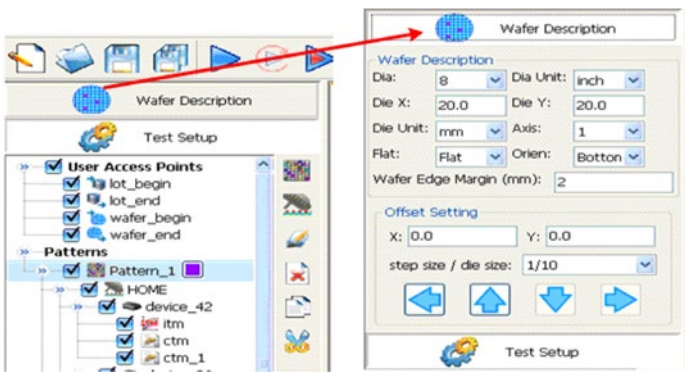
Circle-shaped wafer description panel

The Wafer Description panel allows you to define wafers. It is displayed when you select the Wafer Description tab. You can edit the wafer information in this panel and the results immediately display in the ACS main work area. You can review the following table for wafer description editing elements and their corresponding descriptions.

CAUTION

When you change the Orien setting of the wafer, the sites that you defined in the wafer map also change to be consistent with the flat or notch orientation. However, the Axis does not change. This may cause incorrect operation of the prober to your sites and subsites. It is highly recommended that you define the Orien setting before you define any sites or subsites.

Figure 35: Wafer Description panel activated



Wafer Description Elements	Description
Dia	Changes the intended diameter of the wafer to be tested from a list. When Dia is changed, the wafer map is redrawn and all dies previously selected are cleared.
Dia Unit	Changes the unit diameter (inches or millimeters). When the Dia Unit is changed, the content of Dia is automatically converted.
Die X	Sets the X die size (millimeters or mils). When Die X is changed, the wafer map is automatically redrawn and part of selected image previously is cleared[1].
Die Y	Sets the Y die size (millimeters or mils). When Die Y is changed, the wafer map is redrawn automatically and part of selected image is cleared[1].
Die Unit	Changes the unit die (millimeters or mils). When the Die Unit is changed, the X and Y die size and X and Y of the subsite values are converted automatically.
Axis	Sets the direction in which the X and Y coordinates increase.
Flat	Sets whether the wafer has a flat or notched edge.
Orien	Specifies the flat or notch position to be 0, 45, 90, 135, 180, 225, 270, or 315 degrees on the wafer.
Wafer Edge Margin (mm)	Sets the width of the wafer margin in millimeters.
Offset Settings	There are four arrows for the offset setting: Left, up, down, and right. When an offset function or button is selected, the wafer moves in the corresponding direction.

[1] There is a wafer setting in the instruments tab of the Preference Setting that you can select to clear all previous scripts that were generated for the Series 2600B instruments.

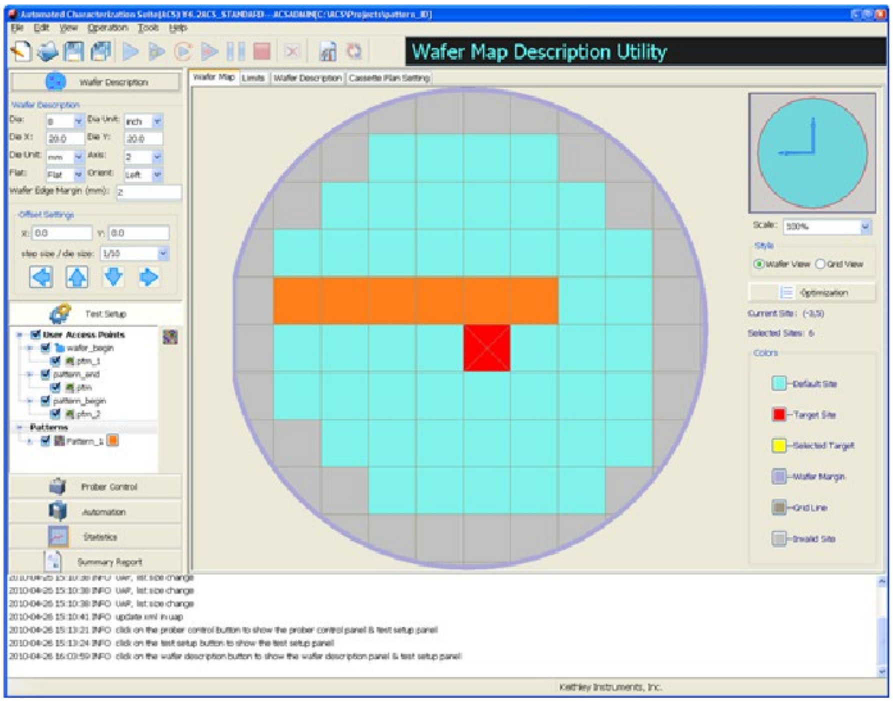
Wafer pattern

At the macroscopic level, one or more semiconductor dies are built-up at a given wafer location.

This location is comprised of end-product dies and usually one or more parametric test structures or subsites. ACS refers to such a repeating pattern of dies and test structures as a pattern.

Once you have inserted the wafer description information, the wafer map opens in the ACS GUI work area. You are now ready to define the pattern information.

Figure 36: Wafer Map Description

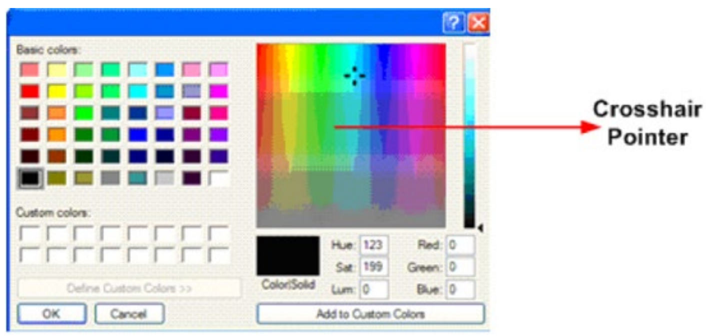


ACS supports multiple patterns in a single wafer. All patterns share the same wafer map, but they can have their own test plans, including different sites, subsites, devices, and test modules. You can use assorted colors to distinguish patterns.

To select a pattern:

1. Double-click the Pattern in the configuration navigator.
2. Select Color (to the right of the pattern name), and the Color dialog opens. You can choose either a basic color or customize your own color.

Figure 37: Color choice pointer



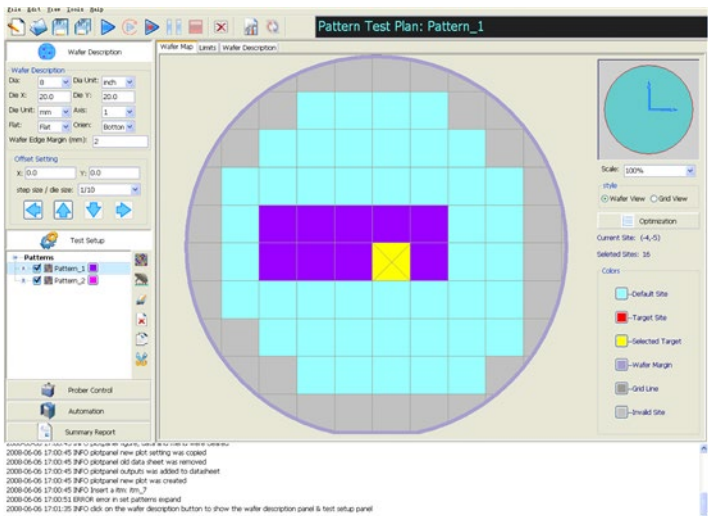
3. To choose a basic color, select one of the color squares. To choose a custom color, drag the crosshair pointer onto the color. The selected color is displayed in Custom colors below the Basic colors.
4. Select **OK**. The color is assigned to the selected pattern.

Display wafer pattern

One wafer can display the selected sites of a pattern, of several patterns, or all patterns. Every pattern has a color. You can choose an assorted color for each pattern. If you clear a pattern in the Test Setup configuration navigator, the pattern does not display in the Wafer Map and may not be available for testing.

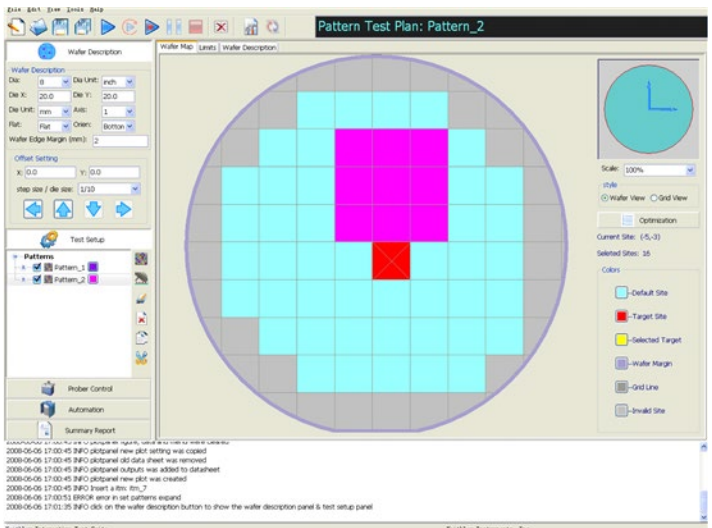
As shown in the next figure, the Wafer Map of Pattern_1 is selected and its sites are displayed. Pattern_2 is not selected, which means the sites are not displayed in the Wafer Map.

Figure 38: Wafer display Pattern_1



You can select both Pattern_1 and Pattern_2. If both patterns are selected, the sites of both patterns are displayed. In the following figure, only Pattern_2 is displayed.

Figure 39: Display Pattern_1 and Pattern_2 sites



Circle-shaped wafer map

After a pattern is selected, the Wafer Map is displayed in the work area in the ACS main dialog. The Wafer Map displays the customized wafer in the Wafer Description panel and captures and sets the size and location information where the test structures are to be probed.

All the information in the Wafer Map is in accordance with the pattern you have selected.

You can select a site that is under test by selecting the Wafer Map.

To select a site that is under test:

1. Clear the site by selecting it. Multisites (or blocks of sites) may be selected or not selected by selecting and dragging. For more, see the Site selection by dragging (on page 4-24) topic.
2. Choose from the options displayed in the menu: Select Sites, Clear Sites, Erase Sites, or Recover Sites.
3. Right-click the Wafer Map work area to display the menu displays.

The following table describes the options available in this menu.

Wafer Map Menu Item	Description
Select All	Selects all possible sites of the wafer.
Clear All	Clears all sites that are selected.
Set As Target	Sets the selected site as the target site, which defines the first stopping point after alignment.
Set As Reference	Sets the selected site as the reference site, which defines the origin of site coordinates.
Set As Target With (X, Y)	Sets the selected site as the target site, which defines the position of the target site (X,Y).
Set As Erased Site	Mark sites as bad (erased). No tests are performed on erased sites, but they are available for operations such as binning. The erased site is gray on the wafer map.
Recover the Erased Site	Recovers and returns the erased site.
Recover all Erased Sites	Recovers and returns all erased sites.
Allow Partial	Allows the collection of sites that do not fall completely within the usable wafer boundaries to be selectable.
Disallow Partial	Only allows the collection of those sites that are fully within the usable wafer boundaries to be selectable.
Define Logical Site	Allows you to define the attributes of a predetermined logical site (for applications such as parallel testing).

Allow Partial and Disallow Partial

Choosing Allow Partial enables you to test all the dies that are both whole and partial. Partial dies are that fall within the outline of the proposed wafer shape.

When Disallow Partial is selected, you are prevented from testing the partial dies. However, the full dies in the wafer shape can be tested. When Disallow Partial is selected, the wafer shows the full dies in the color cyan and the partial dies in gray.

Figure 40: Disallow Partial dies

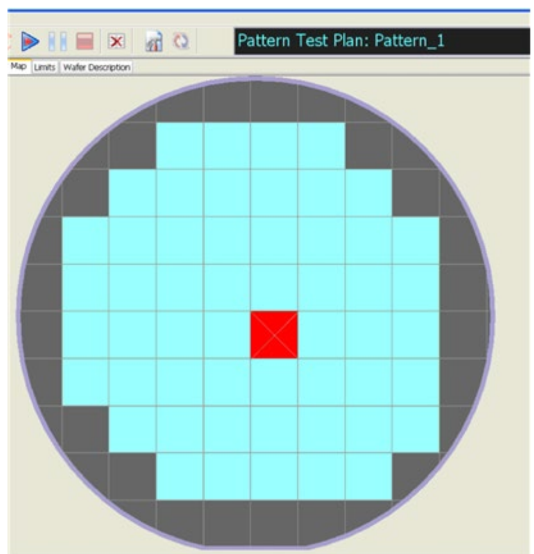
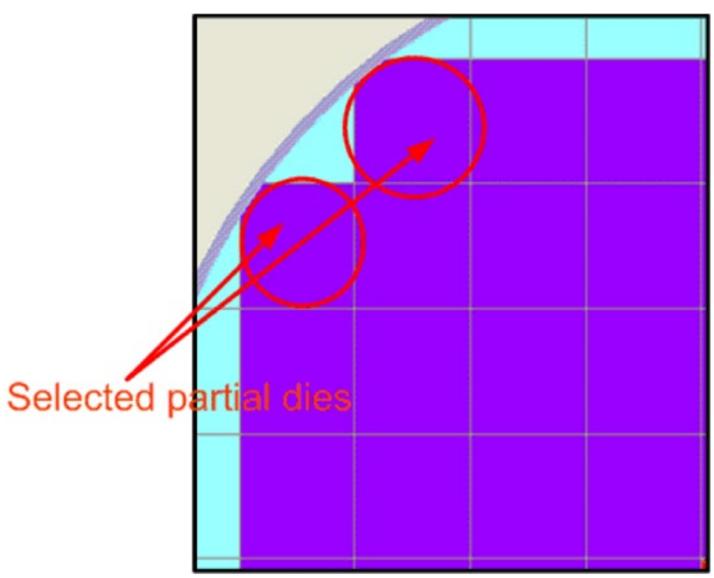


Figure 41: Selected partial dies

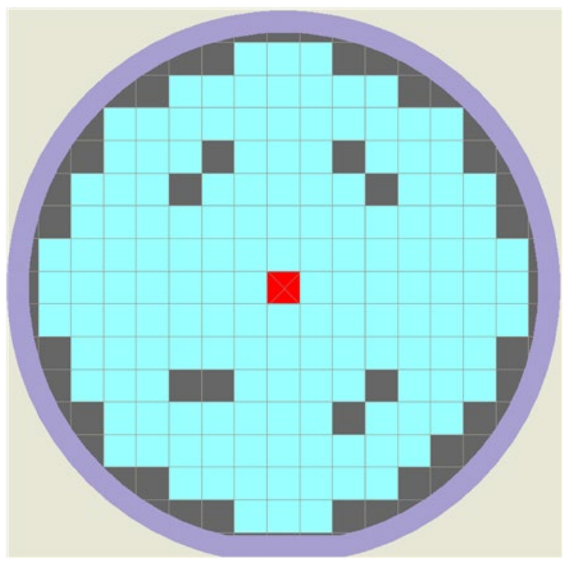


Erased sites

You can erase multiple sites from the wafer map so that they cannot be selected for testing. After you run automation, you can choose to mark those sites as erased.

To mark a site erased, right-click the site you want to erase, then choose Set As Erased Site. The selected site is set as erased and displayed in gray in the wafer map. ACS does not run tests on erased sites. To recover an erased site, select **Recover the Erased Site**. To recover all erased sites, select **Recover all Erased Sites**.

Figure 42: Wafer Map with erased sites



Logical sites

Physical sites are the sites on the wafer. The size and position of the sites are the same as the real wafer.

Logical sites are defined to run parallel tests on the wafer with several physical sites grouped together to simultaneously perform the same tests during the automation test process. The number of sites depends on how many groups of Series 2600B instruments you have. Such a group of physical sites is called a logical site. Logical sites allow testing of a group of physical sites with only one prober card touch, which sharply decreases the number of prober movements, which shortens the automation test time.

Defining a logical site

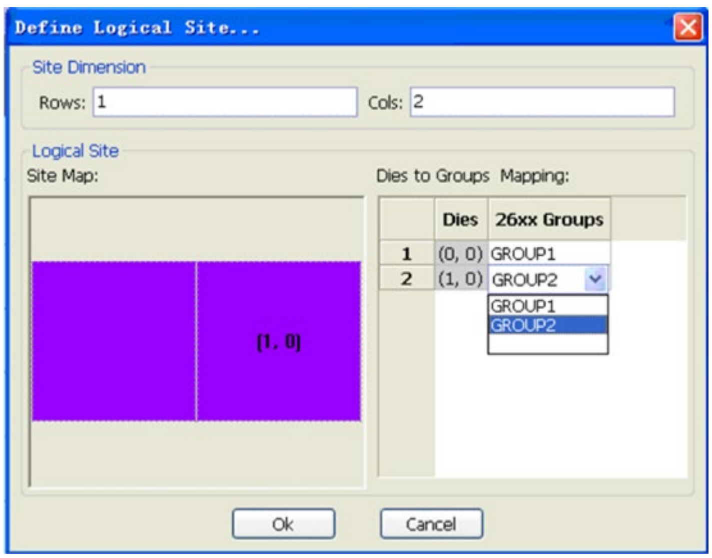
NOTE

If the row is set to 1 and the column is set to 1, the logical site is disabled.

To define a logical site:

1. Right-click the wafer map to display a menu. Select **Define Logical Site**.
2. Enter the **Site Dimension**. This sets how many physical sites are in the row and column of a logical site.

Figure 43: Define Logical Site



3. Set **Dies to Groups Mapping**. This determines the mapping of physical sites and Series 2600B groups.

NOTE

If you do not assign a physical site, the site is not chosen and no test is run.

4. Select **OK**.

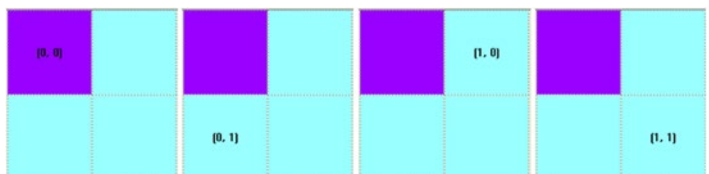
Defining wafer map with logical site

The collection of logical sites on the wafer map is like the physical sites.

When you select a die on the wafer map, it displays a logical site with an icon on the upper left site. The marked site is the reference site (0, 0) in this logical site. In the Define Logical Site dialog, if you move and hover your cursor over the defined sites, the reference position numbers and coordinates appear.

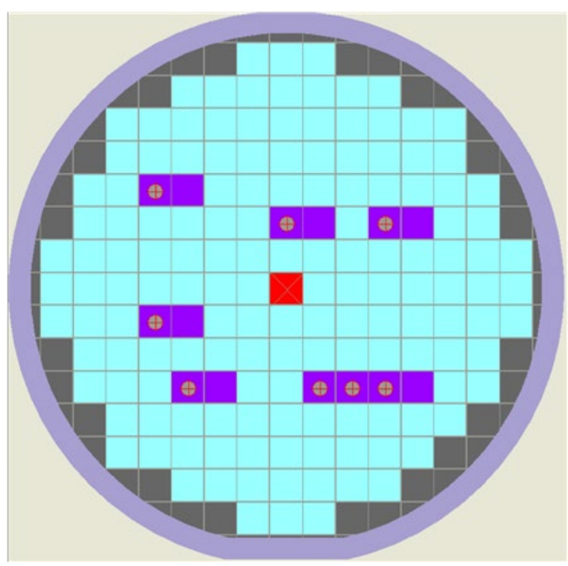
The following figure shows an example of a two-row, two-column logical site definition.

Figure 44: Logical site, two-row, two-column position example



The following figure shows an example of a one-row, two-column logical site definition that has been applied to a wafer map.

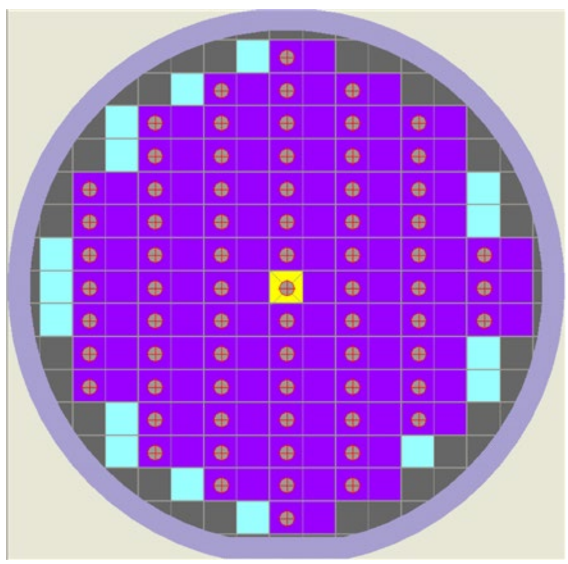
Figure 45: Logical site, one-row, two-column example



Guidelines for working with sites:

- If you select the upper left site (marked) of a selected logical site, it is not selected.
- If you select unmarked sites, another logical site is selected with a new upper-left site marked.
- When you click and drag the sites, you can create a collection of selected logical sites.
- If you choose Select All, the software calculates and covers as many physical sites as possible with logical sites.
- If you want to cover all the sites on the wafer, select the wafer map to select them all.
- After defining the wafer map with logical sites, you can run a parallel test in automation, which tests several sites at the same time with different Series 2600B groups.

Figure 46: Logical site, Click All command example

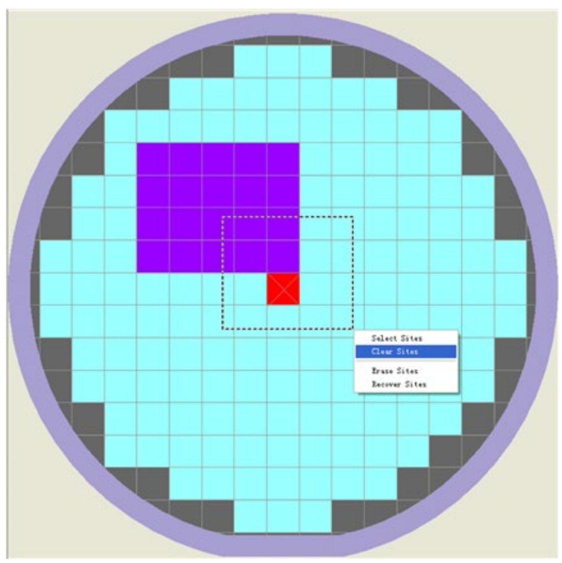


Site collection by dragging

You can drag to choose or clear several sites on the wafer map. To drag, select a site using the left mouse button, then hold the mouse button and drag around the sites to select. A Site Selection menu is displayed with the following options:

- **Select Sites:** Chooses all sites inside the rectangular area.
- **Clear Sites:** Clears all sites inside the rectangular area.
- **Erase Sites:** Marks all sites inside the rectangular area as invalid.
- **Recover Sites:** Recovers all previously marked invalid sites inside the rectangular area.

Figure 47: Clear Sites collection example



Circle-shaped wafer map advanced settings

The advanced wafer map settings in the right panel of the wafer map include:

- Wafer Map Scale (on page 4-16) (100% to 500%)
- Wafer Map Style (on page 4-17) (Wafer View or Grid View)
- Sites Sequence Optimization (on page 4-17) (select for several options)
- Sites indicator (on page 4-19) (current site and selected sites)
- Wafer Map Color Editor (on page 4-20) (six color choices)

Wafer Map Scale

Using the Wafer Map Scale setting, the wafer map view can be enlarged by increasing the Wafer Map Scale input. For instance, if the Scale is set to 200%, the wafer map view is displayed at 200% of the original, with only part of the wafer displayed.

NOTE

The axis direction of the wafer map be changed using the Axis element under Wafer Description.

Figure 48: Wafer Map scale

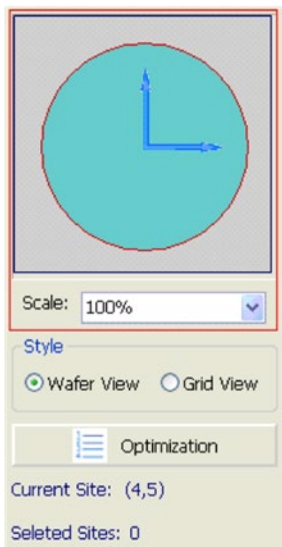
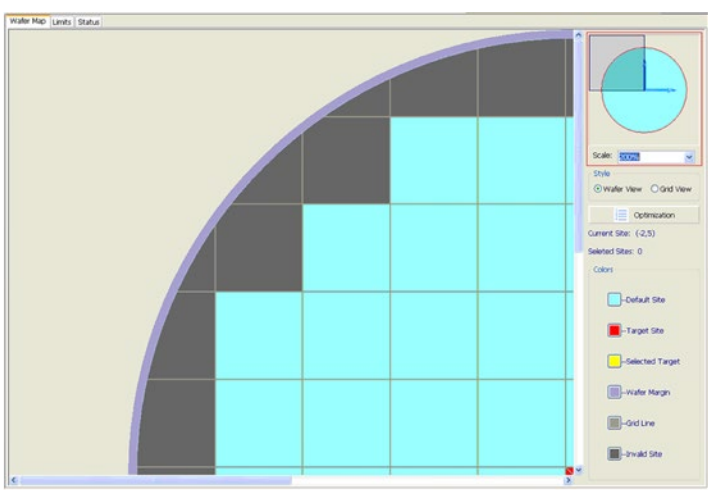


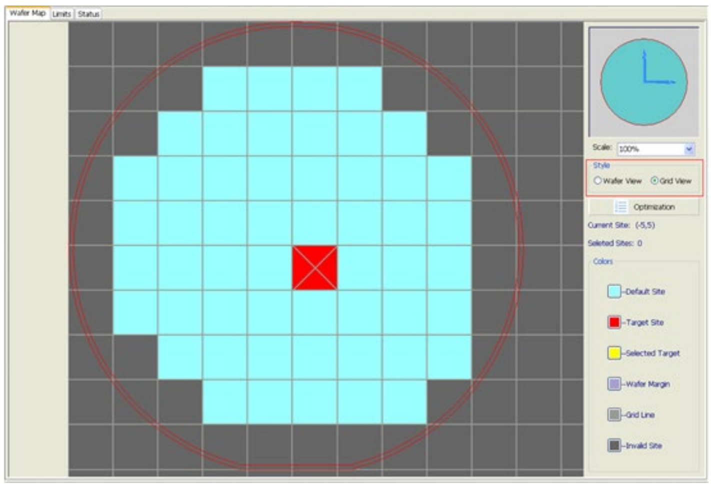
Figure 49: Wafer Map scale close view



Wafer Map Style

The default style of the wafer map is the Wafer View. You can change it to Grid View in the Style setting. Choosing Grid View displays all the grids in the background. The difference between Wafer View and Grid View is that you can set a reference site outside the wafer boundary by referencing the grid beyond the edge of the wafer.

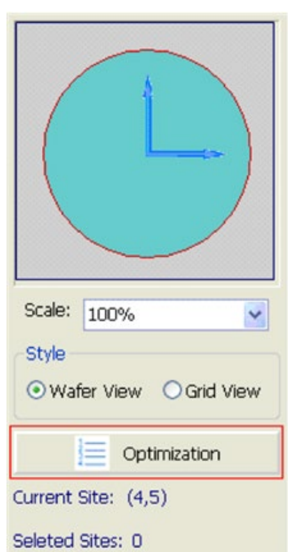
Figure 50: Wafer Map style grid view



Sites Sequence Optimization

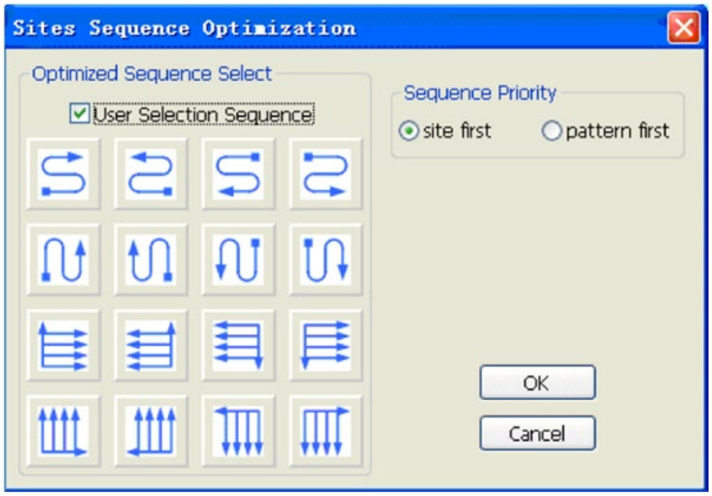
The Sites Sequence Optimization feature is accessed by selecting Optimization.

Figure 51: Optimization function



After selecting Optimization, a dialog displays that allows you to set the sites sequence characteristics for automation tests.

Figure 52: Sites Sequence Optimization

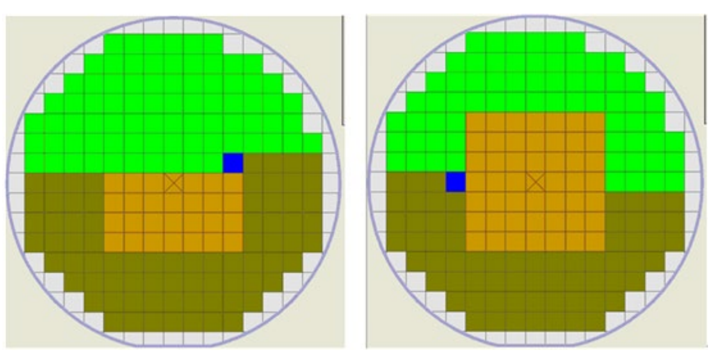


When a sequence is selected, the test initiates, following the directional sequence of the selected pattern.

For example, if the icon is selected, the test starts from the lower left corner site and sweeps left to right in an upwards direction. Select **OK** to confirm the sequence.

In the sequence priority area, you can select site first or pattern first. These two modes are used when more than two patterns are selected in the test plan. Site first means that the prober does not move to the next site until all patterns in a selected site have been finished. Pattern first means that the prober does not move to Pattern2 until all sites in Pattern1 are finished.

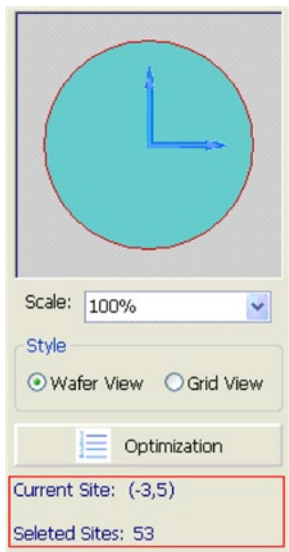
Figure 53: Sites Sequence left image first, right image pattern first



Sites indicator

Below the Optimization function is the Current Site indicator. The Current Site indicator displays the coordinates of the site where the last mouse selection or drag action was performed on the wafer map. The Selected Sites indicator shows the total number of presently selected sites on the viewable wafer map.

Figure 54: Sites indicator



Wafer Map Color Editor

The options under Colors, shown in the following figure, changes the color settings of a wafer map.

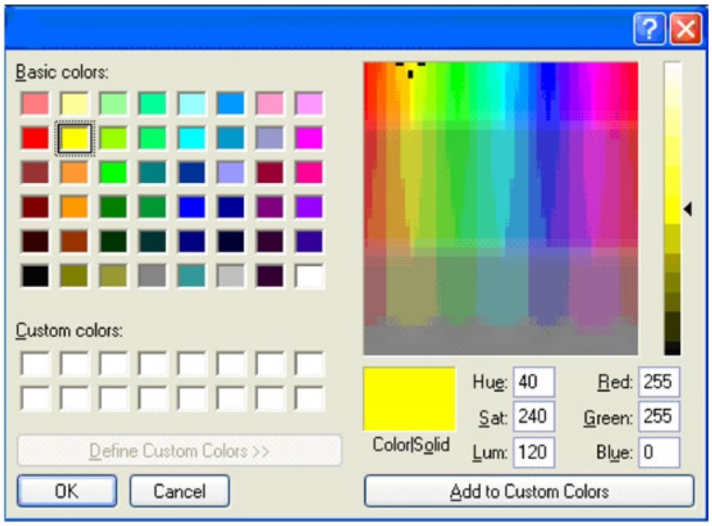
Figure 55: Wafer Map Colors editor

The options are:

- **Default Site:** Sets and shows the color of the wafer background.
- **Target Site:** Sets and shows the color of the target site.
- **Selected Target:** Sets and shows the color of the selected target site, indicating whether the target site is selected or not.
- **Wafer Margin:** Sets and shows the color of the wafer margin.
- **Grid Line:** Sets and shows the color of the dividing grid lines.
- **Invalid Site:** Sets and shows the color of the invalid sites.

When you select one of the Colors, the color dialog is displayed. To change the color, select the new color, then select **OK**.

Figure 56: Color choice



Square-shaped wafer description

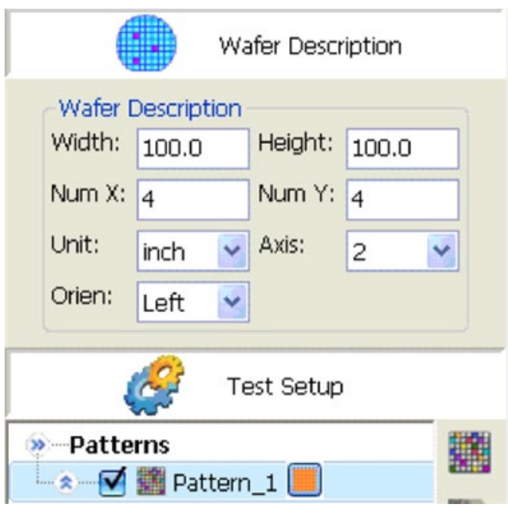
If you want to use a square-shaped wafer, the Wafer Description panel is modified to allow for defining its specific attributes. To create a square-shaped wafer, select **Tools > Preferences > Prober**. Refer to Wafer introduction (on page 4-1) for more information.

The wafer description functions are on the edit panel in the ACS GUI. You can edit the wafer information in this panel and the results immediately display in the ACS GUI work area. Review the following table for wafer description editing elements and their corresponding descriptions.

NOTE

The Orien operation does not change the sites or subsites. It only marks the orientation of the alignment.

Figure 57: Square-shaped Wafer Description



Wafer Description Elements	Description
Height	Height of the square-shaped wafer.
Width	Width of the square-shaped wafer.
Num X	Site number in the X direction.
Num Y	Site number in the Y direction.
Unit	Unit of length, in inches or millimeters.
Axis	The axis direction of the squared-shaped wafer.
Orien	The direction of the alignment.

Wafer pattern and display pattern

For square-shaped wafer, operations on the wafer pattern and display pattern are the same as that of the circle-shaped wafer.

Square-shaped wafer map

After a pattern is selected, the square-shaped Wafer Map is displayed in the work area of the ACS main dialog. It is used to display the customized wafer in the description panel, and to capture and set the size and location information where the test structures are to be probed.

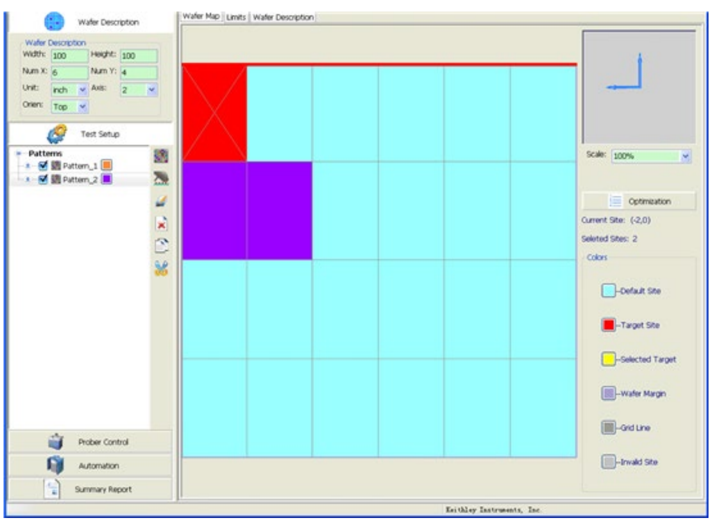
All the information in the Wafer Map is in accordance with the pattern you have selected.

You can select a site that is under test by selecting the Wafer Map.

To select a site that is under test:

1. Clear the site by selecting it. Multisites (or blocks of sites) may be selected or not selected by selecting and dragging. For more, see the Site selection by dragging (on page 4-24) topic.
2. Choose from the options displayed in the menu (Select Sites, Clear Sites, Erase Sites, or Recover Sites).
3. Right-click the Wafer Map work area (a menu displays).

Figure 58: Square-shaped Wafer Map



The following table describes the options available in this menu.

Wafer Map items

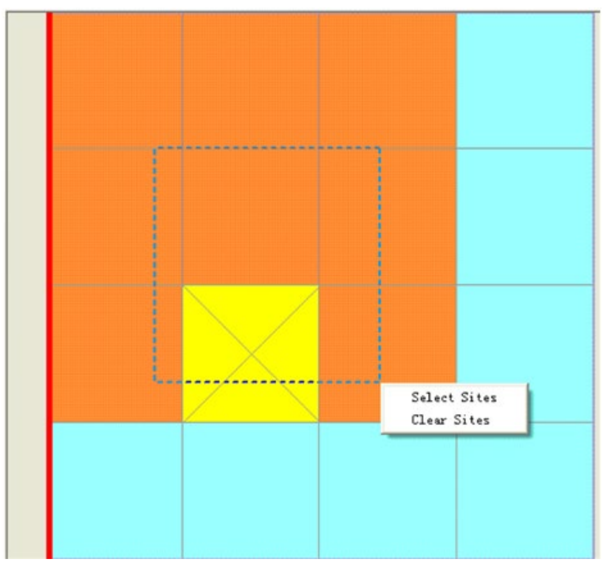
Wafer Map menu item	Description
Select All	Selects all possible sites of the wafer.
Clear All	Clears all selected sites of the wafer.
Set As Target	Sets the selected site as the target site, which defines the first stopping point after alignment.
Set As Reference	Sets the selected site as the reference site, which defines the origin of site coordinates.
Set As Target With (X, Y)	Sets the selected site as the target site, which defines the target site's position (X,Y).

Site selection by dragging

Site Selection by dragging allows you to choose (or clear) several sites on the wafer map by clicking the left mouse button on the site, holding down the mouse function, and dragging a box around the sites you want selected. The choices in the Site Selection menu are:

- **Select Sites:** Select the sites inside the rectangular area.
- **Clear Sites:** Clear the selected sites inside the rectangular area.

Figure 59: Clear Sites selection example



Square-shaped wafer map advanced settings

The advanced wafer map settings are on the right panel of the wafer map, and include:

- Wafer Map Scale (on page 4-16)
- Sites Sequence Optimization (on page 4-17)
- Sites indicator (on page 4-19)
- Wafer Map Color Editor (on page 4-20)

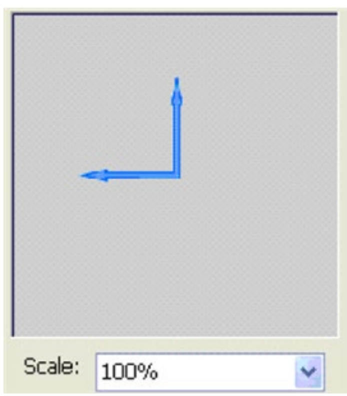
Wafer Map Scale

Using the Wafer Map Scale setting, the wafer map view can be enlarged by increasing the Wafer Map Scale input. For instance, if the Scale is set to 200%, the wafer map view is displayed at 200% of the original, with only part of the wafer displayed.

NOTE

The axis direction of the wafer map be changed using the Axis element under Wafer Description.

Figure 60: Square-shaped Wafer Map Scale



Sites Sequence Optimization and Sites Indicator

The Sites Sequence Optimization feature and Sites Indicator are the same as the Circle-shaped wafer map (on page 4-9).

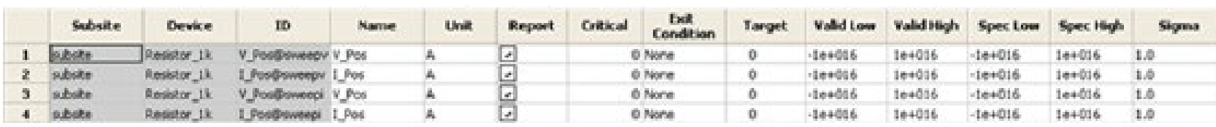
Wafer description limits and status tab

Limits refers to the limits data entry area. The limits data associated with one parameter is arranged as one row. The wafer description tab displays the wafer description and status.

Wafer limits tab

Limits refers to the limits data entry area. The limits data associated with one parameter is arranged as one row.

Figure 61: Limits tab



	Subsite	Device	ID	Name	Unit	Report	Critical	Exit Condition	Target	Valid Low	Valid High	Spec Low	Spec High	Sigma
1	subsite	Resistor_1k	V_Pos@sweepv	V_Pos	A	☑	0	None	0	-1e+016	1e+016	-1e+016	1e+016	1.0
2	subsite	Resistor_1k	I_Pos@sweepv	I_Pos	A	☑	0	None	0	-1e+016	1e+016	-1e+016	1e+016	1.0
3	subsite	Resistor_1k	V_Pos@sweepi	V_Pos	A	☑	0	None	0	-1e+016	1e+016	-1e+016	1e+016	1.0
4	subsite	Resistor_1k	I_Pos@sweepi	I_Pos	A	☑	0	None	0	-1e+016	1e+016	-1e+016	1e+016	1.0

The limits tab fields are as follows:

- **Subsite:** The subsite name defined in configuration navigator. This field cannot be modified.
- **Device:** The device name defined in configuration navigator. This field cannot be modified.
- **ID:** The unique ID for the test parameter. This field cannot be modified.
- **Name:** The Name field provides for a user-friendly name that describes the parameter more clearly than the ID. This field cannot be modified.
- **Units:** Defines the units for the limits test results. This field may contain nonalphanumeric characters (including spaces) and does not need to be a C programming language name. The only constraint is that it must not include commas. The string may contain a maximum of 13 characters. Examples of entries for this field are volts or amps. The default value is `A`.
- **Report:** Determines whether the limit is included in the summary report. To include the report, select this option. To remove it from the report, clear it.
- **Critical:** Allows parameters to be tagged as not critical (0) or as critical with one of nine critical flags (1, 2, 3, 4, 5, 6, 7, 8, and 9).
- **Exit condition:** You can define what Exit action is taken when a failure of the corresponding parameter is found.

- **Target:** Identifies the ideal result expected for the specified result ID. The entry for this field is a numeric value.
- **Valid Low, Valid High:** This pair of limits can be used to check for exit conditions in the testing system. Test results outside this range indicate that the data being produced by the system is invalid. Invalid data triggers the specified exit condition.
- **Spec Low, Spec High:** This pair of limits is used to check for manufacturing standards. Test results outside the specifications range indicates that the device being tested does not meet manufacturing specifications. When run in manual mode, if the test result is out of the specifications range, the background color of the data cell is defined in the Manual Run Fail Color Setting (see Fail data (on page 5-249) for more information). When run in automation mode, if the test result is out of the specifications range, the site in the wafer map will display a color indicating a failure (see Wafer map indicator (on page 7-33) for more detail).
- **Sigma:** Parameter used to show the statistical measurement of variability, showing how much variation exists from a statistical average.

Exit condition

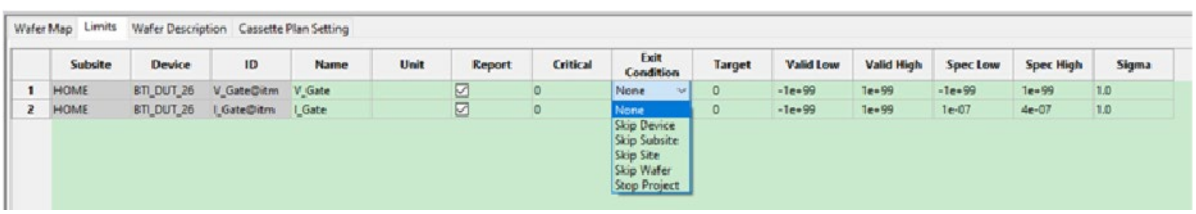
The exit condition column allows you to set the action to take when a failure of the corresponding parameter is found. There are several exit actions:

- None (default)
- Skip Device
- Skip Subsite
- Skip Site
- Skip Wafer
- Skip Project

In the Limits tab under Wafer Description is the Exit Condition column. You can set the parameter range in the Valid Low, Valid High, Spec Low, and Spec High fields in the same area.

There is no Skip Test, because the tests are all run in the Lua engine of the Series 2600B instrument, so on the software level, test flow cannot be controlled.

Figure 62: Limits tab Exit Condition

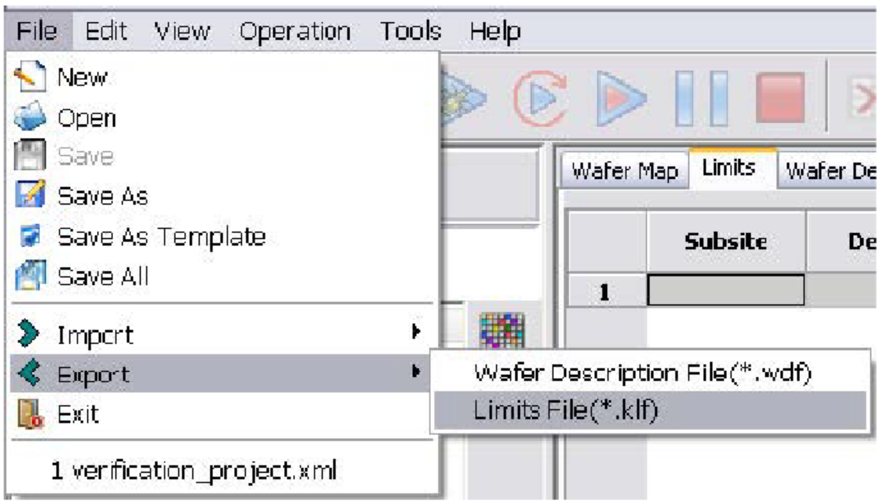


Exporting limits files

There are two ways to export the limits file (`.klf` file):

- Select **Export** > **Limits File (*.klf)** in the File menu.
- Select **Export** at the bottom of Limits tab in the Wafer Description.

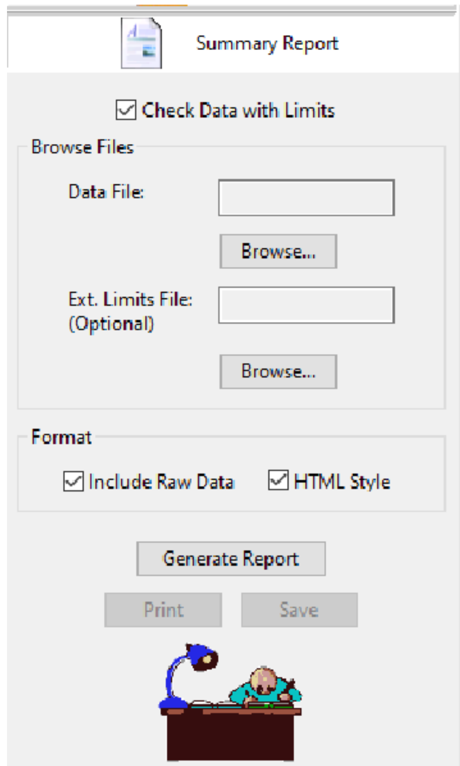
Figure 63: Export Limits File



Using external limits files

The Summary Report in the View menu retrieves data depending on your selections.

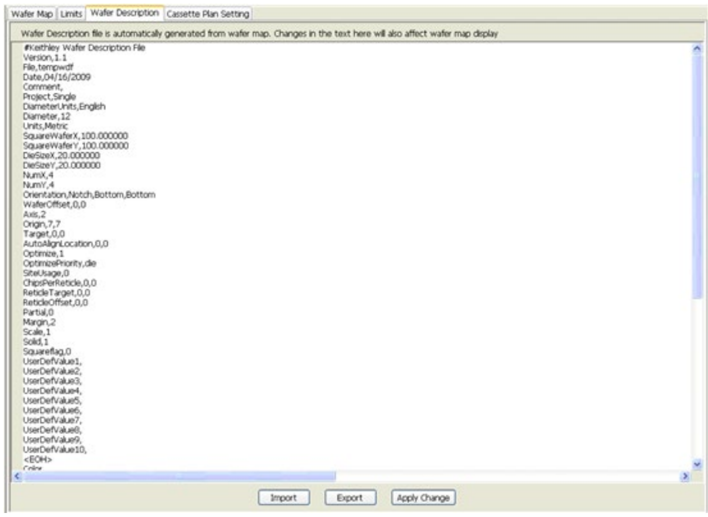
Figure 64: ACS 4.2 Exit Limits File



Wafer description and status tab

This tab displays the wafer description file and the status.

Figure 65: Wafer Description tab

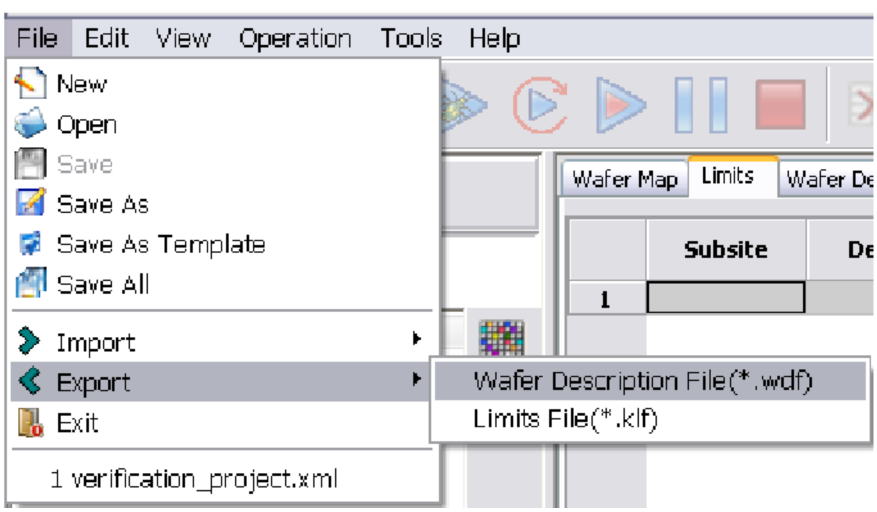


Exporting wafer description files

There are two ways to export the wafer description file (`.wdf` file):

- Select **Export** > **Wafer Description File (*.wdf)** in the File menu.
- Select **Export** at the bottom of the Wafer Description tab.

Figure 66: Export .wdf or .klf files



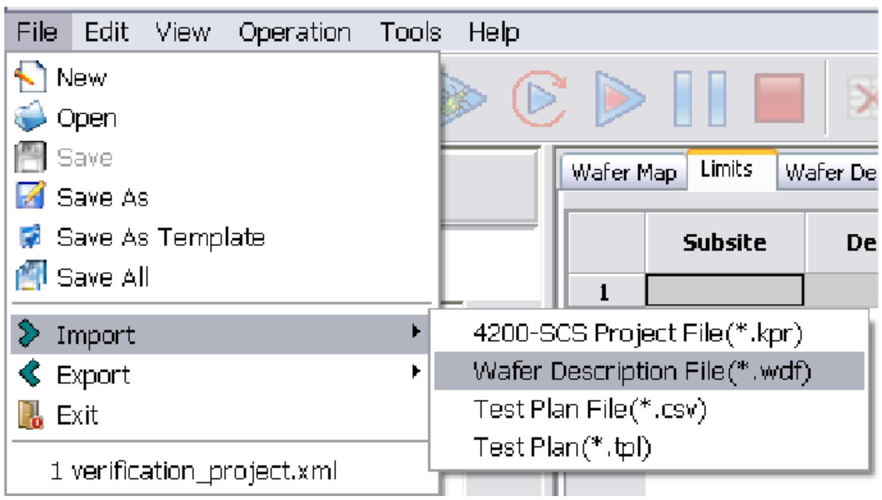
After you export a `.wdf` file and save it to a specified folder, you can import this file for use in other tests, eliminating the need to define a wafer every time you build a test plan.

Importing wafer description files

There are two ways to import the Wafer Description file (`.wdf` file):

- Select **Import** > **Wafer Description File (*.wdf)** in the File menu.
- Select **Import** at the bottom of the Wafer Description tab.

Figure 67: Import .kpr, .wdf, .csv, .tpl files



You can also change the content of the Wafer Description and select **Apply Change**.

Section 5

Test setup

In this section:

Configuration navigator introduction

ACS organizes tests in the following hierarchy, consistent with semiconductor-wafer organization. The configuration navigator is in the following logical hierarchy:

- Pattern (on page 5-2) (multipattern)
- Subsites (on page 5-2)
- Devices (on page 5-2) (multigroup)

- Graphical Interactive Test Module (ITM) (on page 5-4)
- Script Test Module (STM) (on page 5-4)
- C Language Test Module (CTM) (on page 5-5)
- Python Language Test Module (PTM) (on page 5-5)

Pattern

At the macroscopic level, one or more semiconductor sites are built up at a given wafer location. This location is comprised of a product site and usually has one or more parametric test structures or subsites. ACS supports multiple patterns of sites on a single wafer. All patterns share the same wafer map, but they can have their own test plans, which includes different site locations, different subsite locations, and test modules. You can use assorted colors to distinguish different patterns. ACS supports up to eight patterns in total.

Subsites

The terminals of each device on a test structure are connected to a uniformly-spaced series of contact pads. These pads connect the devices to the probes of a prober. Every wafer location that the probe moves to and contacts at any time is called a subsite. Sometimes ACS refers to each such test structure (the combination of test devices that are tested as a group) as a subsite.

Devices

Each subsite contains a series of devices to be characterized, such as transistors, diodes, resistors, and capacitors. A multigroup test is supported on a device. All selected groups of devices are tested simultaneously by a different Series 2600B group.

SMU safety and warnings

WARNING

A safety shield must be used whenever hazardous voltages (>30 V_{RMS}, 42 V_{PEAK}) will be present in the test circuit. To prevent electrical shock that could cause injury or death, never use a Keithley SourceMeter® instrument in a test circuit that may contain hazardous voltages without a properly installed and configured safety shield. The following procedures show the use of a metal safety shield. Connect the enclosure of the metal test fixture to protective earth (safety ground). If using a nonconductive test fixture, it must be rated to double the maximum capability of the test equipment in the system and the two green-yellow ground cables must be connected to protective earth (safety ground). Failure to attach the ground wires to a known protective earth may result in electric shock or death.

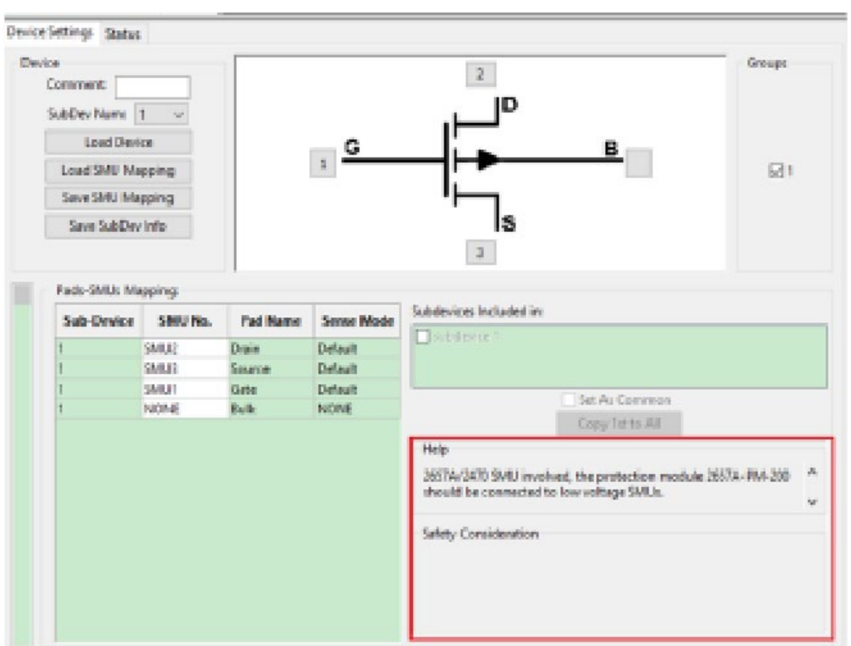
You can add medium-power SMUs and high-voltage SMUs in the same system.

If the 2657A high-voltage SMU is used with a 2611B, 2612B, 2635B, or 2636B, the following message is displayed:

```
2657A/2470 SMU involved, the protection module 2657A-PM-200 should be connected to
   low voltage SMU
```

See the documentation for the 2657A-PM-200 to make these connections.

Figure 68: SMU device settings



If there is a 2657A or 2470 in a system with 2601B, 2602B, 22450, 2460, or 2461 SMUs, there are no protection modules available. In this case, the tests under this device are disabled and the Safety Consideration area of the help displays the following message:

```
CRITICAL  It's not supported that the Model 260x(A/B), 2450, 2460,
2461 are used in system with the Model 2657A/2470 because the Model
2657A-PM-200 Protection Module only protects to 200V.
```

Test modules

There are four classes of tests or test modules in ACS:

- Graphical Interactive Test Module (ITM) (on page 5-4)
- Script Test Module (STM) (on page 5-4)
- C Language Test Module (CTM) (on page 5-5)
- Python Language Test Module (PTM) (on page 5-5)

All these test modules share common data and data plot analysis functions. The main differences between these modules are described in the following ITM, STM, CTM, and PTM topics.

Graphical Interactive Test Module (ITM)

An ITM allows you to define a test interactively in the user interface. It supports the Series 2600B System SourceMeter® and Series 2400 Standard Series SourceMeter®. You can also mix groups of 2600B instruments, 2400 Graphical Series SMU instruments, and 4200A SMUs.

Script Test Module (STM)

STM supports test modules programmed with LUA language and TSP™ LPT commands for testing with Series 2600B System SourceMeter® or Series 2400B SourceMeter® instruments. For details on programming a STM module with TSP LPT commands, refer to the *ACS Programmer's Manual*.

C Language Test Module (CTM)

A CTM can import test modules created by KULT. KULT is in the Clarius software that is loaded on the 4200A-SCS Parametric Analyzer. The test modules are programmed with C language and LPT for a 4200A-SCS and for testing connected instruments.

Python Language Test Module (PTM)

A PTM supports test modules and instruments programmed with the Python® programming language and ACS LPT commands. The Python test programs can be imported or compiled in the PTM interface. For details about programming a PTM module with ACS LPT commands, refer to the *ACS Programmer's Manual*.

Test process flow summary

There are two types of ACS test process flows:

1. Perform tests on only a few devices with no automated flow involved.
2. Perform tests throughout the cassette. When testing throughout the cassette, an autoprober is required.

Working with projects

This section describes how to use an existing project from the standard ACS library and how to create a new project.

CAUTION

This section assumes that hardware configuration is set and that all hardware is turned on. The software connections must accurately reflect the physical hardware connections at the time the test is executed. Incorrect terminal configurations result in anomalous test results or device damage. At every test startup, you must check whether the connections are matched correctly.

Using an existing project

When launched, ACS automatically opens the existing default project. To open another project, perform the following steps.

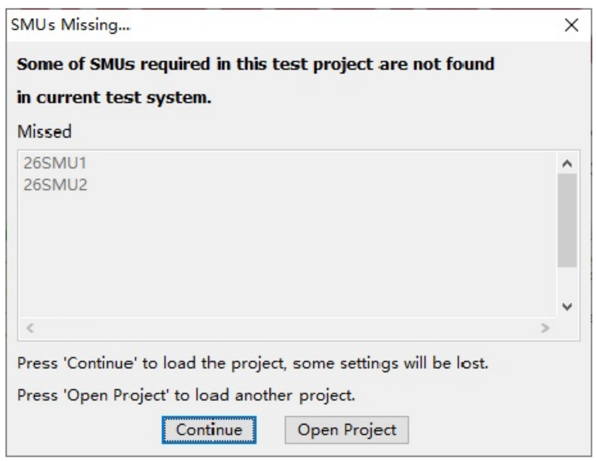
To open a project:

1. On the toolbar, select **Open Project**.
2. Select the project file you want to open. Project files are in `.xml` format.

If SMUs are required in the project you are trying to open, the ACS software tries to detect them in the present physical connection. Once the SMUs are found, the project is loaded in the configuration navigator in the Test Setup panel.

If the SMU connections in this project do not match the present physical connection, the 26xx SMUs Missing dialog opens, as shown in the following figure.

Figure 69: SMUs Missing



To find the missing SMUs, check all module settings in the project and reassign them where necessary until the SMUs Missing dialog no longer opens when you open the project.

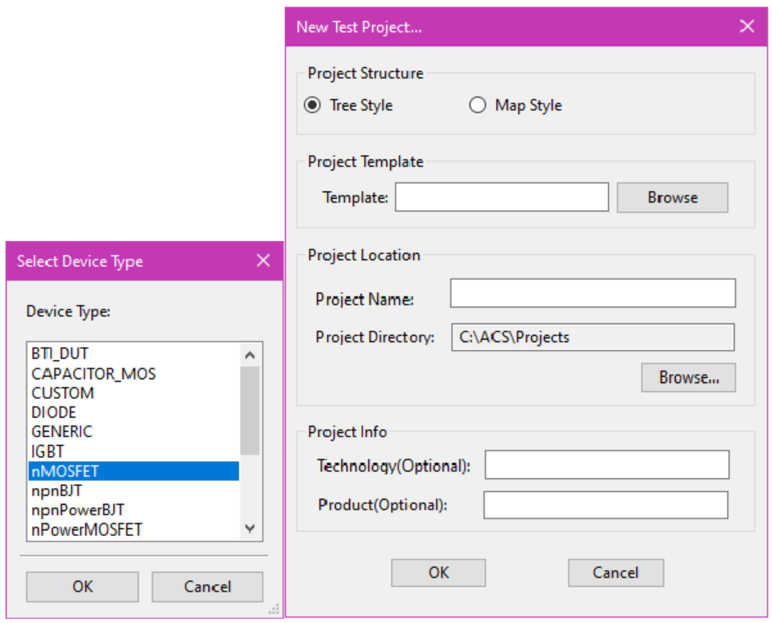
Create a new project using the configuration navigator

The standard naming convention for projects is to omit spaces. Instead, use an underscore to separate words in the name (for example, `CVU_CpGp_Compensation`).

The allowable characters for the Project name are: `A` through `Z`, `a` through `z`, `0` to `9`, and the underscore character (_) . The first character must be a letter (`A` through `Z` or `a` through `z`).

To create a new project using the configuration navigator:

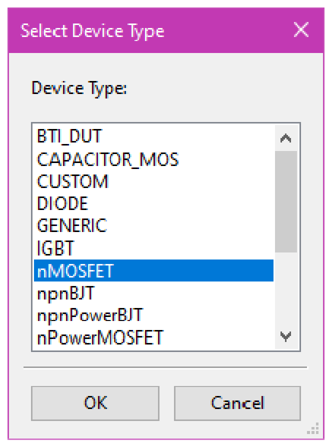
1. Select **New** on the toolbar from the ACS Main dialog (you can also select **New command** from the File menu). The New Test Project dialog opens.



2. In the Project Structure field, select the **Tree Style** or **Map Style** configuration navigator.
3. In the Template field, type the storage location or browse for a directory.
4. Enter the following information.
   - **Project Name:** Enter the new project name.
   - **Project Directory:** Select the storage folder for the Project, browse for a directory, or use the default directory.
   - **Technology:** Optional device information.
   - **Product:** Optional device information.

5. Select **OK**. The project is displayed in the Test Setup edit field. The default Test Setup information displays and includes:
   - **Pattern_1:** Default pattern name. The default pattern name can be changed to any string or color.
   - **HOME:** Default subsite name. The Home default subsite name can be changed to any string.
6. Select the **device** icon on the toolbar (or right-click **HOME** or the subsite to choose **Insert Device**). The SMU type dialog opens.
7. In the SMU Type dialog, select **26xx SMU** or **4200A SMU** and select **OK**. The Select Device Type dialog opens.

Figure 70: Select Device Type



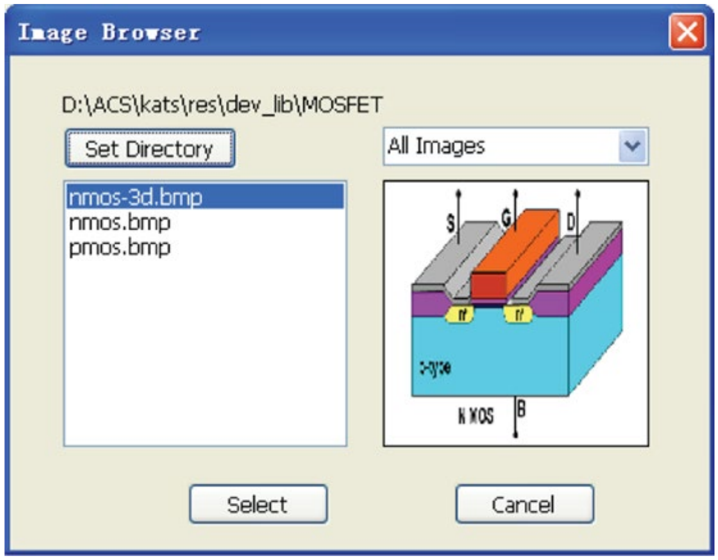
8. Select the device type to be tested and select **OK**. The Image Browser dialog opens.

NOTE

If you choose the 26xx SMU, the new device is named device_26. If you choose the 4200A SMU, the new device is named device_42.

9. Choose the device image that has a terminal connecting to the SMU. These images will appear in the test ITM, STM, PTM, or CTM work area.

Figure 71: Device images



10. Use **Select** to add a new device to configuration navigator.
11. Insert a Test Module by selecting the ITM, STM, PTM, or CTM button on the vertical toolbar.
12. Select the Group Number in the Groups area in the ITM Definition tab or on the STM Script tab.
13. Select **Save**.

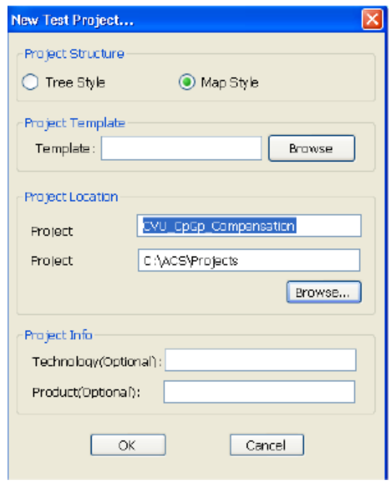
Create a new project using the map style

When creating a new project using the map style, follow the naming conventions described in Create a new project using the configuration navigator (on page 5-7).

To create a new map-style project:

1. Select **New** on the toolbar from the ACS Main dialog. The New Test Project dialog opens.

Figure 72: Map Style define new test project



2. In the Project Structure field, select **Map Style**.
3. In the Template field, enter the storage folder location or browse for a directory.
4. Enter the following information in the New Project dialog.
   - Project name
   - Project directory (**Browse** to find your project directory)
   - Technology (Optional): (device information)
   - Product (Optional): (device information)
5. Select **OK**.

6. The default Test Setup information displays and includes:
   - **Pattern_1:** Default pattern name. The default pattern name can be changed to any string or color.
   - **HOME:** Default subsite name. The Home default subsite name can be changed to any string.
   - **Devices:** Device nodes and test modules can be added to the Devices.

Figure 73: Map Style test setup panel

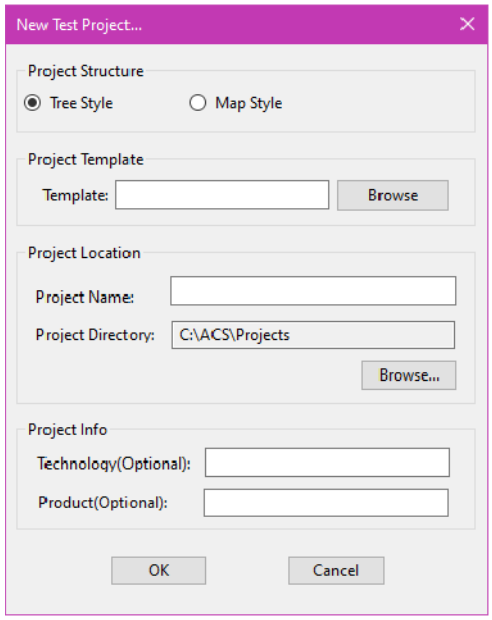
Create a new project from an existing .csv file

For each `.csv` file, it is better to create a new project.

To create a new project from an existing .csv file:

1. From the ACS File menu, select **File > New**. The software opens a dialog.
2. Enter a project name.

Figure 74: Create a new project for stress migration

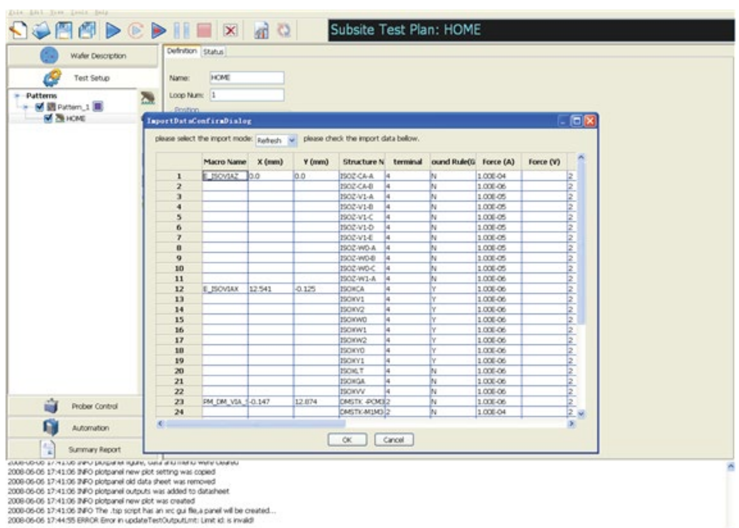


Import the SourceMeter .csv setting file

To import the .csv setting file:

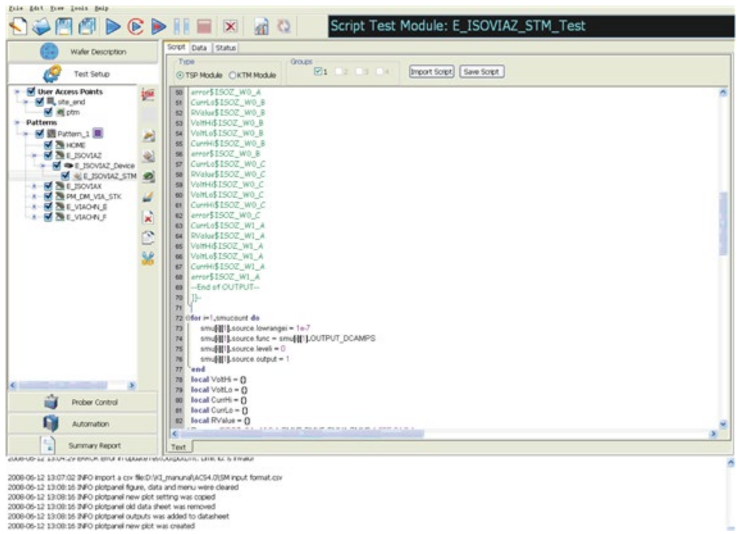
1. From the ACS File menu, select **Import > Test Plan File** (`*.csv`).
2. Select the `.csv` file. The Import Data Confirm dialog opens.
3. Verify that the displayed data file is selected.
4. Select the import mode:
   - **Refresh:** Replaces an existing test if it is the same in the test setup configuration navigator.
   - **Append:** Adds to the test setup configuration navigator.
5. Select **OK**.

Figure 75: Import Data Confirmed



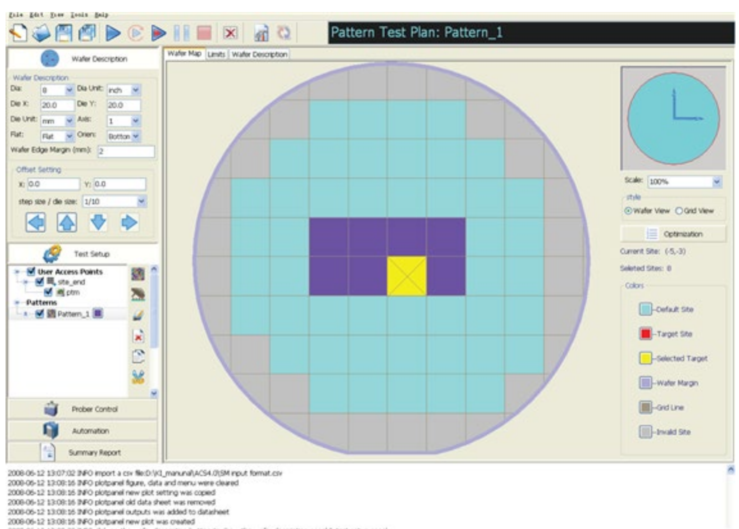
After importing the `.csv` data file, a test is created, including the automatically generated stress migration script test modules, as shown in the following figure.

Figure 76: Automatically generated test setup tree



6. Set the wafer map and select the sites to be tested in the wafer map.
7. Save the project.

Figure 77: Set the wafer map and select the site to be tested



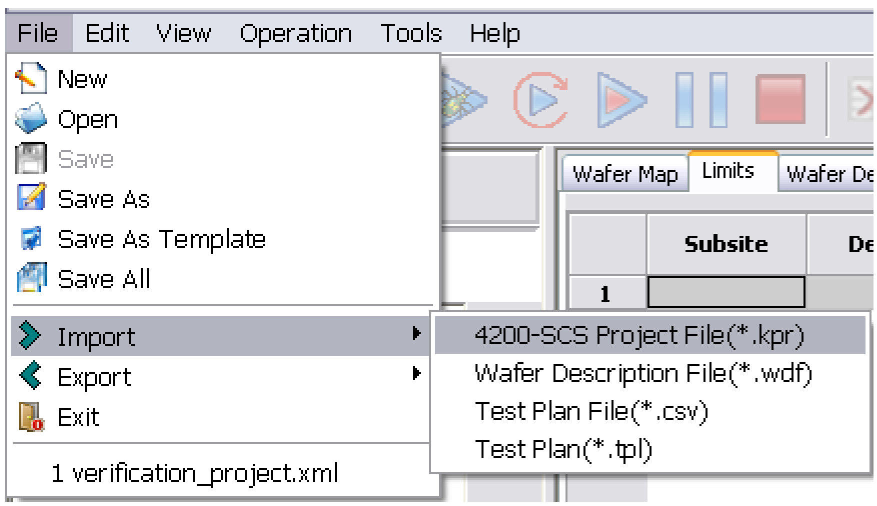
Import a Model 4200A project file

You can import a Model 4200A project file to ACS. The project configuration navigator may include several types of test modules, but ACS can only accept CTMs (C Language Test Modules, which are `.kpr` files). When you import the `.kpr` test module, it appends to the ACS configuration navigator at the subsite level.

To import a Model 4200A project file:

1. Select **File** > **Import** to import a file with a `.kpr` file extension, as shown in the following figure.

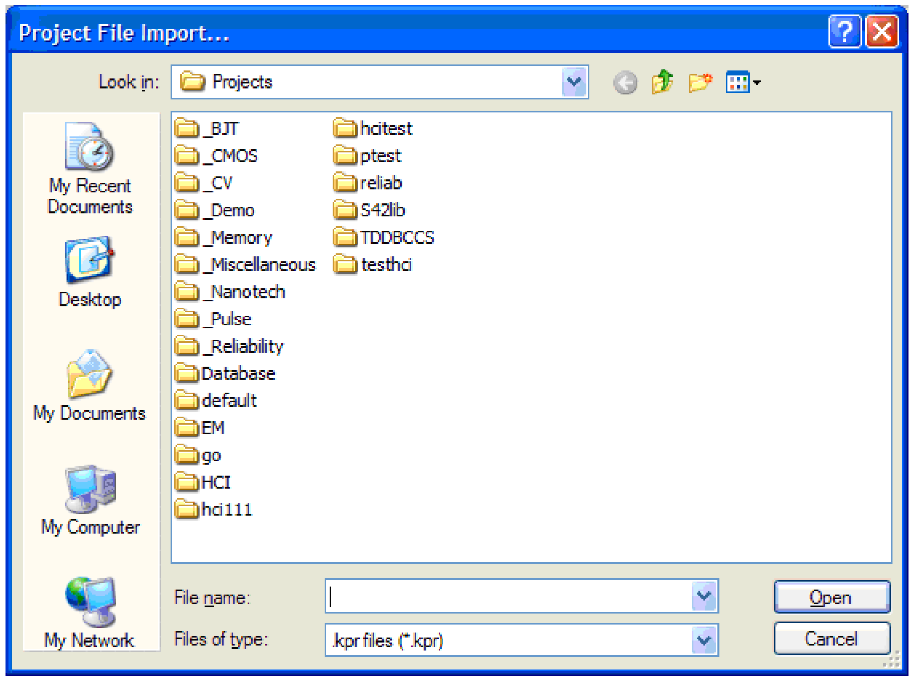
Figure 78: Import a Model 4200 project file



2. After selecting the `.kpr` file, the import project dialog opens.

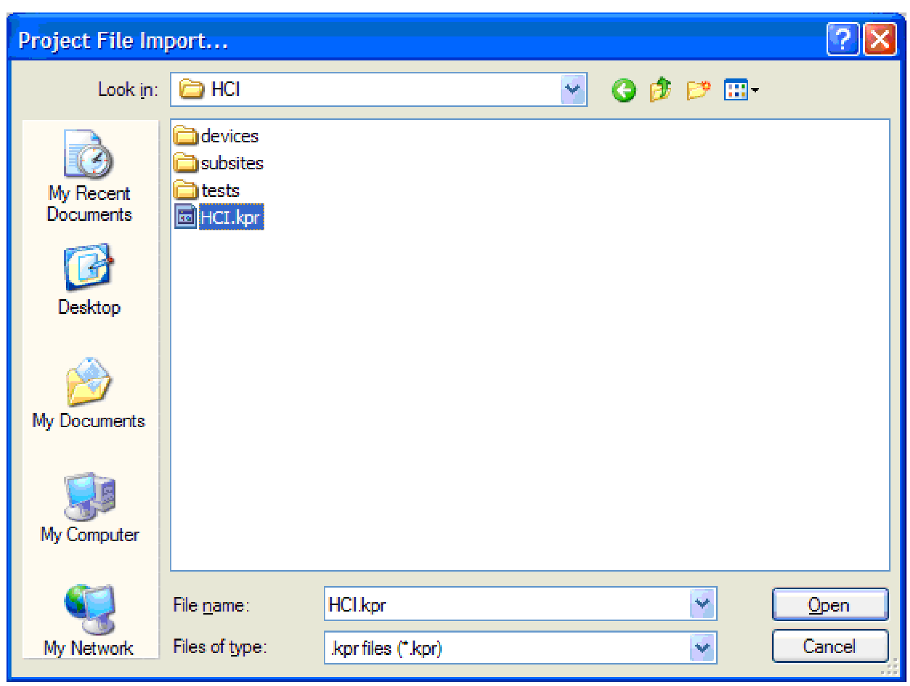
3. Select the project folder, then select **Open**.

Figure 79: Project dialog



4. Select the `.kpr` file and select **Open**.

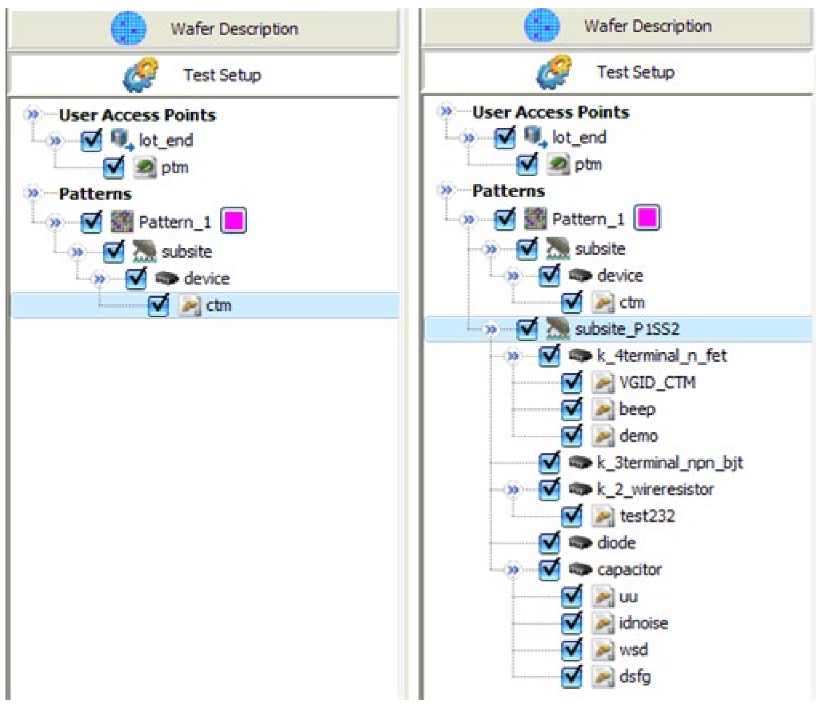
Figure 80: HCI Project File import dialog



After completing the previous steps, the subsite of the Model 4200A project you imported are appended to the configuration navigator of ACS.

The following figure shows the before and after configuration navigator images when importing a Model 4200A project file. The left image shows the configuration navigator before importing the project file. The right image shows the configuration navigator after you import the project file.

Figure 81: Test tree importing a project before and after views

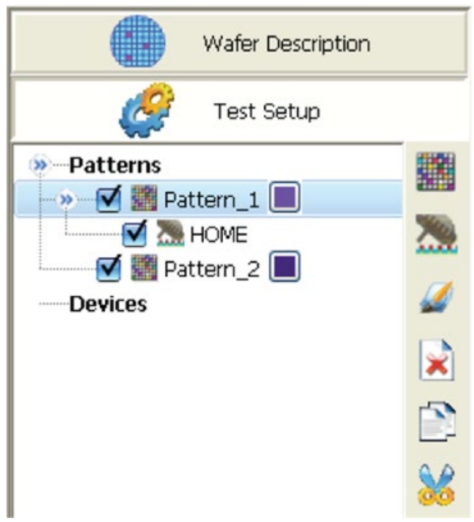


Add a pattern

To add a pattern:

1. Select **New Pattern**. A new pattern is created with the default name of Pattern_2 (sequential number of patterns created).
2. Use the vertical toolbar to rename, delete, copy, or cut the new pattern, as shown in the following figure.
3. Repeat step 1 to add extra patterns as needed.

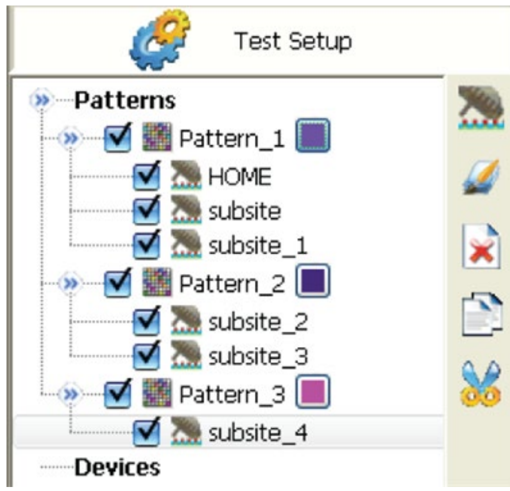
Figure 82: Selected Pattern_1



Add a new subsite

To add a new subsite

1. Select the Pattern for the new subsite.
2. Select **New Subsite**.
3. Repeat steps 1 and 2 to add extra subsites as needed.
4. Use the vertical toolbar to rename, delete, copy, or cut the new subsite.

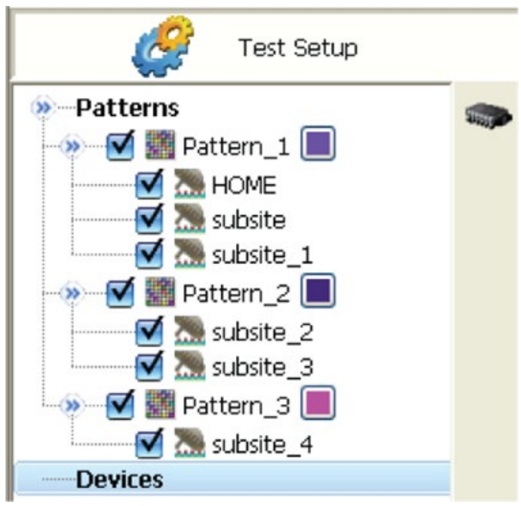


Add a new device

To add a new device

1. Select **Devices** from the Test Setup configuration navigator, as shown in the following figure.

Figure 83: Selected Devices tree item

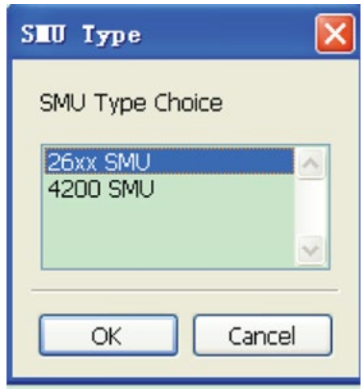


2. Select the **New Device**.
3. In the SMU Type dialog, select 26xx SMU or Model 4200A SMU, as shown in the following figure.

NOTE

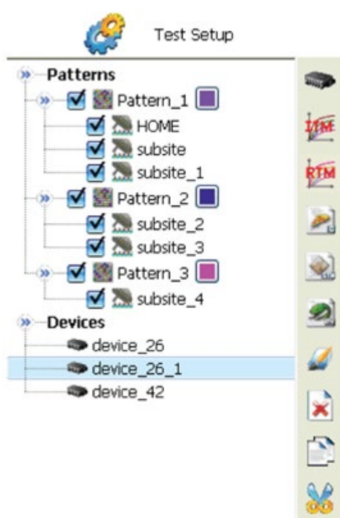
If you select the Model 26xx SMU, the new device name is appended with `_26`. If you select the Model 4200A SMU, the new device name is appended with `_42`.

Figure 84: SMU Type



4. Select **OK**. A dialog opens, allowing you to choose the type of device to test.
5. Choose one of the eleven devices.
6. Select **OK**. A new device is created.
7. Use the vertical toolbar to rename, delete, copy, or cut the new device.
8. Repeat steps 1 through 6 to add new devices as needed.

Figure 85: New Devices created

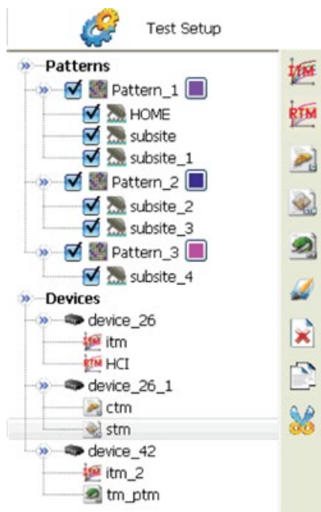


Add a new test module

To add a new test module

1. Select the device where you want to add a new test module.
2. Select a test module. A test module is created for the selected device, as shown in the following figure.
3. Use the vertical toolbar to rename, delete, copy, or cut the new test module.

Figure 86: Created test modules



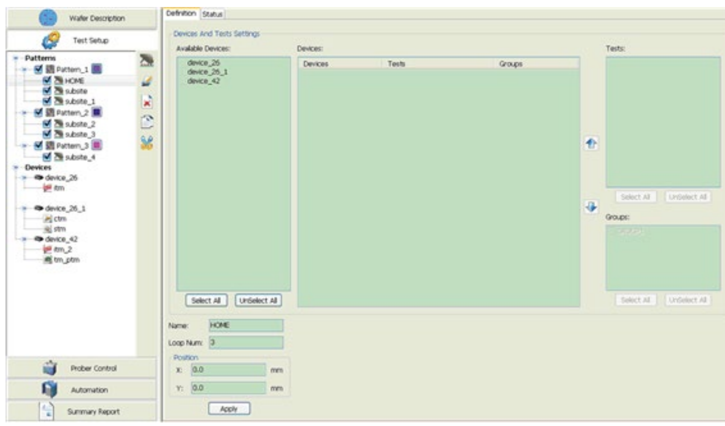
Map subsites and test modules

You can map any devices and test modules under the devices to the subsites. For example, map devices and test modules to the subsite Home (Home is the default subsite of a project).

To map subsites and test modules

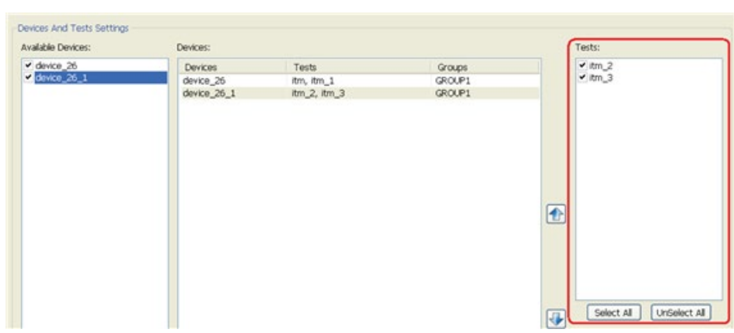
1. Select the subsite pattern you want to map. The definitions tab will display the devices and test settings, as shown in the following figure. There are four parts in the devices and test settings: Available Devices, Devices, Test, and Groups.

Figure 87: Devices and test settings



2. For the available devices, select or clear the device you want to include. You can use **Select All** or **UnSelect All**.
3. The tests box will show all the test modules for a selected device (the highlighted item in Available Devices), as shown in the following figure. All test modules are selected by default. You can clear the selected items to cancel or add a test module you want in your test.

Figure 88: Test Settings

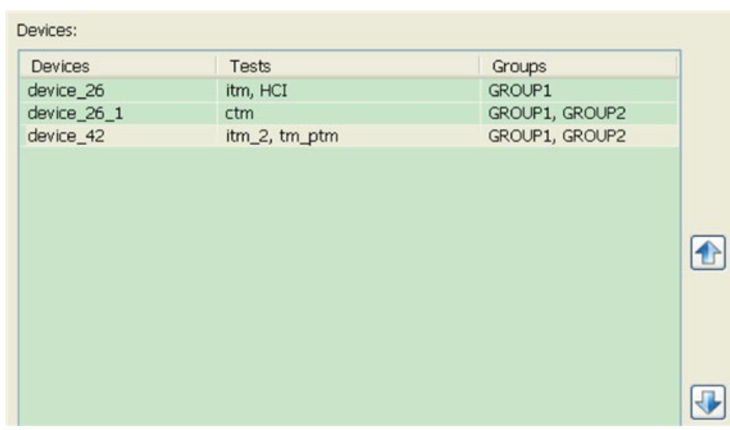


4. The groups are selected by default. Clear any item to remove from the test.

Figure 89: Group settings

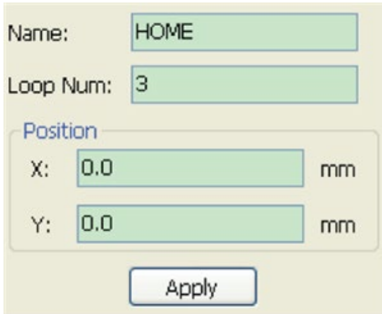
5. The devices section shows all the tests and groups you have selected. To change the test sequence, select the item you want to change and scroll up or down to move the test. For example, if you select `device_42` and scroll up, it will be tested first, then `device_26_1`, and `device_26` last.

Figure 90: Devices



6. Change the subsite name by entering a name in the Name field.
7. Set the Loop Num. You can also change the position by entering a specific X and Y axis value.

Figure 91: Name, loop number, and position settings



8. Select **Apply**.
9. Set any other devices and tests settings you may need for other subsites.
10. After mapping all the subsites, save the settings, then select **Automation** to run this project.

NOTE

In the Map style project, if you want to delete one device or one test module, you need not delete it in all the subsites containing this device or module. Select **Test Setup** and delete it in the device.

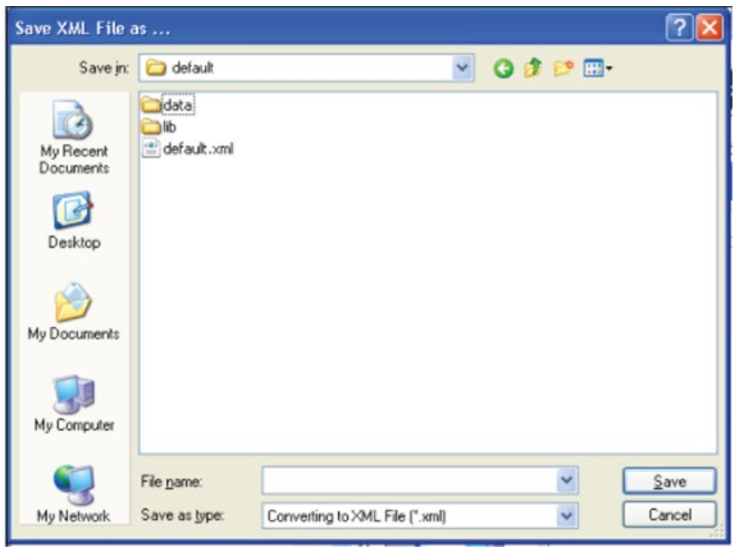
Save as a template and import a project template

The save as a template option makes creating a new test convenient. After the project template is saved, you can import a project into the main menu by choosing the Import Test Plan (`.tpl`) item from the file menu.

To save as a template and import a project template

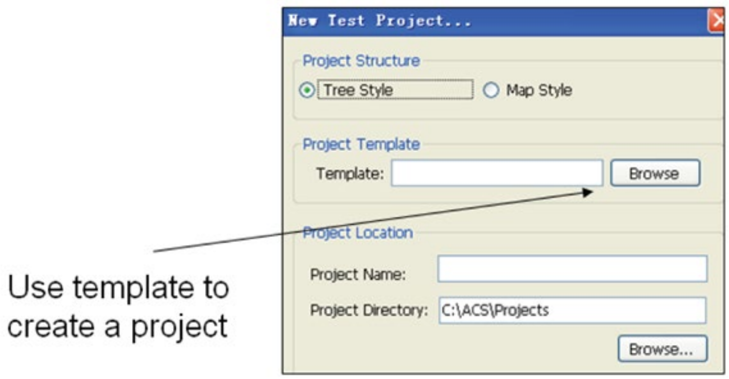
1. Open the existing project file or folder you want to save as a template.

Figure 92: Open existing project



2. From the **File** menu, select **Save As Template**.
3. Enter a name for the template and select a location to save the template.
4. Select **OK**.
5. Use the project template to create a new project.

Figure 93: Use the template to add a new project



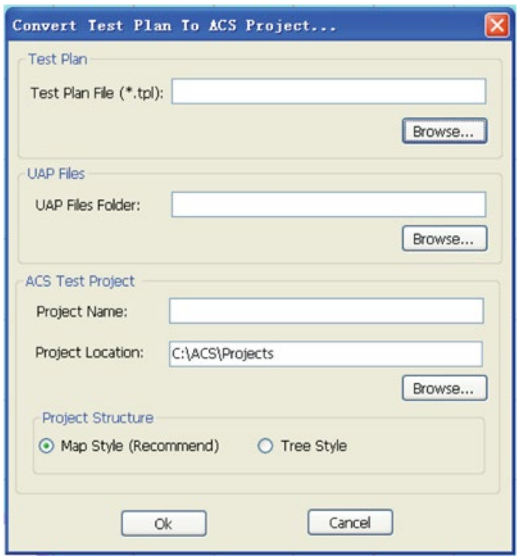
Go to Create a new project (on page 5-12) for more information on creating projects.

Import Test Plan

To import a test plan

1. From the **File** menu, choose **Import > Test Plan** (`*.csv`). The Convert Test Plan To ACS Project dialog opens.

Figure 94: Convert Test Plan to ACS Project



2. Select Browse the browser test plan you need (Test Plan, UAP Files, or ACS Test Project).
3. When you have the correct test plan, enter the **Project Name**.
4. Select **OK**.

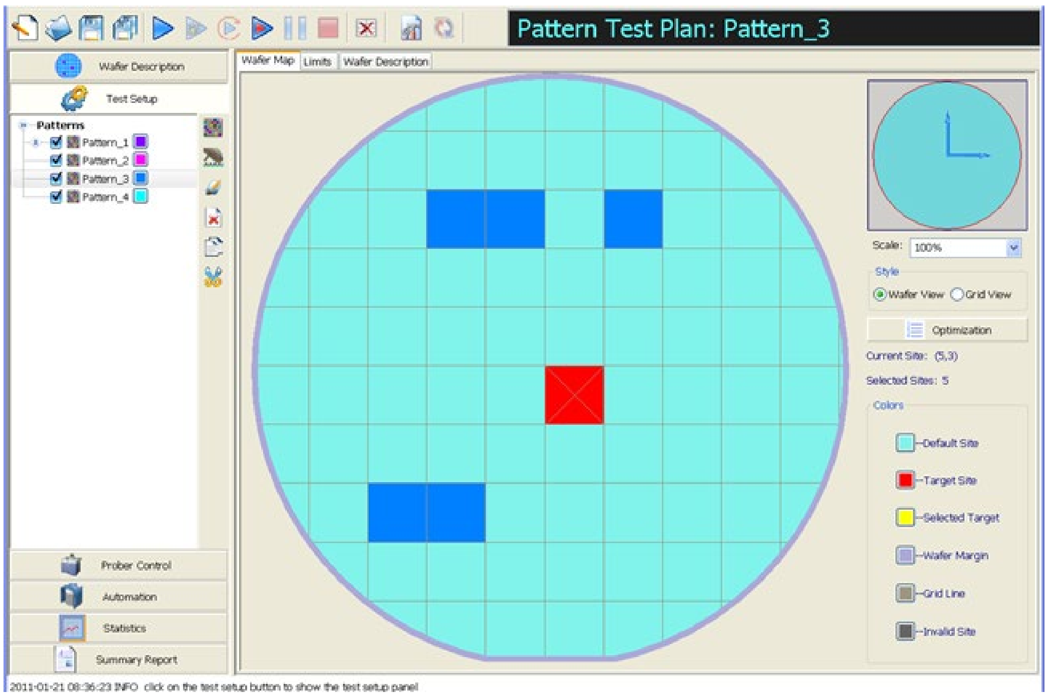
Building a test plan

After you create a new project, you can build and configure a test plan according to your preferences.

Add a new pattern

To add a new pattern, select the **New Pattern** icon on the toolbar for the current project. You can also right-click a pattern name and select **Insert Pattern**.

Figure 95: Pattern work area



There are three tabs in the pattern work area:

- Wafer Map
- Limits
- Wafer Description

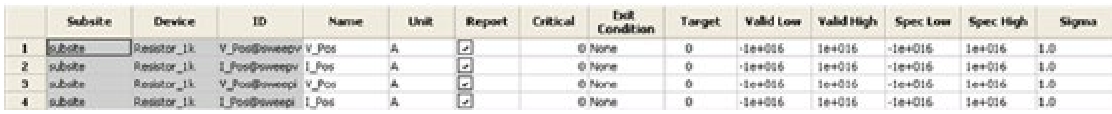
Wafer Map tab

All settings on the Wafer Map are the same as the Wafer Description panel. All patterns share the same wafer map.

Limits tab

The limits data associated with one parameter are arranged as one row. The fields of the Limits tab are shown in the following figure.

Figure 96: Limits tab



	Subsite	Device	ID	Name	Unit	Report	Critical	Exit Condition	Target	Valid Low	Valid High	Spec Low	Spec High	Sigma
1	subsite	Resistor_1k	V_Pos@sweepv	V_Pos	A	☑	0	None	0	-1e+016	1e+016	-1e+016	1e+016	1.0
2	subsite	Resistor_1k	I_Pos@sweepv	I_Pos	A	☑	0	None	0	-1e+016	1e+016	-1e+016	1e+016	1.0
3	subsite	Resistor_1k	V_Pos@sweepi	V_Pos	A	☑	0	None	0	-1e+016	1e+016	-1e+016	1e+016	1.0
4	subsite	Resistor_1k	I_Pos@sweepi	I_Pos	A	☑	0	None	0	-1e+016	1e+016	-1e+016	1e+016	1.0

The Limits tab fields are:

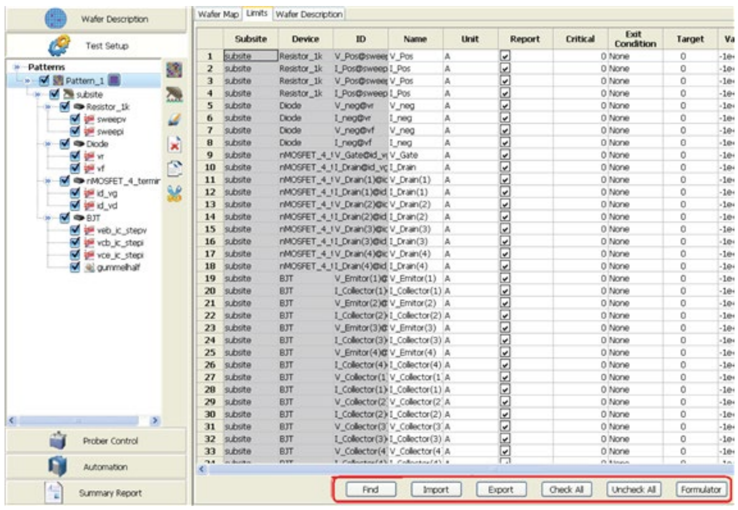
- **ID:** Uniquely identifies each test parameter. This field cannot be modified.
- **Name:** Can be modified.
- **Units:** Defines the units for the limits test results. This field may contain nonalphanumeric characters (including spaces) and does not need to be a C programming language name. The only constraint is that it must not include commas. The string may contain a maximum of 13 characters. Examples of entries for this field are volts or amps. The default value is `A`.
- **Report:** Determines whether the limit is included in the summary report. To include the report, select this option. To remove it from the report, clear it.
- **Critical:** Allows parameters to be tagged as not critical (0) or as critical with one of nine critical flags (1, 2, 3, 4, 5, 6, 7, 8, and 9).
- **Target:** Identifies the ideal result expected for the specified ID. The entry for this field is a numeric value.
- **Valid Low, Valid High:** This pair of limits can be used to check device failures or faults in the testing system. Test results outside this range indicates that the data being produced by the system is invalid, usually due to an equipment malfunction.
- **Spec Low, Spec High:** This pair of limits is used to check for manufacturing standards. Test results outside the specifications range indicates that the device being tested does not meet manufacturing specifications. When run in manual mode, if the test result is out of the specifications range, the background color of the data cell is defined in the Manual Run Fail Color Setting (see Fail data (on page 5-249) for more information). When run in automation mode, if the test result is out of the specifications range, the site in the wafer map will display a color indicating a failure (see Wafer map indicator (on page 7-33) for more detail).

- **Sigma:** Parameter used to show the statistical measurement of variability, showing how much variation exists from a statistical average.

Limits tab Formulator function

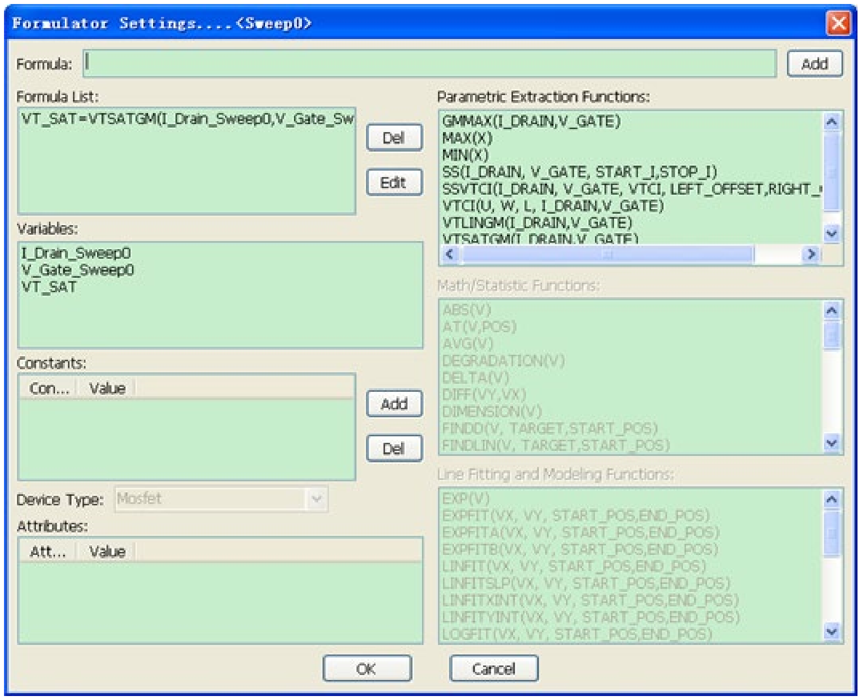
The Limits tab contains the Formulator. The Formulator is active when you select a pattern on the configuration navigator.

Figure 97: Limits tab functions



Select **Formulator** and the Formulator Setting displays.

Figure 98: Formulator Settings dialog



The Formula box is where you create new formulas or edit existing formulas.

NOTE

The Add formula function is part of the Formula.

When you select **Add** to the Formula, the following occurs:

- Starts a calculation execution of the formula
- Moves a formula to the Formula List box

The Formula List displays formulas that have been created. The **Del** function deletes a formula that is selected in the Formula List box. The Edit function moves a formula from the Formula List to the Formula box to edit.

The Constants indicate each Constant that you can insert by name. When you select the constant in the list it is added to the Formula equation, at the cursor position. Selecting **Add** opens a dialog that allows you to add a new constant to the list. After inserting a name and value, select **OK**. To delete a constant, select the constant then the **Del** function.

There are six Parameter Extraction Functions:

GMMAX	SS	SSVTCI
VTCI	VTLINGM	VTSATGM

There are 26 Math/Statistic Functions:

ABS	AT	AVG	DELTA	DIFF	DIMENSION
FINDD	FINDLIN	FINDU	FIRSTPOS	JOIN	LASTPOS
LN	LOG	LOG10	MAX	MAXPOS	MIN
MINPOS	POW	RES	RES_4WIRE	RES_4WIRE_AVG	SMOOTH
SUBARRAY1	SUBARRAY2				

There are 23 Line Fitting and Modeling Functions:

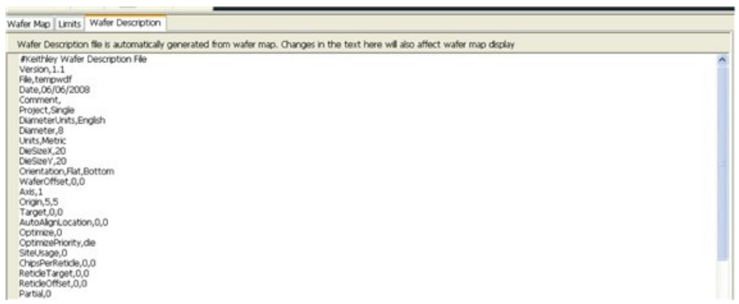
EXP	EXPFIT	EXFITA	EXPFITB	LINEFIT
LINFITSLP	LINEFITXINT	LINEFITYINT	LOGFIT	REGFIT
REGFITSLP	REGFITXINT	REGFITYINT	REGFIT_LGX_LGY	REGFIT_LGX_Y
REGFIT_X_LGY	RSQUAR	TANFIT	TANFITSLP	TANFITXINT
TANFITYINT	TARGERX	TARGETY		

For more information, refer to Formulator functions (on page 5-255).

Wafer Description tab

The Wafer Description tab displays the content of the file, as shown in the following figure. You can modify the description.

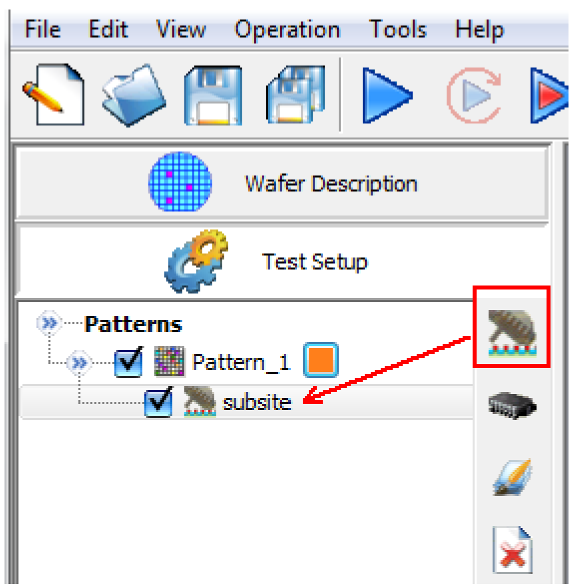
Figure 99: Status tab of pattern work area



Add a new subsite

Select the **New Subsite** icon on the toolbar to add a new subsite to the current project. The Subsite dialog opens, as shown in the following figure.

Figure 100: Add a New Subsite icon

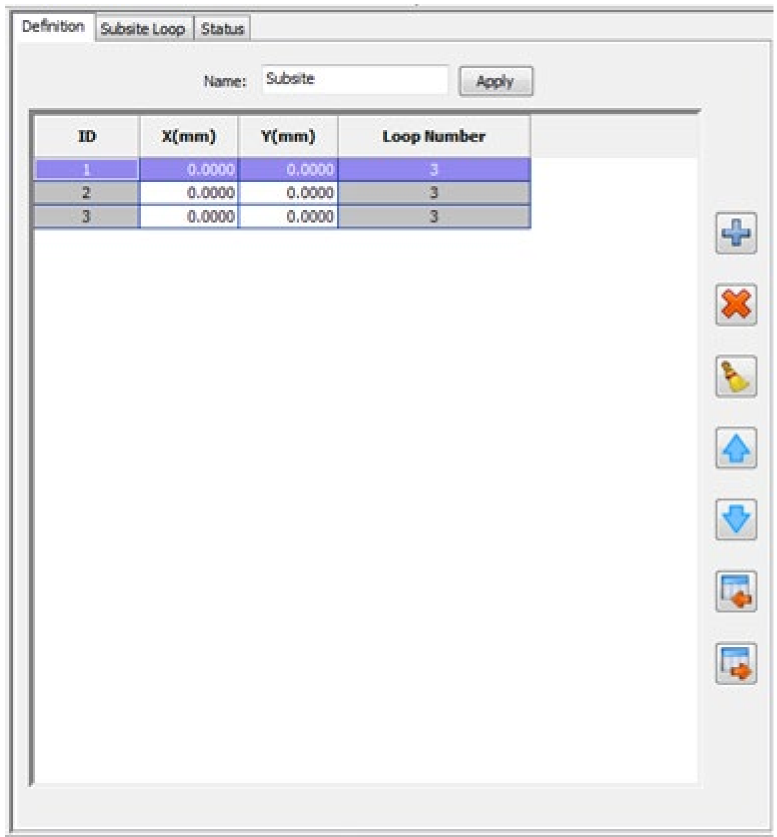


Definition tab

When you select the New Subsite icon, you will see three tabs in the dialog: Definition, Subsite Loop, and Status, as shown in the following figure.

- Add a name to the subsite in the Name field.
- In the X and Y axis cells, you set the subsite position relative to the site so that the probe is moved to the correct position. The axis direction is the same as in the Wafer Description. The X axis defines the horizontal position, and the Y axis defines the vertical position.
- The Loop Number displays the Number of cycles in Cycle Mode or total cycles in Stress and Measure Mode (this is set in the Subsite Loop tab).
- You can add multiple subsites and configure as appropriate. This function will save time in instances where you have the same device and test modules for a subsite.
- Select the **add** icon to add subsite parameters to the new subsite (in the Definition tab).
- Select the **delete** icon to delete selected subsites.
- Select the **broom** icon to clear all subsites and all data (note that one default row remains).
- Select the **up-arrow** icon to move selected subsites up.
- Select the **down-arrow** icon to move selected subsites down.
- Select the **import** icon to import a subsite list from a .csv file.
- Select the **export** icon to export a subsite list to a .csv file.

Figure 101: Subsite Definition tab

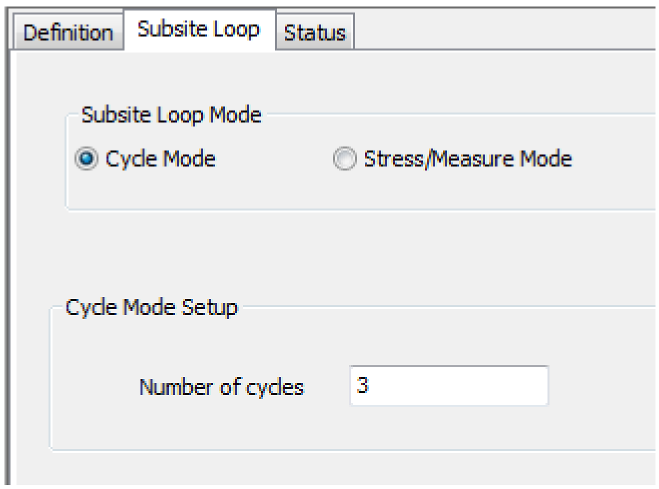


Subsite Loop tab

You can use the Subsite Loop to Cycle Mode or Stress and Measure Mode, as shown in the following figure.

Cycle Mode executes the subsite based on the configured Number of cycles.

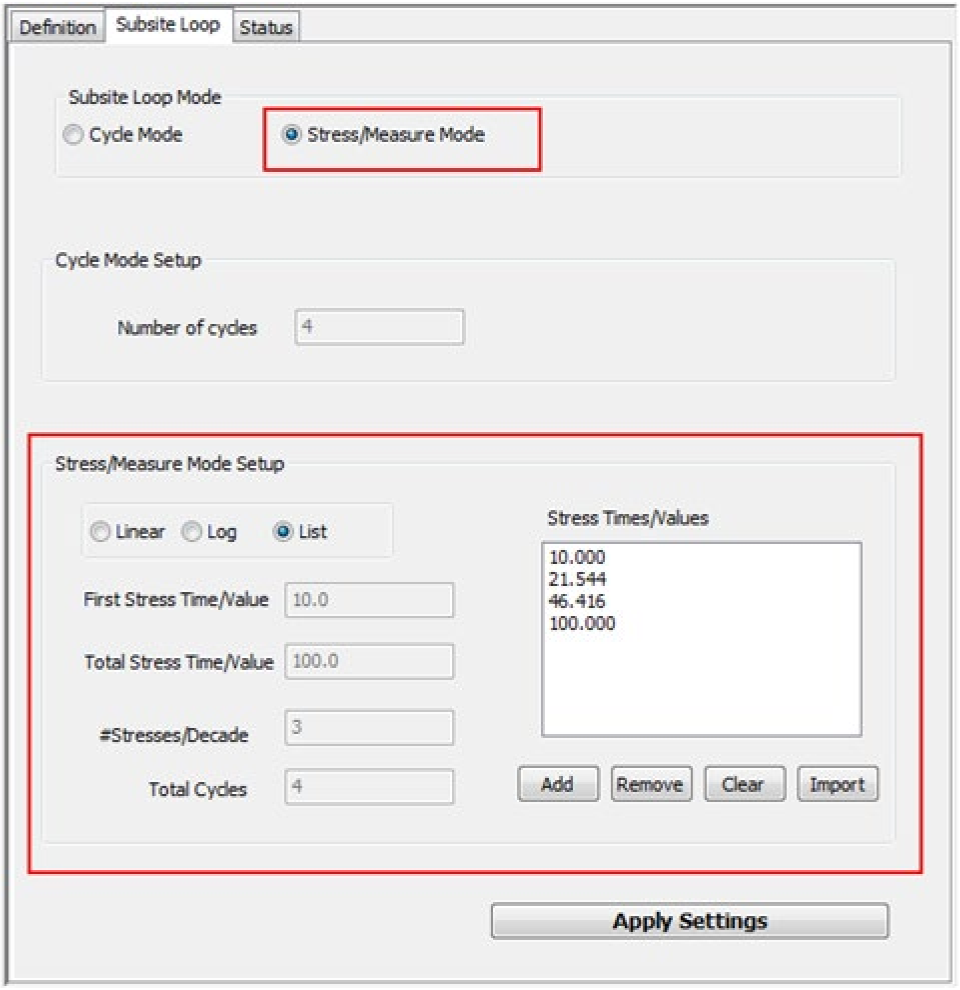
Figure 102: ACS Subsite Loop tab



Stress and Measure Mode controls the stress and measure loop reliability and sets Stress Times and Values. Note that time is not a limitation and that there are three Stress and Measure Mode Setup choices: Linear, Log, and List, as shown in the following figure.

- **Linear:** Ensures all Stress Times and Values are the same.
- **Log:** Ensures all Stress Times and Values increase logarithmically with each cycle.
- **List:** Allows you to specify the Stress Times and Values (note that the Add, Remove, Clear, and Import buttons are available).
- **First Stress Time and Value:** Enter the amount of time (seconds) to stress your devices during the first cycle.
- **Total Stress Time and Value:** Enter the total stress time.
- **Stresses and Decade:** Enter the number of cycles that you want performed.
- **Total Cycles:** Automatically calculated based on how many Stress Times and Values are entered when using the List function.

Figure 103: Stress Measure Mode setting



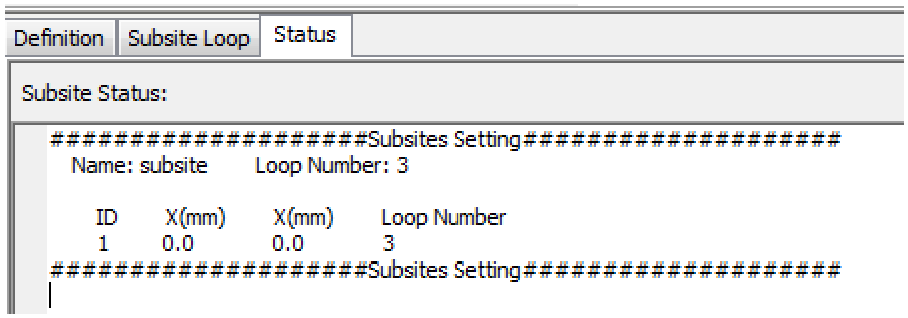
If you select List, the Stress Times and Values options are active:

- **Add:** Adds a stress value to the list.
- **Remove:** Removes the selected list items.
- **Clear:** Removes the entire stress list.
- **Import:** Imports a `.csv` stress list file.

Status tab

This tab shows the Subsite information.

Figure 104: Status tab

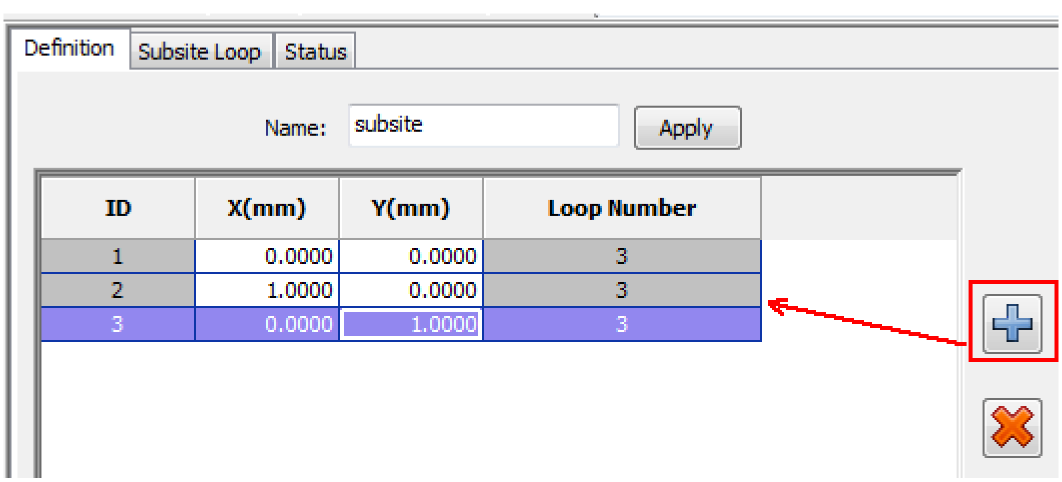


Run a subsite loop

To run a subsite loop:

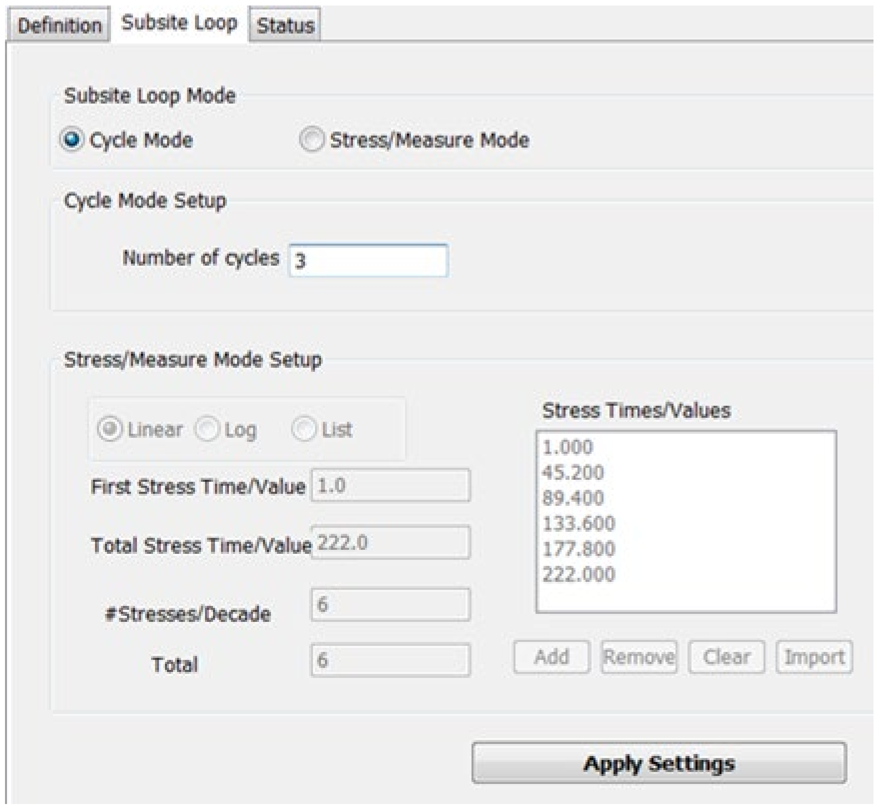
1. Select the **add** icon to add additional subsite parameters to the subsite (each row is appended to the subsite list).
2. Set the X axis (horizontal position) and Y axis (vertical position) for every subsite.

Figure 105: Add a new subsite



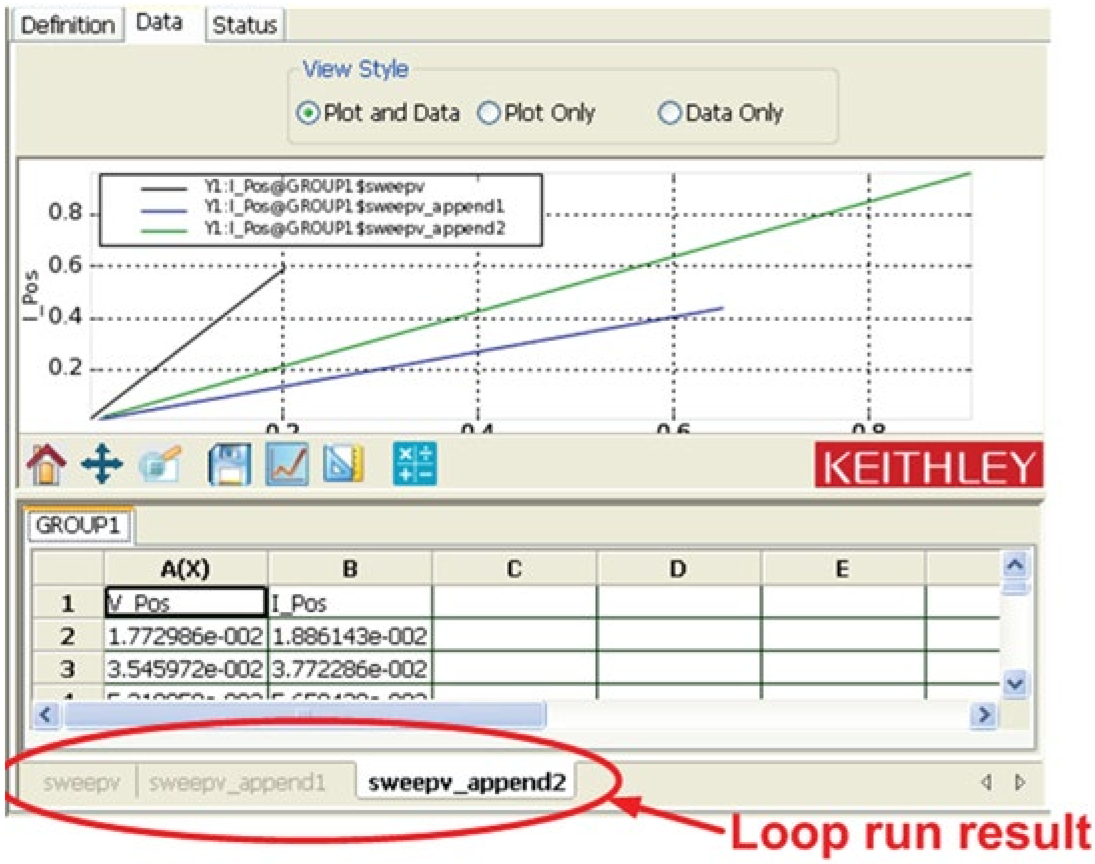
3. Set the Subsite Loop information and select **Apply Settings**.

Figure 106: Subsite Loop information



4. Select the **Run** icon (refer to Executing individual subsite plans (on page 5-231) for more information). The subsite modules cycle the number of cycles specified on the Subsite Loop tab. You will see groups of data in the Data tab.

Figure 107: Subsite Loop Data tab results



If you want to see real-time parametric curve information, select the Automation tab. The Automation tab allows you to see results in real time (refer to Real-Time Plot Settings (on page 7-17) for more information).

For more information about performing a stress test for device reliability analysis, refer to “Pulse stress-measure loop project” or “DC stress-measure loop project” in the *ACS Advanced Features Reference Manual*.

Add a new device

Each device is associated with a specific device work area. The device work area is the interface used to connect every device.

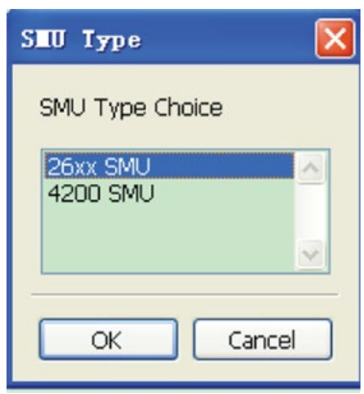
To add a new device for the current project:

1. Select **New Device** on the vertical toolbar.

NOTE

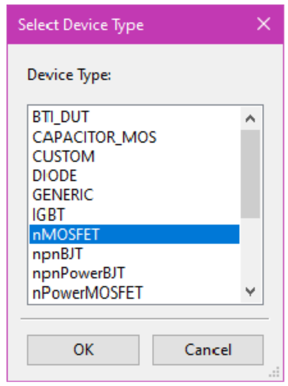
If the present hardware configuration includes the Model 4200A SMU and a Series 2600B SMU, the SMU type dialog opens. If you are including only one type of SMU, the dialog does not open and the SMU type is used by default.

Figure 108: SMU Type



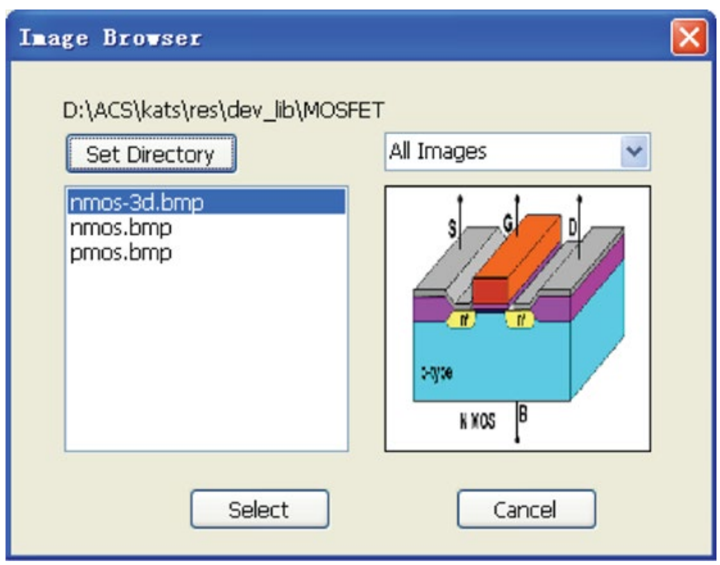
2. Select **4200A SMU** to create a new device to test or select **2600B SMU** to create a new SourceMeter instrument test.
3. Select a device, as shown in the following figure.

Figure 109: Select new device



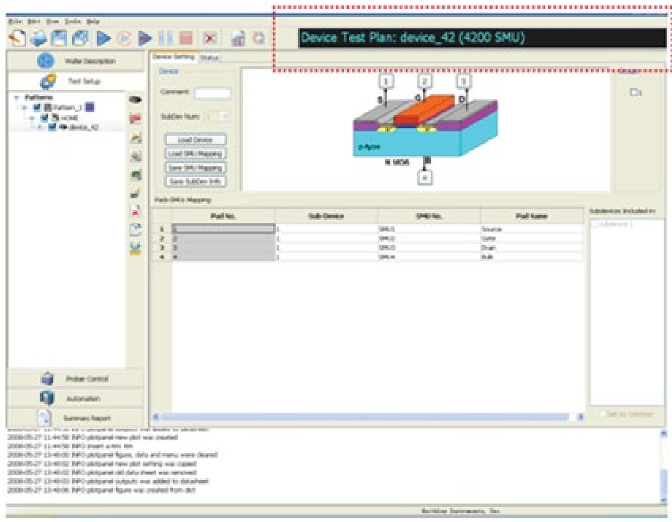
4. Select **OK**. The Image Browser dialog opens, as shown in the following figure.

Figure 110: Device images



5. Select the device image that has a terminal connecting to the SMU.
6. Choose **Select** and the new device is added to the configuration navigator. It is named based on the device type chosen and appended with either `_42` or `_26`, depending on the selected SMU type. The indicator area of the device work area is displayed, as shown in the following figure.

Figure 111: Device work area



Load and save device

The Device Setting tab opens by default. The column labels of the device Status tab are also displayed. In the Device Setting tab, you can assign the device setting based on the physical connection of the device. The device information can then be seen in the device Status tab.

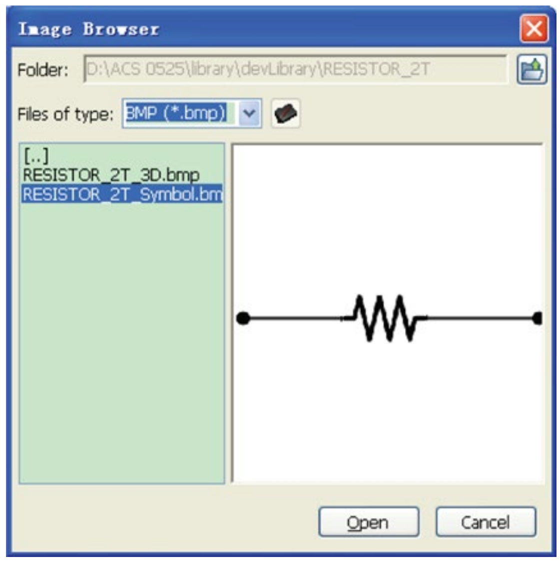
The Device Setting options are Load Device, Load SMU Mapping, Save SMU Mapping, and Save SubDev Info.

Figure 112: Functions on Device Setting tab

To change the device image or type for an existing device:

1. Select **Load Device**. The Select Device Type dialog opens.
2. Select a device type.
3. Select **OK** to load the device. The Image Browser dialog opens, as shown in the following figure.

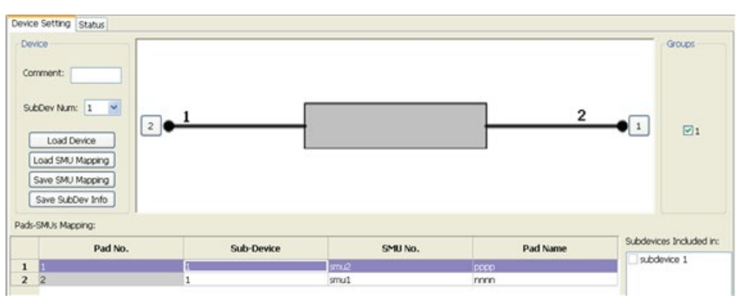
Figure 113: Image browser



4. Choose an image by selecting the device name, and then select **OK**.
5. Select **Open** to finish loading the device.

After loading the device, an image appears in the device work area, as shown in the following figure. With the device loaded, you can adjust the settings as needed.

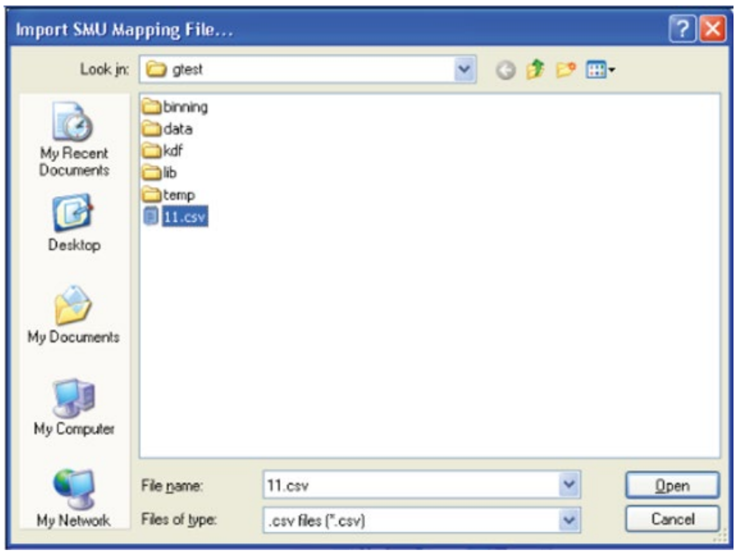
Figure 114: Device Setting work area



To load an existing SMU mapping file for specific devices:

1. Select **Load SMU Mapping**. The Import SMU Mapping File dialog opens.
2. Select an existing comma separated value (.csv) file, which has Pin No. and SMU No. matched respectively. For example, in the following figure, `11.csv` is an available SMU Mapping file.

Figure 115: Import SMU Mapping File



3. Select **Open**.

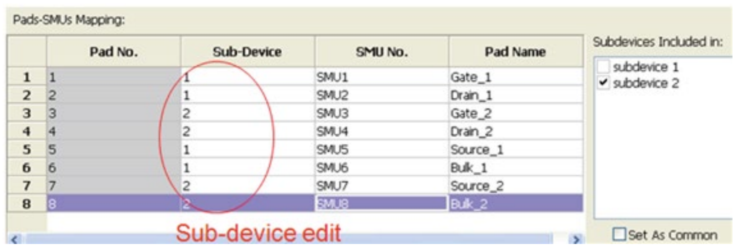
To save SMU mapping:

1. Select **Save SMU Mapping**. The Save SMU Mapping File dialog opens. Select the path for the .csv file to save the changes.
2. Select **Save**.

Save Sub-Device information

Under the device work area, in Pads-SMU Mapping item, there are four columns: Pad No., Sub-Device, SMU No., Pad Name, as shown in the following figure.

Figure 116: Load SMU mapping

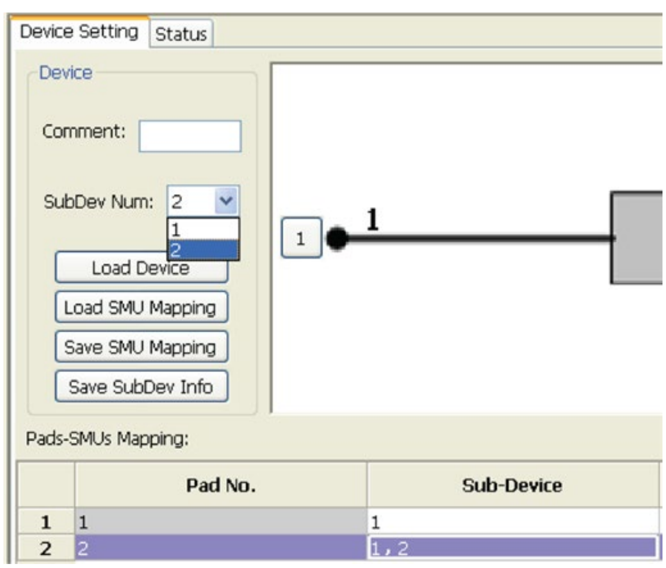


- **Pad No.**: The number of each pad that is preset and fixed.
- **Sub-Device**: indicates where each pad belongs.
- **SMU No.**: Listed according to the SMU Mapping, which is loaded in Load SMU Mapping step.
- **Pad Name**: Name of a pad.

To specify subdevice information:

1. Edit the subdevice field by entering numbers within the limits of SubDev No. For example, if the SubDev No. is 2, as shown in the following figure, the most numbers entered in Sub-Device editing box is 1, 2).

Figure 117: Device Setting; SubDev Num



2. To finish editing the Sub-Device, select Save SubDev Info.

Device Settings tab

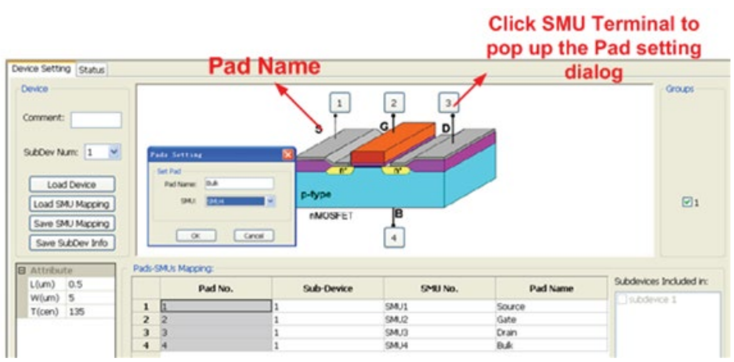
After inserting a new device into the project plan, it must be configured to match the physical connection. ACS supports single-device test, multi-device test, and multi-group test.

Setting a single device

To set up a single device:

1. From the Device Setting tab of the device work area, select the SMU terminal. The pad setting dialog opens, as shown in the following figure.

Figure 118: SMU Setting

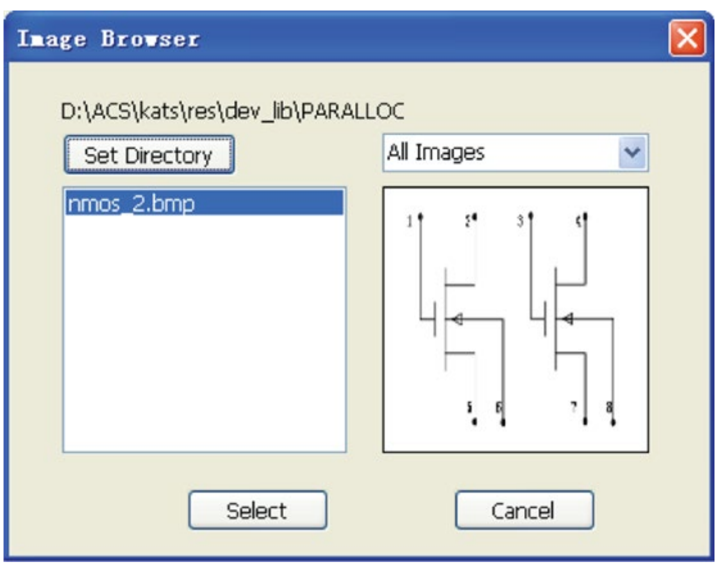


2. Select the connection in the instrument SMU list box that matches the physical connection of the device terminal. If the device pad is not connected (open circuit) or is connected to a particular SMU, you can select **NONE** (or choose not to set this SMU terminal). If the device pad is connected to Ground, select **GND**.
3. You can set the SMU number in the SMU No. column.
4. Select **OK**.
5. Repeat the above steps to set every SMU terminal. Edit the Pad Name (optional). You cannot name two pads the same name. If named the same, an Invalid Pad Name error is displayed.
6. Set the Attribute box. Input the device information in the Attribute box.
   - **L(um):** The length of device under test
   - **W(um):** The width of device under test
   - **T(cen):** The test temperature
7. Select **Save**.
8. If you are setting up a multi-group test, you do not need to configure every device one at a time. You can check the group number to be tested in the appropriate area.

Select a multi-device

Select the multi-device from the Image Browser dialog.

Figure 119: Select multidevice image



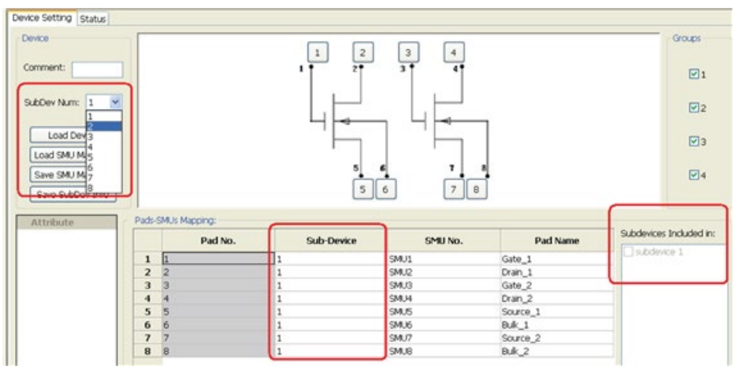
The settings for the multi-device are like the settings for the single device, except you choose the device number and matching subdevice number to the device terminals.

Define a multi-device

To define a multi-device:

1. Select **Load Device**.
2. Choose **Select Device Type**. The Device Setting dialog is displayed.
3. Define the type of subdevice. The subdevice is listed in the Subdevices Included In box, as shown in the following figure.

Figure 120: Setting two subdevices



4. Highlight any row in the Pad-SMUs Mapping area.
5. Set the SMU as indicated in steps 1 to 3. Check the subdevice in Subdevices Included In area to set the subdevice.
6. Match SMU, Pads, and Subdevice to complete the test device structure definition.

NOTE

If a pad is set as NONE or GND, it does not appear in the test set up page. You need to make the GND connection yourself. If a pad is set as common, it is shared by all subdevices.

Set up a multigroup test

When setting up a multi-group test, do not configure every device individually. Select the group number for each test in each area.

CAUTION

The software connections must accurately reflect the physical hardware connections at the time the test is executed. Incorrect terminal configurations result in anomalous test results and damage to the device. At test startup, you must check whether the SMU terminal connected to the device pad is matched correctly.

At the test startup, you must check that the SMU terminal connected to the device Pad is correctly matched. Perform the following steps (refer to Test and control connections (on page 11-1) for more detail).

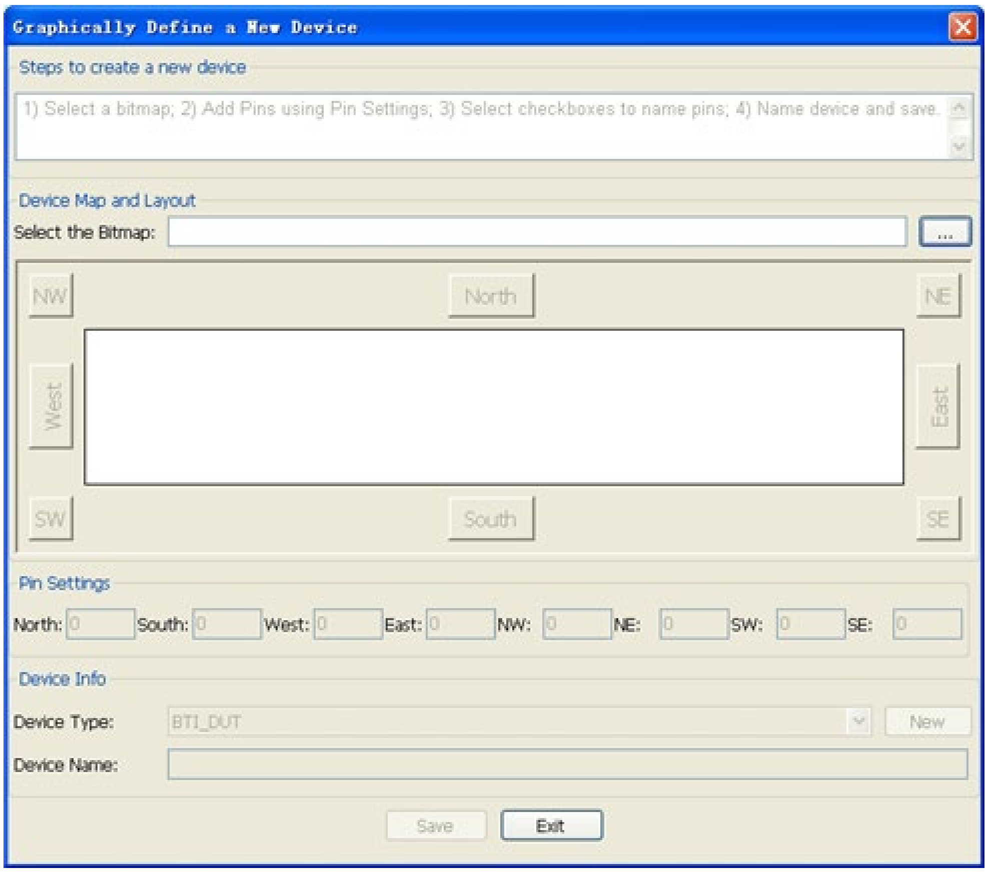
1. Check the physical connection of each test device terminal. Verify that it is not connected (open circuit), connected to the Ground Unit (KI_GND), or connected to a particular SMU.
2. Check the SMU No. list boxes to ensure that they are properly matched to the physical connection.
3. For each SMU in the Definition tab that is improperly matched, or not yet matched to a physical connection, assign, or reassign the terminal connection as indicated.

Graphically define a new device

To graphically define a new device:

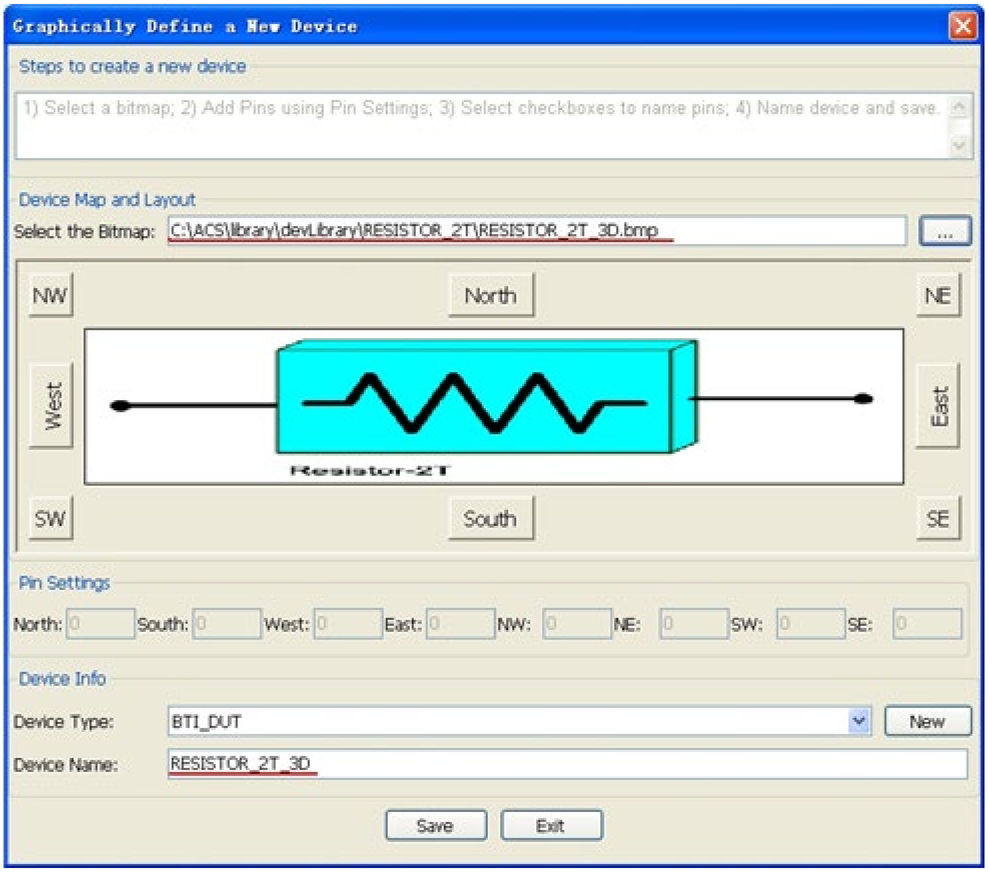
1. From the **Tools** menu, select **Graphically Define a New Device**. The Graphically Define a New Device dialog opens, as shown in the following figure.

Figure 121: Blank Graphically Define a New Device



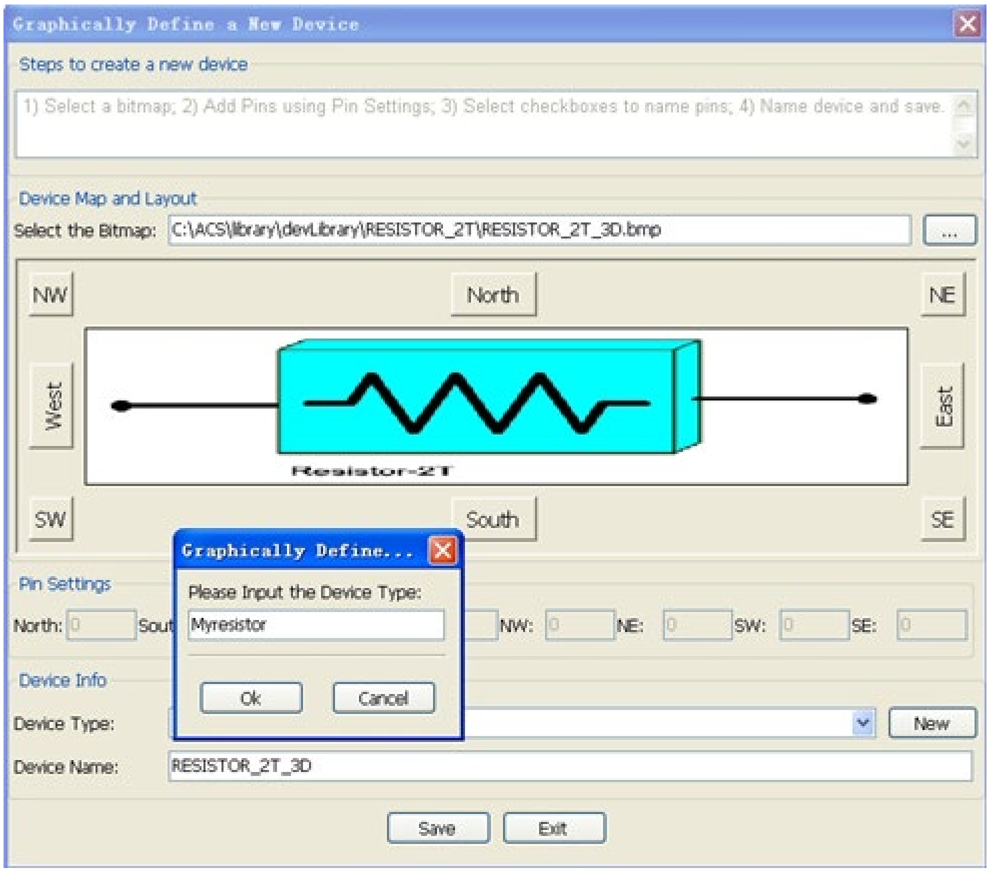
2. For Select the Bitmap, browse to choose the device bitmap. The name of the bitmap appears in the Device Name edit field as the default name, as shown in the following figure.

Figure 122: Select device bitmap



3. Select a **Device Type**. If needed, select **New** to input a new device. Edit the device name in the **Device Name** edit field, as shown in the following figure.

Figure 123: Graphically Define the device type



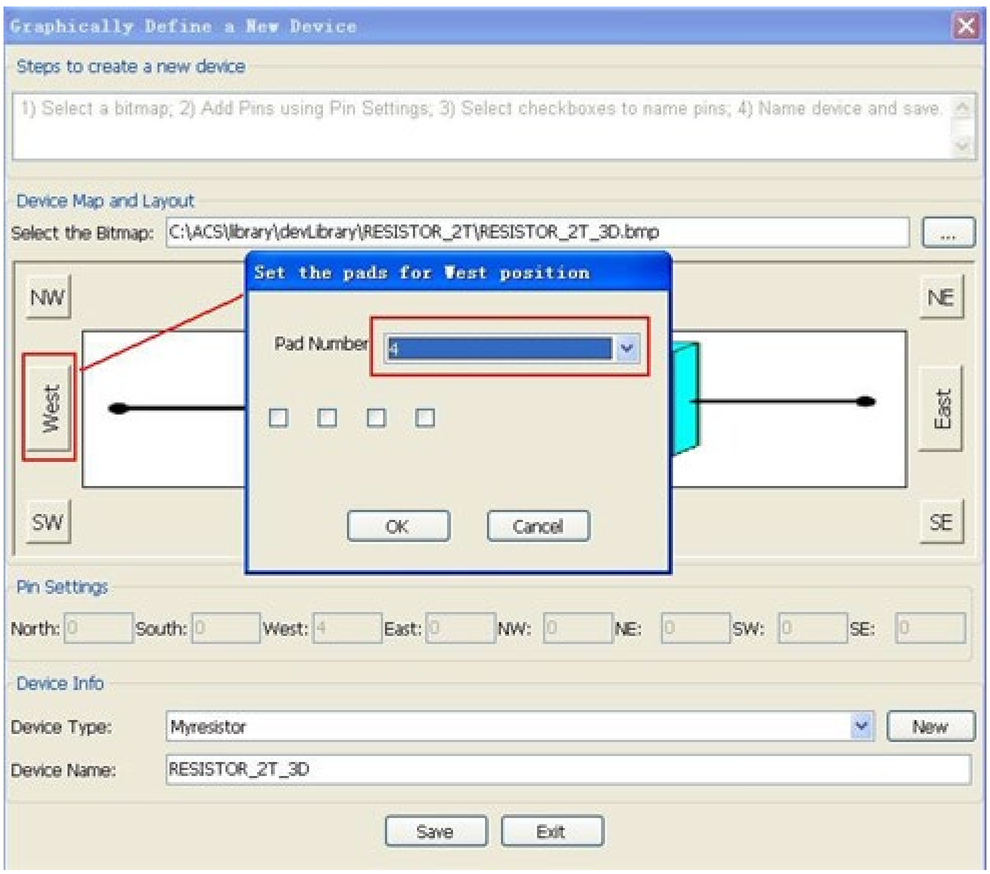
Enter the Pin Settings

The device bitmap is defined as North, South, West, East, NE (Northeast), SE (Southeast), NW (Northwest), and SW (Southwest).

Input the pad number in the North, South, West, and East according to the device bitmap. For example, in the following figure, the number 4 is entered in the West dialog.

When you select the Enter key or you move to another control, the pad boxes are added in the specified display areas corresponding to the counts entered in the Pin Settings.

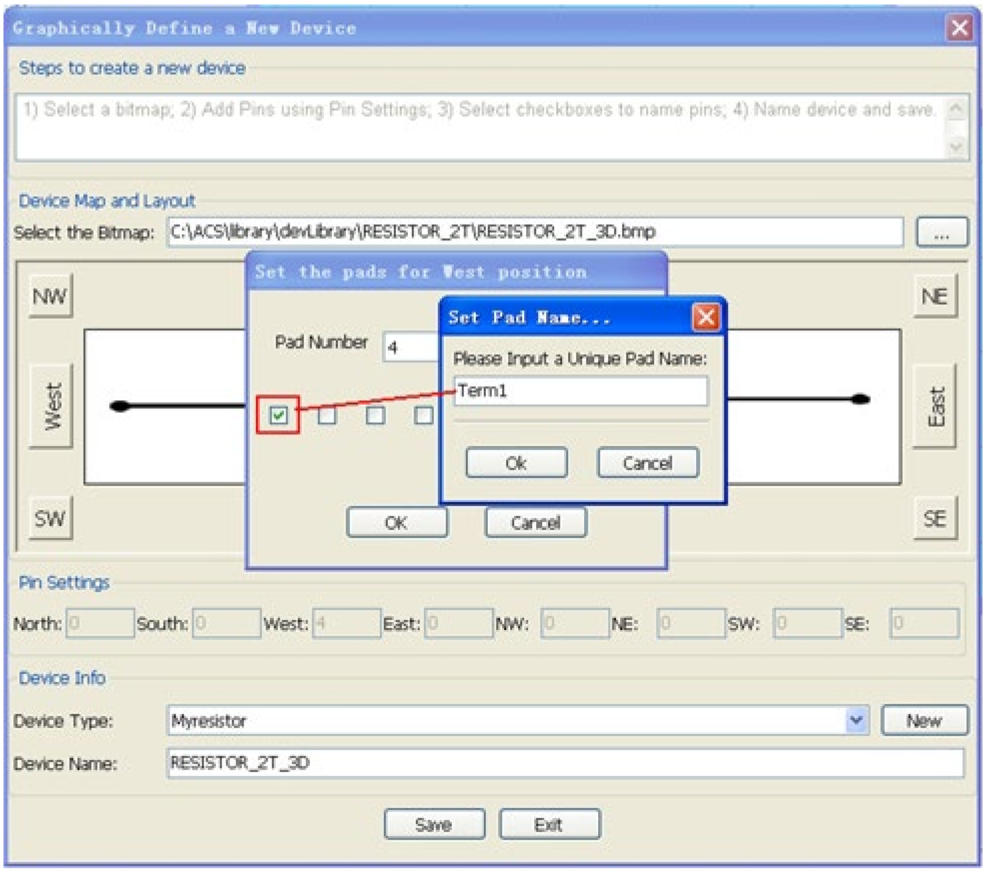
Figure 124: Enter Pin Settings



To set the Pad Name:

1. Select a box in the Device Map and Layout area. The Set Pad Name dialog opens, as shown in the following figure.

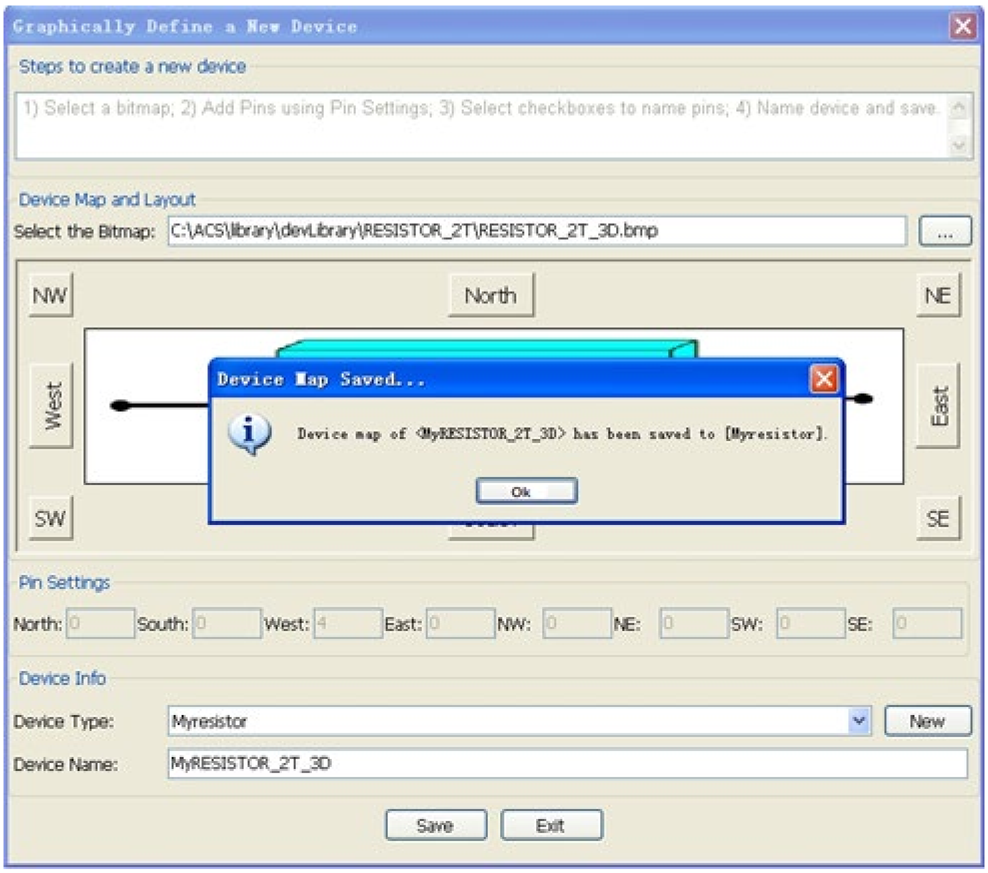
Figure 125: Set Pad Name



2. Enter a unique pad name into the Set Pad Name box.
3. Select **OK**.
4. Select **Save** to save this device to the loaded path.

The following figure shows the device MyRESISTOR_2T_3D is saved and ready for use by ACS.

Figure 126: Save the device to loaded path



Configure a graphical interactive test module (ITM)

An ITM allows you to define a test interactively using the user interface. ITMs are available for the Model 4200A SMUs or Series 2600B System SourceMeter instruments.

After inserting an ITM into the configuration navigator, it must be configured to meet your test requirements.

ITM

NOTE

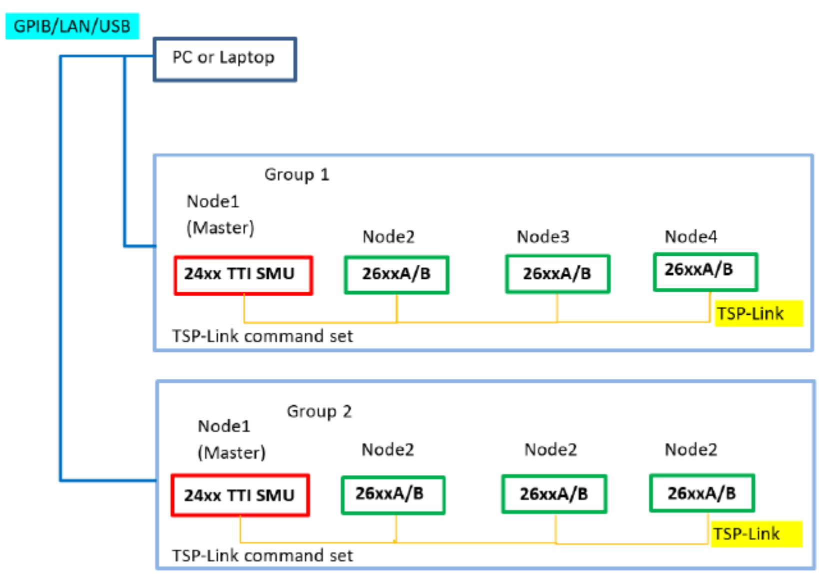
If you are using a Series 2400 SourceMeter, you need to set the TSP mode to ITM.

The ITM supports testing on the Series 2600B trigger model and it supports both DC and pulse measurement. It also supports the Series 2400 Graphical Interface SMUs in trigger mode and high-power SMUs, such as Models 265xA, 2606B, and 26x4b in a safe configuration.

NOTE

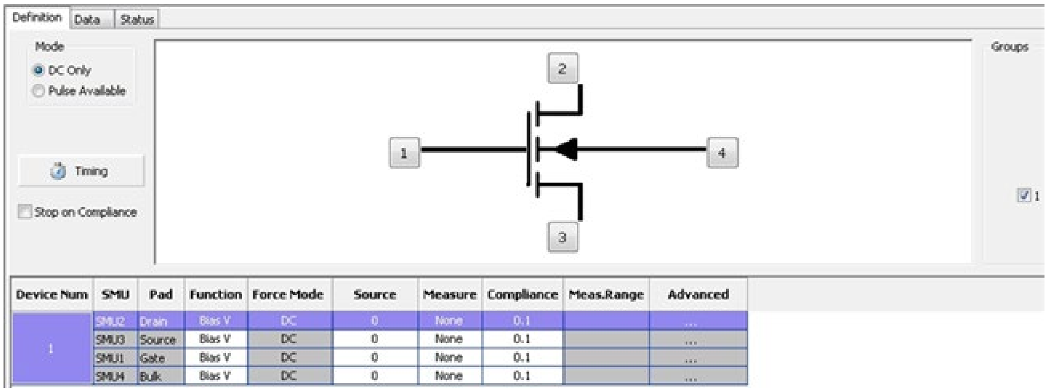
If you are using a mix of Series 2600B and 2400 Graphical Interface SMUs (Models 2450, 2460, 2461, 2470), the master node must be on the 2400 SMU. See the following figure for an example.

Figure 127: Series 2600B and 2400 Graphical Interface SMUs



Each ITM is associated with a specific ITM work area. An ITM work area displays the ITM's definition (configuration), test data, and status information, as shown in the following figure.

Figure 128: Definition tab of the Model 26xxB and 24xx SMU ITM



The Definition tab contains the following elements.

Mode:

- DC Only: Select if you want all SMUs to only operate in DC Mode.
- Pulse Available: Select if you want SMUs to operate in DC or Pulse Mode. Pulse timings for each SMU may be assigned to one of two Pulse Timing configurations. Because accurate timing is required for pulsing, some features, such as autoranging, are disabled when you select Pulse Available.

Timing: Select this button to open the ITM timing dialog, where you can set detailed timing settings for pulse and synchronization between SMUs.

Stop on Compliance: Select if you want to abort the test and turn off the SMU output when the test reaches compliance.

Device schematic: Shows the device image and pad settings according to the device level settings in the configuration navigator.

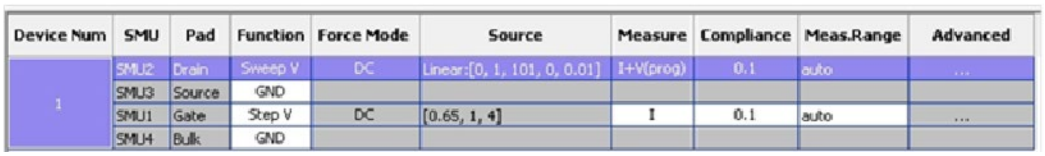
Groups: Select the group or groups where you want to run the test. The group is determined by the GPIB or ethernet address. Make sure that all the groups you select have the same hardware configurations as your ITM.

SMU Parameters table: Set the parameters for testing (see Model 26xxB and 24xx SMU parameters (on page 5-61) for more information).

Model 26xxB and 24xx SMU parameters

The following figure displays the items to set up testing for forcing and measuring functions for each SMU.

Figure 129: SMU parameters table



Device Num	SMU	Pad	Function	Force Mode	Source	Measure	Compliance	Meas.Range	Advanced
1	SMU2	Drain	Sweep V	DC	Linear:[0, 1, 101, 0, 0.01]	I+V(prog)	0.1	auto	...
	SMU3	Source	GND						
	SMU1	Gate	Step V	DC	[0.65, 1, 4]	I	0.1	auto	...
	SMU4	Bulk	GND						

Device Num: Sub-device number defined in the device work area.

SMU: SMU number defined in the device work area.

Pad: Pad name defined in the device work area.

Function: Force function type.

Force Mode: DC for DC Only. DC, pulse timing 1, or pulse timing 2 for pulse available.

Source: The source value of SMU.

Measure: Selects the measurement function option:

- I: Actual measured current values.
- V: Actual measured voltage values.
- V (prog): Reports the programmed voltage. No measurement is made.
- I (prog): Reports the programmed current. No measurement is made.
- I+V: Measures both current and voltage.
- I+V (prog): Measures current and returns voltage (prog).
- V+I (prog): Measures voltage and returns current (prog).

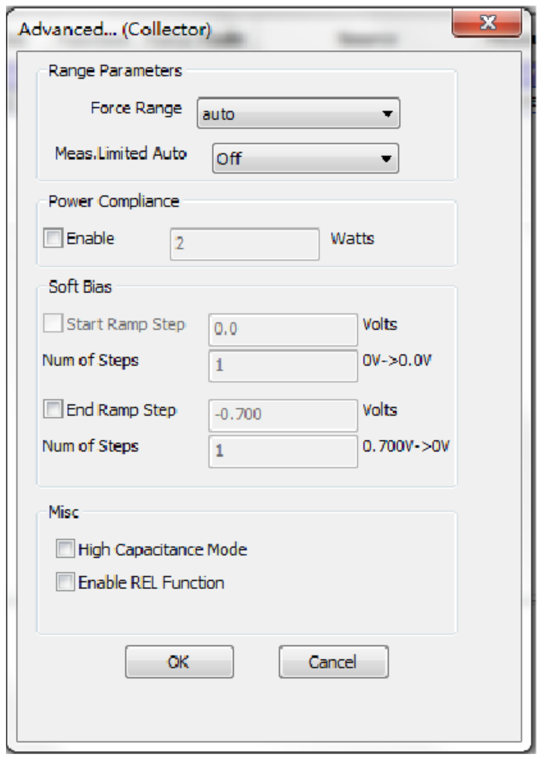
Compliance: Sets current limit for voltage source or voltage limit for current source.

Meas. Range: Specifies the SMU range used. The current measure range is listed for voltage forcing functions, and the voltage measure range is present for current forcing functions only. If current is selected in the Measure cell, ACS enforces the Force range first, and ignores the measure range. You can specify the range used by the SMU using the following options:

- Auto commands the SMU to automatically optimize the measurement range. This option provides the best resolution.
- Numerical ranges allow you to manually select a fixed measurement range to suit your needs.

Advanced: Double-click the cell to display a dialog for advanced settings for this SMU.

Figure 130: SMU Advanced settings dialog



Range Parameters:

- **Force Range**: Specifies the SMU force range used to force voltage or current, including auto, Best Fixed, and numerical range options.
- **Meas. Limits Auto**: When measuring current with Meas.Range set to auto, the application enables this option. This parameter is used with autoranging to put a lower bound on the range used. Since lower ranges generally require greater settling times, setting the lowest range limit might make measurements require less settling time. This option is a compromise between the full auto option and a fixed range option. You can specify the minimum range that the SMU uses when automatically optimizing the current measurements. This option saves time when maximum resolution at minimum currents is not needed.

Power Compliance: Set the power compliance value (watts) for a 2600B SMU. The firmware version of the 2600B SMU must be at least higher than 2.2.1.

Soft Bias: Apply Soft Ramp at the beginning of the bias or first sweep or list point and at the end of the bias or last sweep or list point. Typical use cases are for high power MOSFET measurements. For more information about using Soft Ramp, refer to the following topics:

- SMU forcing functions (on page 5-64)
- General-purpose PTM configuration (on page 5-177)

Misc:

- **High Capacitance Mode**: Sets the SMU to high capacitance mode. This is only available for the Series 2600B SMU and Series 2400 SMU.

NOTE

The enable REL function is available when ITM REL Enable (DC compensation project) is selected in the toolbar (Tools > Preference > Instruments).

- **Enable REL Function**: When Enable REL (relative) is selected, the ITM will enable the REL function for the SMU and set the global data offset value to the relative offset value. When the REL function is disabled, the ITM will disable the relative offset. Note that an SMU can execute DC compensation when the fixture opens. The SMU forces 0 volts, measures the offset current at the user specified range, and ACS software keeps the value as global data. For more information, refer to DC compensation.
- **On Sequence for (pad name)**: The sequence number for turning on the SMU pad in the test. If set to 1, the SMU pad will be the first to turn on.
- **Sequence Delay**: The delay before turning on the SMU pad.

NOTE

The On Sequence for (pad name) and the Sequence Delay is available when the ITM Sequence is selected from the toolbar (**Tools > Preference > Instruments**).

SMU forcing functions

The SMU forcing functions are selected in the SMU parameters table.

The functions are bias V, bias I, sweep V, sweep I, list V, list I, step V, step I, V meter, GND, and open.

In DC mode you can select sweep2 V and sweep2 I.

Bias voltage

Bias V forcing maintains a defined constant voltage, subject to the specified compliance value of the connected SMU. Select Bias V to set a valid value in the source cell.

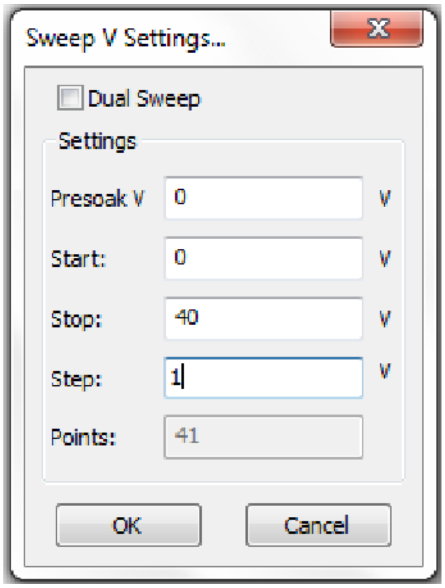
Bias current

Bias I forcing maintains a defined constant current, subject to the specified compliance value of the connected SMU. Select I to set a valid value in the source cell.

Sweep voltage, Sweep current

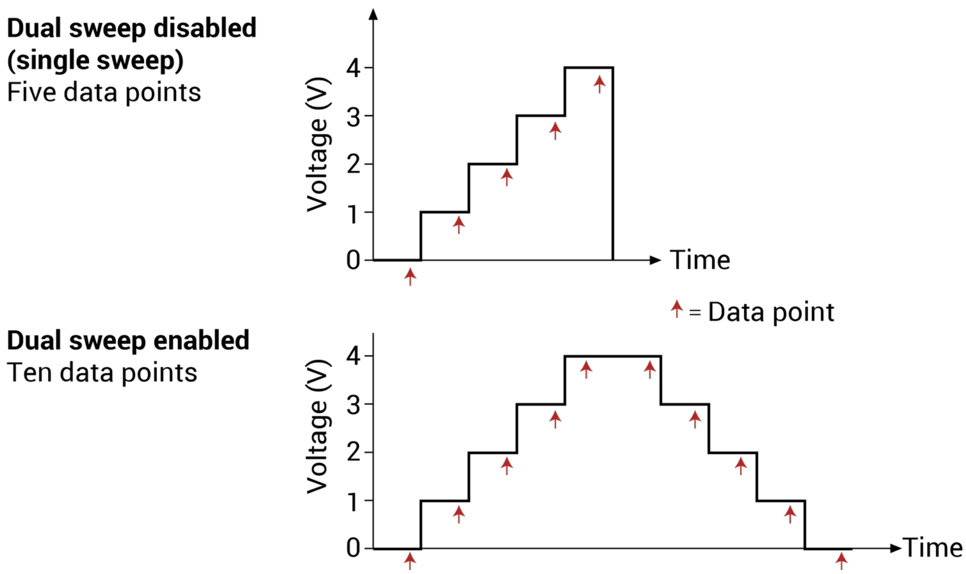
A sweep increments or decrements the current or voltage in a series of steps at a speed that is determined by the speed and timing settings. If you select **Sweep V** or **Sweep I**, you can set the sweep settings by double-clicking the source cell and making the necessary changes in the dialog.

Figure 131: Source settings for SMU sweep functions



With dual sweep enabled, the SMU will perform two sweeps. The first sweep steps from the start level to the stop level. The behavior of the second sweep is determined by the forward or reverse selection.

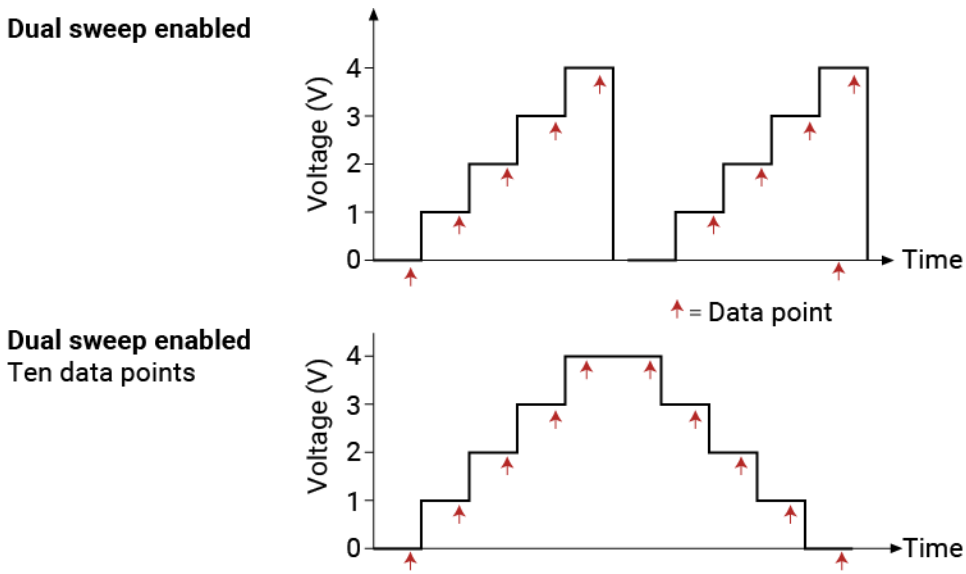
Figure 132: Single and dual sweep examples (linear voltage sweep; 0 V to 4 V in 1 V steps)



Forward: First sweep steps from the start level to the stop level. The SMU continues with the second sweep and proceeds in the same manner as the first sweep.

Reverse: First sweep steps from the start level to the stop level. The SMU continues with the second sweep and steps from the stop level back to the start level.

Figure 133: Dual sweep mode - forward and reverse



Mode:

- Linear: Sweep mode based on the linear scale.
- Log: Sweep mode based on the logarithm scale.

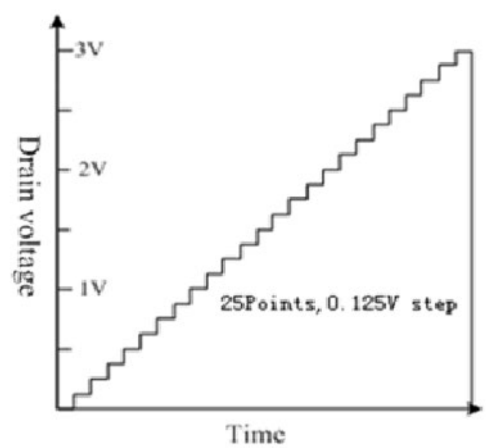
Settings:

- Start: Specifies a starting point for a sweep for any valid SMU voltage or current depending on the source function.
- Stop: Specifies a stopping point for a sweep for any valid SMU voltage or current depending on the source function.
- Step: Specifies the incremental value between every two points. You can edit the step and it will automatically calculate depending on the Start, Stop, and Points values.

Points: When a sweep is executed, it shows the number of data points.

Base: When pulse available is selected, this function sets the pulse base level.

Figure 134: Example sweep drain voltage on a MOSFET



Step voltage, Step current

Step X always works with a sweep X function. For each step X, a complete sweep X is performed. Measurement data is arranged by step number in the data tab.

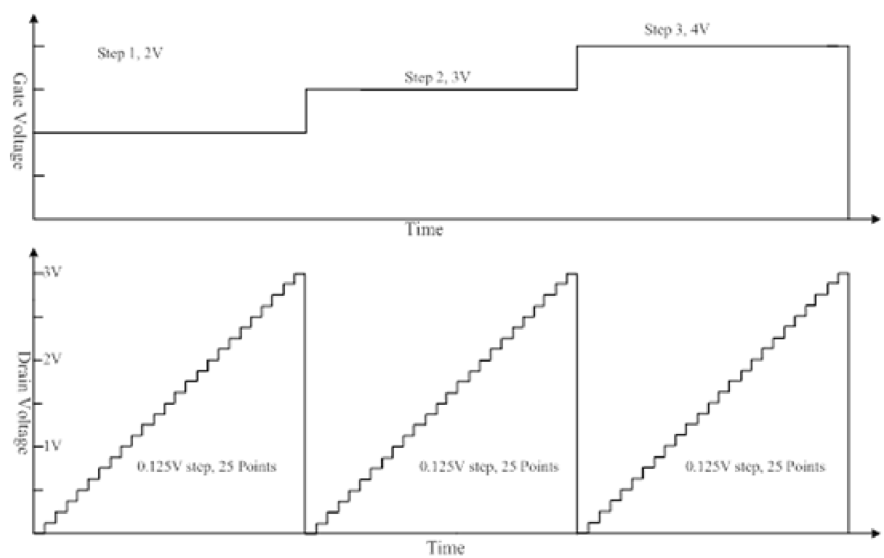
A typical voltage step forcing function example is shown in the following figure. This figure illustrates how one definition tab (for the Vds-Id ITM) specifies both step and sweep forcing functions.

Figure 135: Typical step forcing parameters

Device Num	SMU	Pad	Function	Force Range	Source	Measure	Compliance	Meas Range	Limits Auto
1	SMU1	Source	Bias V	auto	0.000	None	0.100		
	SMU2	Gate	Step V	auto	[2, 4, 3, 0]	V(prog)	0.100		
	SMU3	Drain	Sweep V	auto	Linear:[0, 3, 25, 0]	I+V(prog)	0.100	auto	
	SMU4	Bulk	Bias V	auto	0.000	None	0.100		

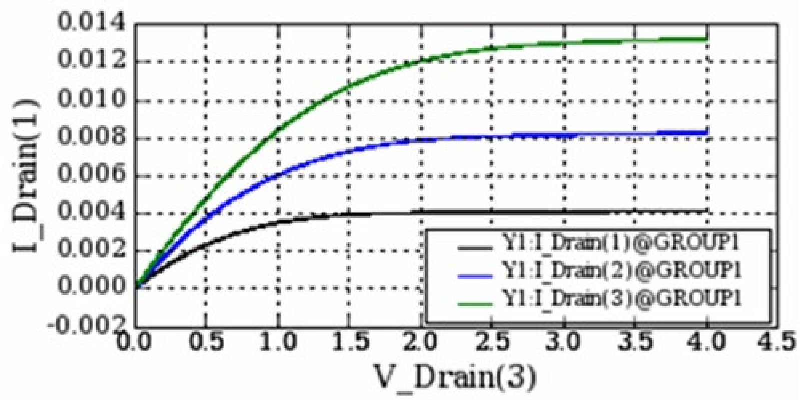
The following figure graphically illustrates the combined step and sweep forcing functions specified in the previous figure.

Figure 136: Example MOSFET step gate and sweep drain voltage



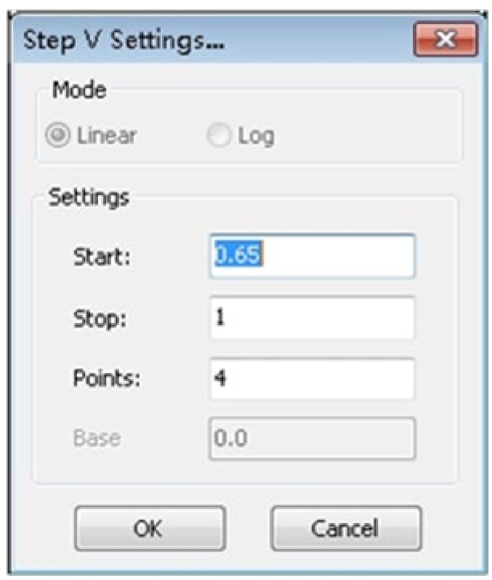
The following figure is a graph with a data plot from a typical forcing parameters figure.

Figure 137: Vds-Id graph



Double-click the source cell of the SMU that contains step V or step I, to display the step settings.

Figure 138: Source settings for SMU step functions



Settings:

- Start: Specifies the initial SMU output level.
- Stop: Specifies the final SMU output level.
- Points: Specifies the number of data points that are input when a step is executed. It also specifies the value of the voltage increments of every step value. At least two points must be entered in the Data Points cell. For example: If Start = 0 V, Stop = 4 V, and Points = 5, step value = (4 V - 0 V) divided by (5-1) = 1 V (five values are forced: 0 V, 1 V, 2 V, 3 V, 4 V).
- Base: Set the Pulse base level when Pulse Available is selected.

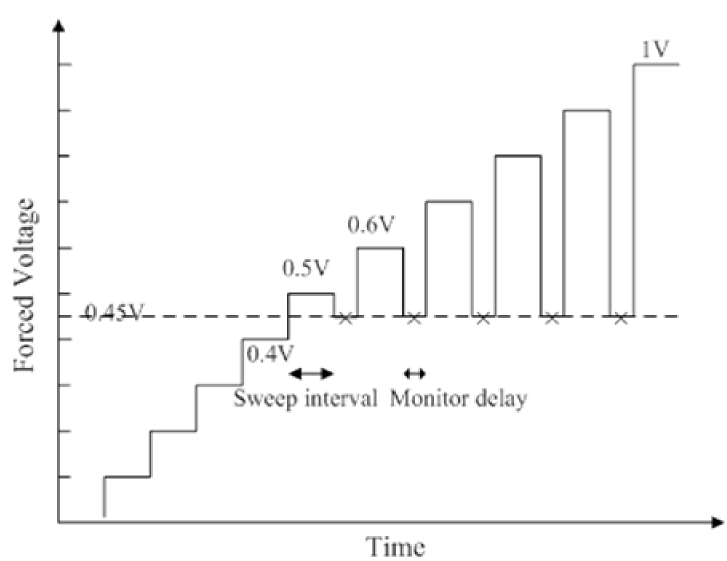
Sweep2 voltage, Sweep2 current

Sweep2 is a specific sweep test when used in DC Only mode. When the forced value is greater than a set value, a monitor test adds to each sweep step.

The test can generate two test results: Measured results and monitored results.

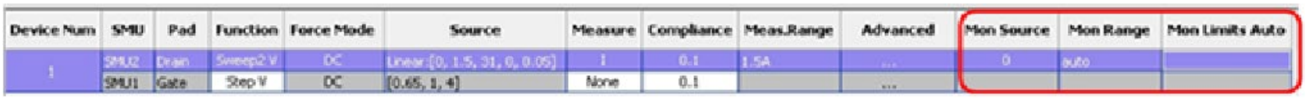
Specifically, when each sweep interval increases, it equals the value of the sweep delay and monitor delay. In the following figure; the sweep 2 V waveform has a 0.45 V as the monitor source voltage.

Figure 139: Sweep 2 V waveform



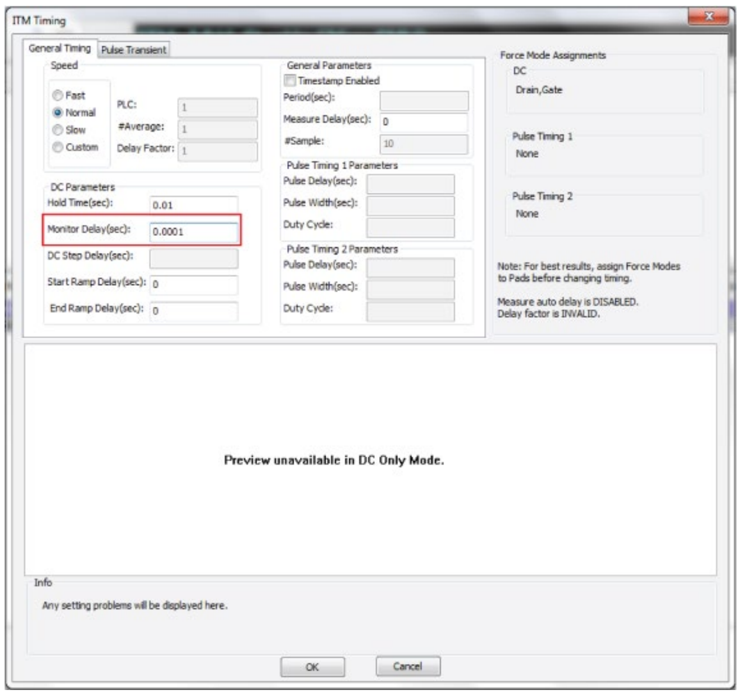
Select the Function cell and choose **Sweep2 V** or **Sweep2 I**. The monitor items are appended to the Force and Meas cell, as shown in the following figure.

Figure 140: The SMU monitor items



Also, the Monitor Delay (sec) item is enabled, as shown in the following figure.

Figure 141: Enable monitor delay

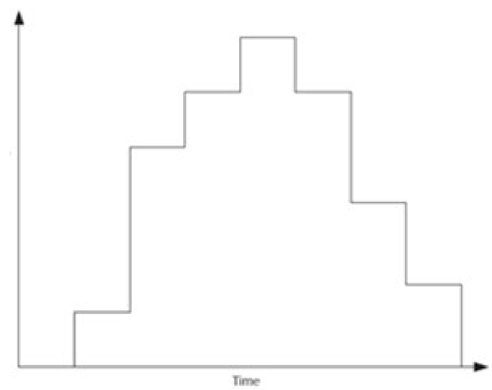


List voltage, List current

List V and List I define a voltage or current list according to the requirements of the test. The SMU forces the input values one by one like a sweep.

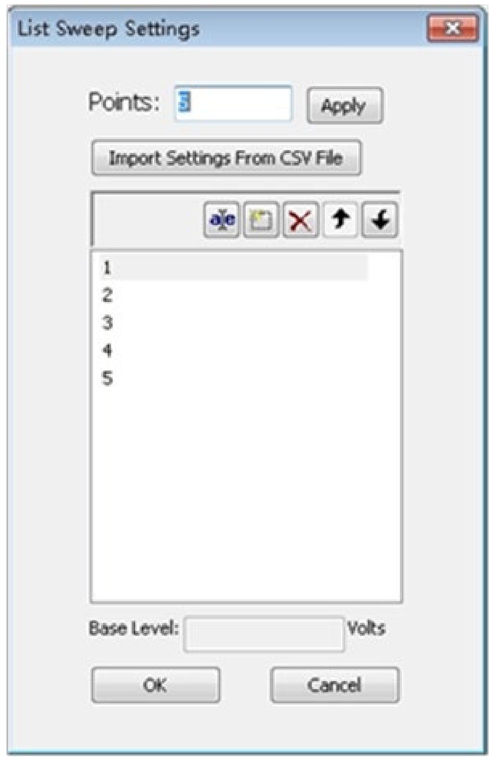
List sweeps allow you to make measurements only at selected forced voltages and currents. For example, they allow you to skip measurement points or to synthesize a custom sweep that is based on a mathematical equation.

Figure 142: List sweep general illustration



Double-click the source cell of the SMU to open the **List Sweep Settings** dialog.

Figure 143: List Sweep Settings dialog



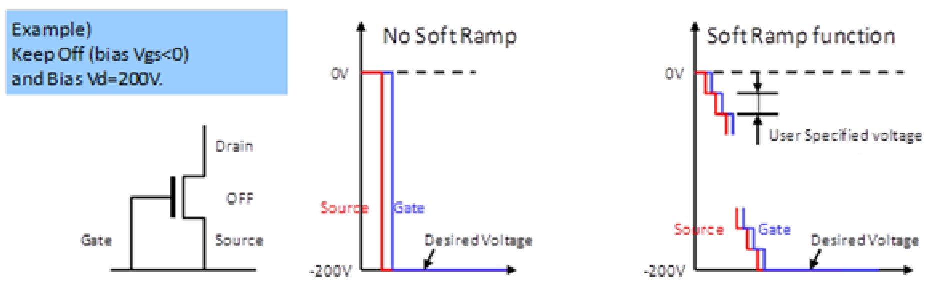
- **Points**: The number of points for the forcing values.
- **Import Settings From CSV File:** You can import (and then save) a `.csv` file by selecting Import Settings From CSV File (see previous figure).
- **Edit item:** The default number can be changed to a different value.
- **New item:** A new item is added when you select this icon.
- **Delete item:** A selected item is deleted when you select this icon.
- **Move up:** A selected item moves up when you select this icon.
- **Move down:** A selected item moves down when you select this icon.
- **Base Level:** When Pulse Available is selected, displays the Pulse base level.

Soft Ramp Bias

DC Only mode enables you to apply incremental test levels at the beginning of the bias, first sweep, or list point tests, and at the end of the bias, last sweep, or list point tests. This function allows the forced voltage or current to achieve the set value after a certain number of steps.

The soft ramp is from 0 to the set value (bias value, first and last sweep, or list value). If the bias, first or last sweep, or list point is 0 and no soft ramp is necessary. In the following figure, the No Soft Ramp gate breakdown occurs because of the 200 V stress applied. However, with the Soft Ramp function, you can keep the voltage differences at smaller levels until the set voltage is applied, as shown in the following figure.

Figure 144: Comparison of Soft Ramp and without Soft Ramp



Soft Ramp is enabled in the Advanced settings in the SMU parameters table. The soft ramp timing is set in the ITM General Timing tab (see the following figures).

Figure 145: Advanced Soft Bias settings

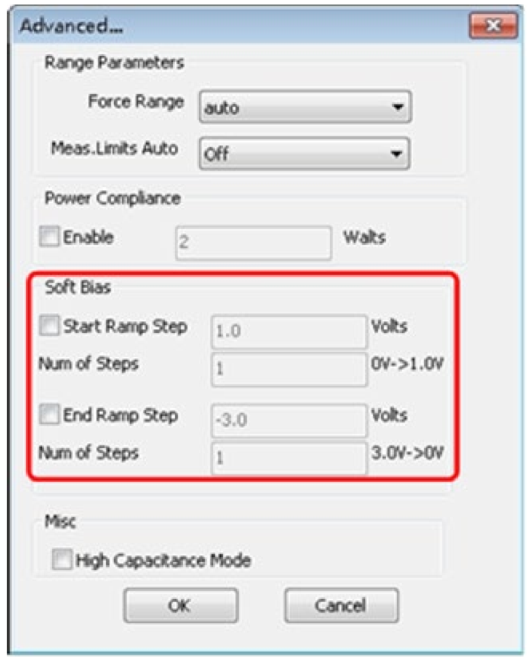
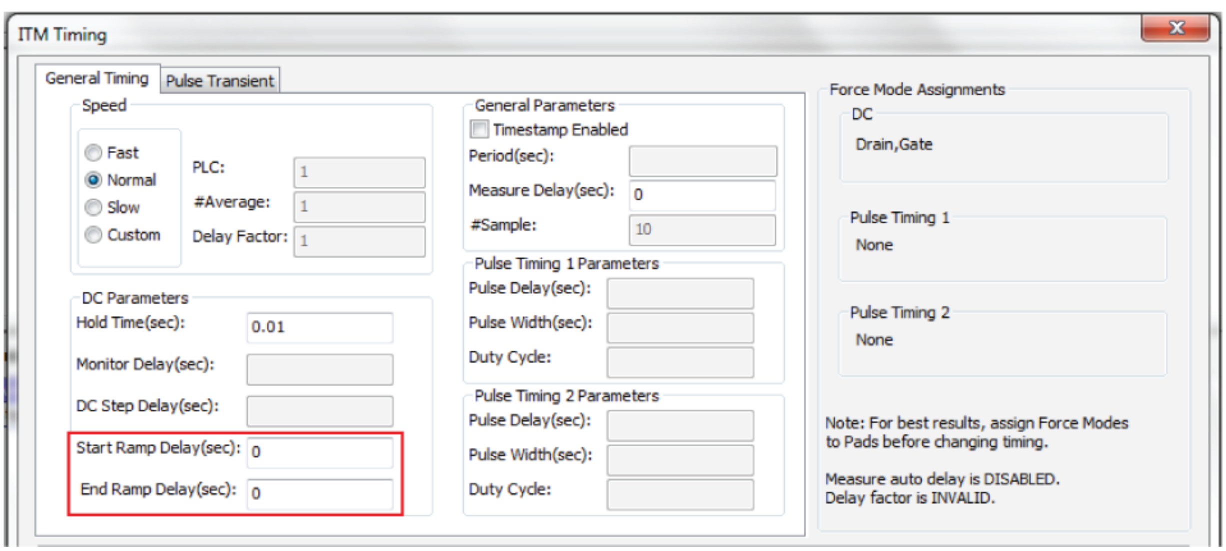


Figure 146: ITM General Timing tab settings



If the Start Ramp and End Ramp are selected, the test applies a Soft Ramp before and after the bias or sweep. The Soft Ramp is from 0 to the set value (bias value or first and last sweep or list value). If the bias or first or last sweep, list point is 0 and no soft ramp is necessary.

In the Advanced settings dialog, when setting the Ramp step, the number of steps is automatically calculated.

- **Start Ramp Step**: Sets the step size for the start ramp.
- **Stop Ramp Step**: Sets the step size for the stop ramp.
- **Num of Steps**: The number of steps defined for all the SMUs, however, the start and stop steps are defined separately. The default value for Num of steps is 1. The actual value depends on the Force Value and the Start and Stop Ramp Steps.

In the ITM Timing settings dialog, the Start Ramp Delay and End Ramp Delay determines the step time of the Soft Ramp.

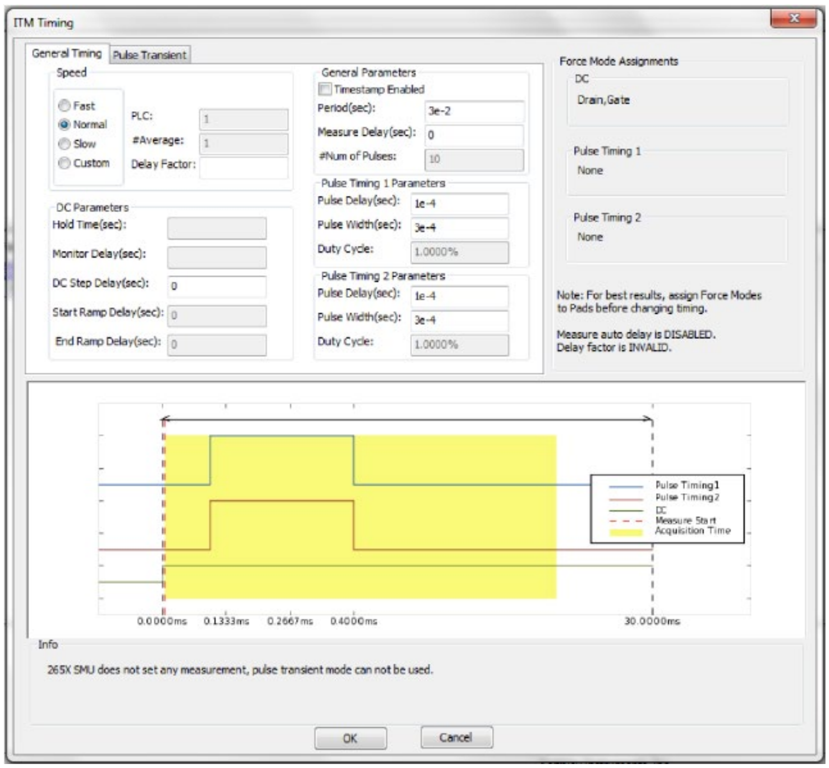
GND, Open

The GND function maintains a constant voltage at 0 V. Open is equivalent to high impedance mode (HIGH_Z) set to ON.

ITM timing settings

When you select the ITM Timing icon in the ITM Definition tab, the ITM Timing dialog opens, as shown in the following figure. Configure the ITM Timing in the interface settings.

Figure 147: ITM Timing dialog



NOTE

In Force Mode Assignments (on page 5-84), it is recommended to assign Force Mode to Pads (SMUs) in the SMU Parameters table before changing timing settings (see Force Mode Assignments (on page 5-84) for more detail).

Speed

The Speed area of the ITM Timing dialog allows you to:

- Locally select the Fast, Normal, Slow, or Custom measurement speed mode.
- Configure Custom speed settings in the Custom measurement speed mode.

Fast, Normal, Slow, and Custom modes are defined as follows:

- **Fast**: Optimizes speed at the expense of noise performance. Choose this option for fast measurements where noise and settling time are not concerns.
- **Normal**: The default and most used setting. It provides a good combination of speed and low noise (the best setting in most cases).
- **Slow**: Optimizes the low-noise measurements at the expense of speed. If speed is not a critical consideration, choose this option when you need the lowest noise and most accurate measurements.
- **Custom**: Allows you to fine-tune the timing parameters to meet a particular need. With Custom, you can configure the integration time and the number of readings to produce a composite setting. The entries in the PLC and # Average fields control the A/D (analog-to-digital) converter integration time used to measure a signal. A short integration time for each A/D conversion results in a relatively fast measurement speed, at the expense of noise. A long integration time results in a relatively low noise reading, at the expense of speed. The integration time setting is based on the number of power line cycles (NPLCs):

  For 60 Hz line power, 1.0 PLC = 16.67 ms (1/60 of a second).

  For 50 Hz line power, 1.0 PLC = 20 ms (1/50 of a second).

If you selected the Custom measurement Speed mode, any value between 0.001 and 25 NPLC can be entered. If you selected the Fast, Normal, or Slow measurement speed mode, ACS sets the PLC integration time correspondingly. The next table summarizes the allowed settings for each of the measurement speed modes.

Speed mode	NPLC	# Average
Fast	0.001	1
Normal	1	1
Slow	10	1
Custom	0.001 to 25	≥ 1

Delay factor

After applying a forced voltage or current, a SMU waits for a delay time before making a measurement. The delay time allows for source settling. The default delay time is pre-programmed and range-dependent to allow for the long settling times for very low current ranges. The applied delay time is a multiple of the default delay time and the value in the delay factor (see the previous figure).

Applied delay time = (default delay time) x (delay factor)

For example, if the default delay time is 1 ms and the delay factor is 0.7, the applied delay time is 0.7 ms (1 ms x 0.7).

If you select the Custom measurement speed mode, you can enter a custom delay factor of 0 to 100. If you select the Fast, Normal, or Slow measurement speed mode, the SMU sets an appropriate fixed delay factor.

Measurement speed	Delay factor
Fast	0.7
Normal	1.0
Slow	1.3
Custom	0 through 100

General parameters

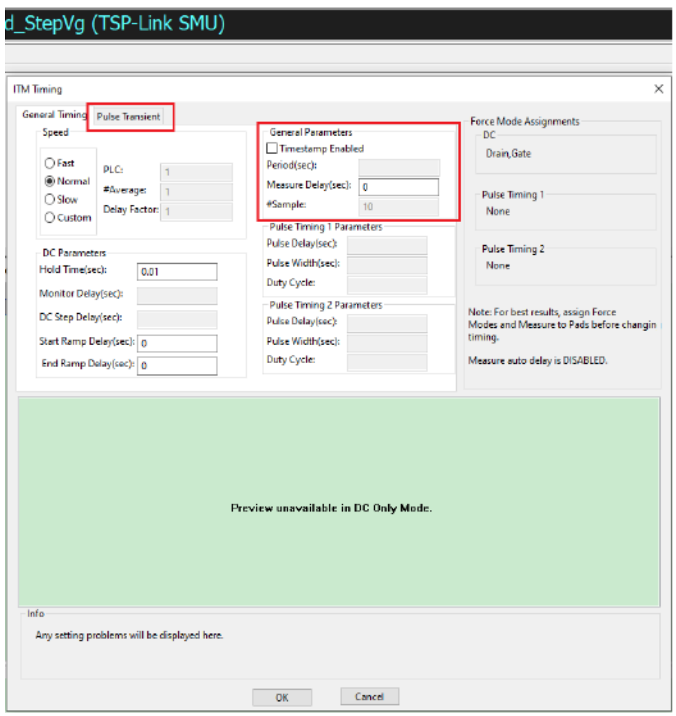
Timestamp Enabled: An additional data column (Time) in the Data tab will be added to record the measurement time.

Period(sec): The pulse period, in seconds, that is applied to Pulse Timing 1 and Pulse Timing 2.

Measure Delay(sec): The delay time before measurement.

#Num of Pulses: The number of pulses when all the SMUs are set to bias in Pulse Available.

Figure 148: General Parameters ITM timing settings



DC Parameters

Hold Time(s): DC Only. When the test is started, the Hold Time (HT) provides extra settling time for the initial step change of the source. The Hold Time is a global setting. Therefore, it is the same for all SMUs in the test system. Note that the Hold Time is applied at the start (only) of each sweep.

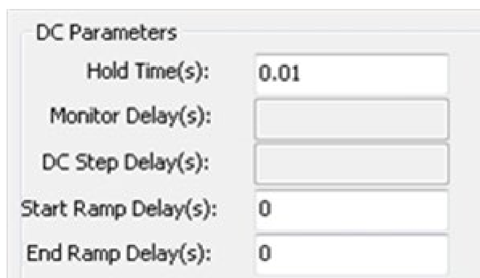
Monitor Delay(s): Sets the monitor delay time in the second sweep phase in of the sweep2 function.

DC Step Delay(s): Sets the time in seconds from the start of the period to the start of the transition for DC signals.

Start Ramp Delay(s): Sets the delay time of steps in the start ramp of the soft ramp bias.

End Ramp Delay(s): Sets the delay time of steps in the end ramp of the soft ramp bias

Figure 149: DC Parameters ITM timing settings



Pulse Timing Parameters

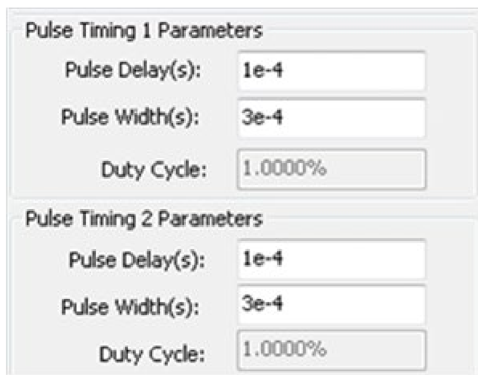
ACS allows you to configure a total of two pulse timing configurations.

Pulse Delay(s): Sets the time in seconds from the start of the period to the start of the rising edge for pulse signals assigned to the Force Mode Pulse Timing 2 Parameters.

Pulse Width(s): Sets the width of the pulse in seconds for pulse signals assigned to the Force Mode Pulse Timing 2 Parameters.

Duty Cycle: Displays the duty cycle for pulse signals assigned to the Force Mode Pulse Timing 2 Parameters. Duty cycle is calculated by dividing the pulse width by the period.

Figure 150: Pulse Timing Parameters for an ITM



Force Mode Assignments

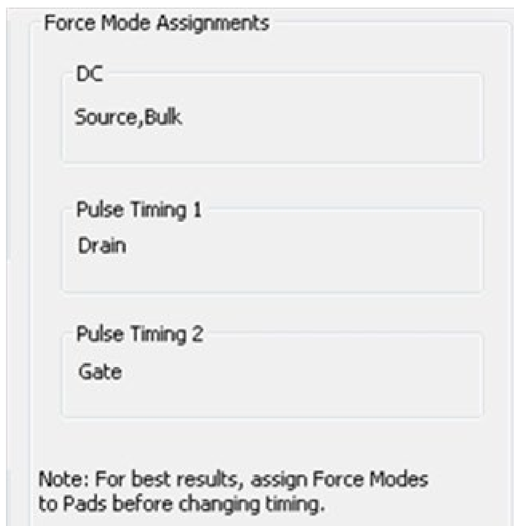
The force mode assignments section displays each pad (SMU). It is recommended to assign Force Mode to Pads (SMUs) in the SMU Parameters table before changing timing settings.

DC: Displays the SMUs by Pad Name that have their Force Mode assigned to DC.

Pulse Timing 1: Displays the SMUs by Pad Name that have their Force Mode assigned to Pulse Timing 1.

Pulse Timing 2: Displays the SMUs by Pad Name that have their Force Mode assigned to Pulse Timing 2.

Figure 151: Force Mode Assignments for an ITM



Timing schematic

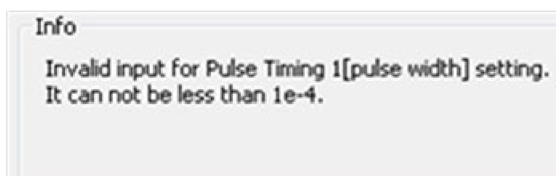
The timing schematic is displayed according to the settings and is only visible in Pulse Available mode. DC Only mode has no timing schematic.

Figure 152: Timing schematic for an ITM

ITM Timing dialog Info

Information about any problems related to the timing settings will be displayed in Info.

Figure 153: Info in the ITM Timing dialog



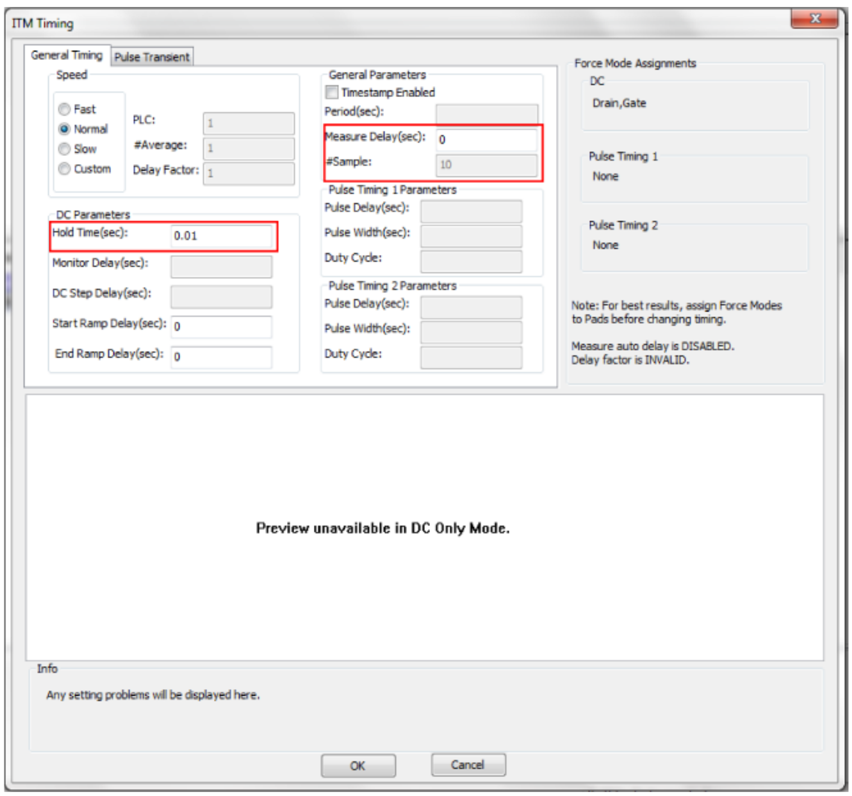
DC Only Timing settings

In DC Only mode, two types of tests can be implemented: Sweep or sampling.

Sweep: If any terminal of the device under test is configured for a dynamic forcing function (step or sweep forcing), the sweeping mode is automatically enabled.

#Sample: When all the SMUs are set to bias and sets the number of samples for measurement. It is only available when all the SMUs are set to bias forcing.

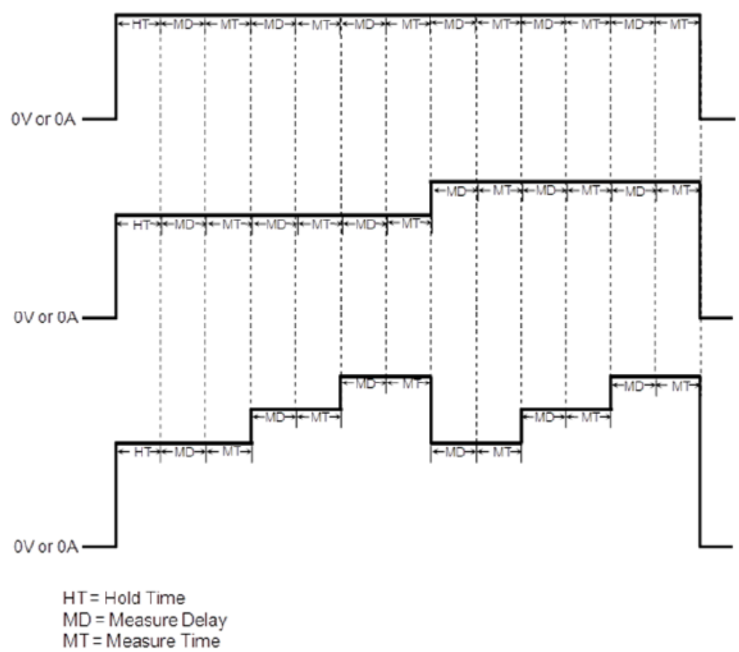
Figure 154: ITM Timing for DC only



Measure Delay: If you are using sweep forcing and want additional settling time before each measurement, you can specify an additional delay in the sweep delay edit field. You can specify a measure delay from 0 to 1000 seconds. The default Measure Delay is 0 seconds. Setting measure delay to 0 does not disable Model 26xx auto delay. That setting is in **Preferences** > **Instruments** under **26xx Settings**.

Hold Time: The starting voltage or current of a sweep may be substantially larger than the voltage or current increments of the sweep. Accordingly, the source settling time required to reach the starting voltage or current of a sweep may be substantially larger than the settling times required to increment the sweep. To compensate, you can specify a hold time delay to be applied only at the beginning of the sweep. You can specify a hold time delay from 0 to 1000 seconds. The default hold time is 0 s.

Figure 155: Sweep timing diagram in DC only

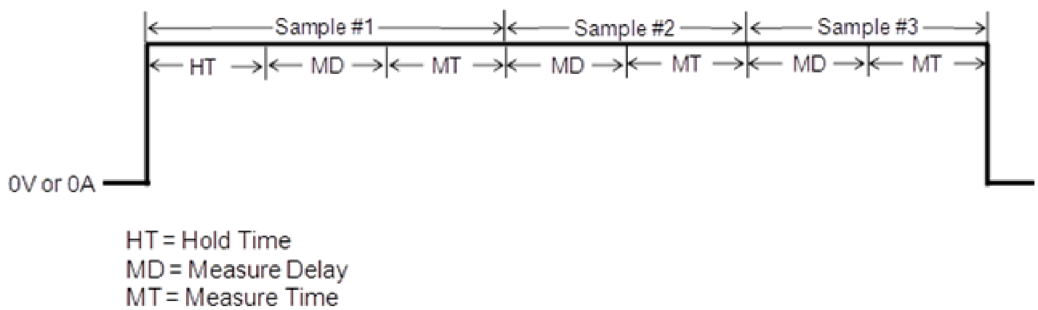


Sampling

Sampling mode measures voltage and current as a function of time while forcing constant voltage and current (bias I or bias V). For example, the sampling mode measures voltage while forcing a constant current. Time is measured relative to when an SMU applies the forced voltage or current (for example, t = 0 at the step change from 0.0 V or 0.0 A to the applied voltage and current).

ACS enables the sampling mode option only when all terminals of the device under test are configured for static forcing functions: voltage bias or current bias, and the mode must be DC only.

Figure 156: Sampling timing diagram in DC only



Pulse Available Timing settings

When the Pulse Available mode is selected, the ITM will allow DC timing and two pulse timing settings for the SMUs. All the SMUs will have the same measure speed, period, and measure delay. However, they have separate transition delays (DC step delay, pulse delay), and the two pulse timings have their own pulse width.

The following figure indicates the settings for the pulse timing:

1. Measure speed: Measurement acquisition time (in PLC) and number of individual measurements to average together for the result.
2. Delay time for DC bias SMU.
3. Pulse period and measure delay.
4. Pulse delay and pulse width for each pulse timing. The Duty Cycle is calculated according to pulse period and width.
5. Force mode assignments for each pad.
6. Timing diagram according to the settings.
7. Description of any problems with settings.

Figure 157: Pulse timing settings

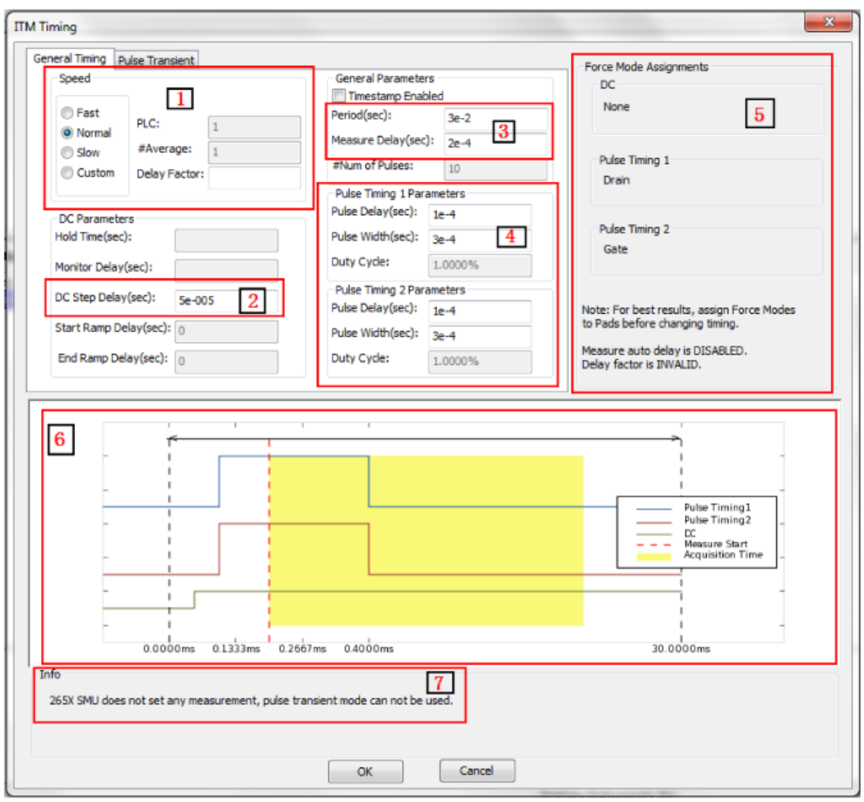
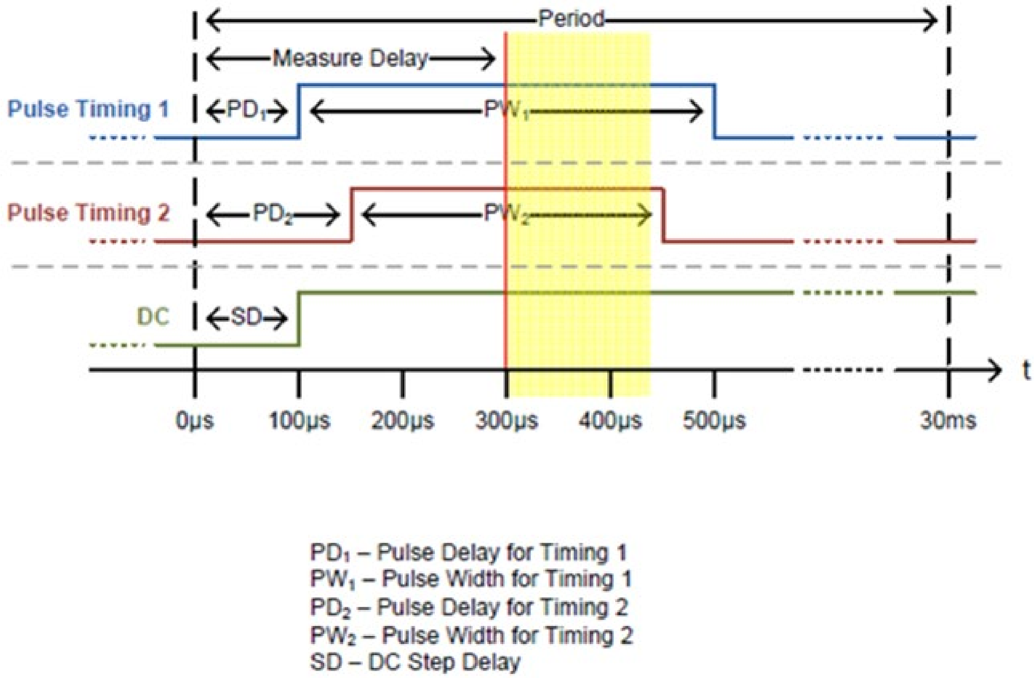


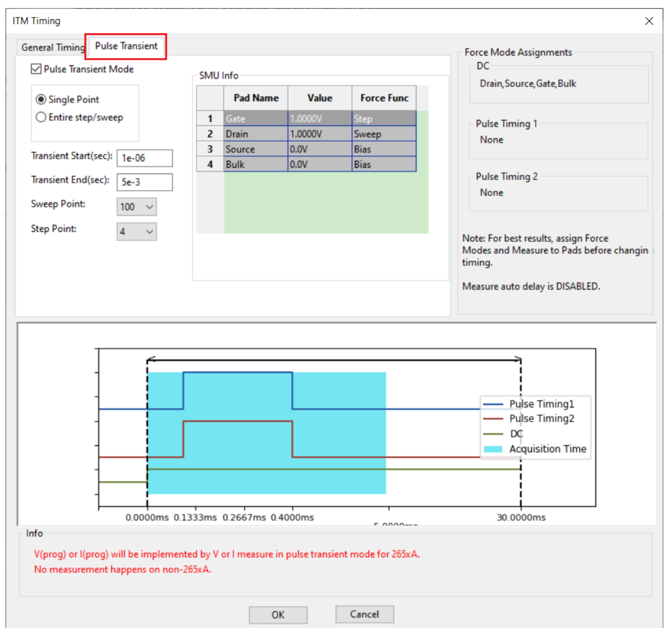
Figure 158: Pulse timing diagram



Pulse Transient

Pulse Transient mode uses Fast ADC, which is available on certain SMU models (refer to the *Model 2651A* or *2657A High Power System SourceMeter Instrument Reference Manual*) to capture the waveform for a single point in the step and sweep sequence. It is useful for determining the proper measurement delay setting for the step and sweep sequence by capturing the settling time of the SMU outputs. It is also useful for debugging troublesome data points in the step and sweep sequence without the need for an oscilloscope. It is only available for 265xA SMUs in Pulse Available mode. The Pulse Transient mode timing settings are in the ITM Timing setting dialog on the Pulse Transient tab, as shown in the following figure.

Figure 159: Pulse Transient mode settings



Single Point or Entire step and sweep: Selects whether to source a single point from the step and sweep sequence or the entire step and sweep sequence to perform the pulse transient capture. Single Point only sources the selected point of the step and sweep sequence. The test does not have to run through the entire sequence. Entire step and sweep outputs the entire step, and sweep sequence and makes the transient measurement on the selected point. The step and sweep is useful when the measurement on the device changes due to effects like device self-heating. By outputting the entire step and sweep sequence, the captured waveform will be an accurate representation of what is happening during that point in the step and sweep sequence.

Transient Start(s): Sets the time in seconds from the start of the period to the start of the acquisition.

Transient End(s): Sets the time in seconds from the start of the period to the end of the acquisition.

Sweep Point: Sets the point in the sweep that will be captured.

Step Point: Sets the step point that will be captured.

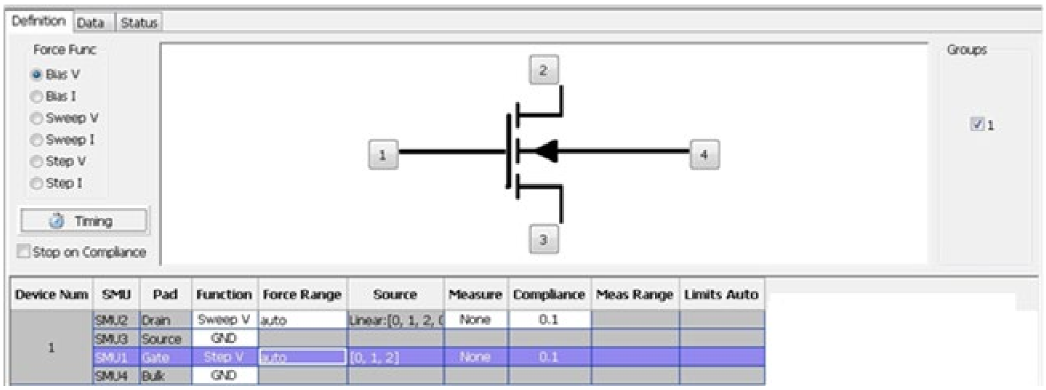
The sweep point and the step point together appoint a certain period for the waveform capture. The SMU table lists the pad name, force value, and force function of the selected period.

Model 42xx SMU ITM

This ITM is based on the Model 42xx SMU device, and it supports DC measurements. Each ITM is associated with a specific ITM work area. An ITM work area displays the ITM's definition (configuration), test data, and status information, as shown in the following figure.

Note that Model 42xx ITMs return a test status code in the Data sheet, in addition to the other results.

Figure 160: Definition tab of the Model 42XX SMU ITM



Timing: This button opens the ITM timing dialog, where you can set detailed timing settings for pulse and synchronization between SMUs.

Stop on Compliance: Enables the test to abort and turn off the SMU output when it reaches compliance.

Device schematic: Shows the device image and pad settings according to the device level settings in the configuration navigator.

Groups: Select the group or groups where you want to run the test. The group is determined by the GPIB or ethernet address. Make sure that all the groups you select have the same hardware configurations as your ITM.

SMU parameters: SMU Parameters table is where you can set your parameters for testing.

Model 42xx SMU ITM set up

The following figure displays the items to set up testing to force and measure the SMU.

Figure 161: Model 42XX SMU parameters table

Device Num	SMU	Pad	Function	Force Range	Source	Measure	Compliance	Meas Range	Limits Auto
1	SMU2	Drain	Sweep V	auto	Linear:[0, 1, 2, (	None	0.1		
	SMU3	Source	GND						
	SMU1	Gate	Step V	auto	[0, 1, 2]	None	0.1		
	SMU4	Bulk	GND						

Device Num: Sub-device number defined in the device work area.

SMU: SMU number defined in the device work area.

Pad: Pad name defined in the device work area.

Function: Force function type.

Force Range: Specifies the SMU range used to force on all functions (except V-meter and GNDU):

- Auto option commands the SMU to automatically optimize the force range as the sweep progresses. This option provides the best resolution and control when sweeping several decades.
- Numerical range options allow you to manually select a SMU range to meet your needs (the drop list contains all the supported ranges according to the SMU and force function). This range must accommodate the stop or start value, whichever is greater. For example, when forcing stop 1.5 mA, you should choose the 10 mA SMU range or the Auto SMU range. The Auto SMU range allows the SMU to select the most appropriate range.

Source: Allows you to set the source value according to the selected function. When the function is bias V or bias I, it sets the force value in this cell. When the function is sweep V, sweep I, step V, or step I, the source settings are similar to the Model 26xx SMU parameters (on page 5-61).

Measure: Selects the measurement function options:

- I: Actual measured current values.
- V: Actual measured voltage values.
- V (prog): Reports the programmed voltage. No measurement is made.
- I (prog): Reports the programmed current. No measurement is made.
- I+V: Measures both current and V.
- I+V (prog): Measures current and returns V (prog).
- V+I (prog): Measures V and returns current (prog).

Compliance: Specifies any valid SMU current or voltage compliance limit. The current compliance is present for voltage forcing functions and the voltage compliance is used for current forcing functions.

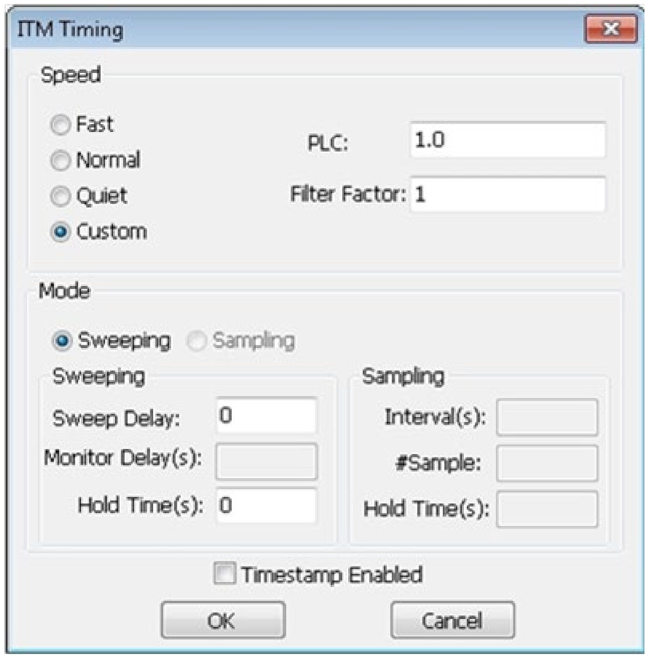
Meas. Range: Specifies the SMU range used. The current measure range is present for voltage forcing functions, and the voltage measure range is present for current forcing functions only. If current is selected in the Measure cell, ACS enforces the Force range first, and ignores the measure range. You can specify the range used by the SMU for current using the following options:

- Auto option commands the SMU to automatically optimize the current measurement range. This option provides the best resolution
- Limits Auto is only enabled when the force function is voltage and there is current measurement. This option is a compromise between the full Auto option and a fixed range option. You can specify the minimum range that the SMU uses when automatically optimizing the current measurements. This option saves time when maximum resolution at minimum currents is not needed.
- Numerical range options allow you to manually select a fixed current measurement range to suit your needs. It selects current ranges when the source function is voltage and selects voltage ranges when the source function is current. All the supported measurement ranges are listed according to the SMU type.

Limits Auto: Will be enabled after selecting Limits Auto in the Meas Range cell.

When you select the ITM Timing icon in the ITM Definition tab, the ITM Timing dialog opens. The ITM Timing dialog is where you set the measurement speed and delay time of the test.

Figure 162: 42XX SMU ITM Timing settings



Speed

The Speed area of the ITM Timing dialog allows you to:

- Locally select the Fast, Normal, Slow, or Custom measurement speed modes.
- Configure Custom speed settings in the Custom measurement speed mode.

Fast, Normal, Slow, and Custom modes are defined as follows:

- **Fast:** Optimizes speed at the expense of noise performance. It is a viable choice for fast measurements where noise and settling time are not concerns.
- **Normal:** The default and most used setting. It provides a good combination of speed and low noise and is the best setting in most cases.
- **Slow:** Optimizes the low-noise measurements at the expense of speed. If speed is not a critical consideration, it is a viable choice when you need the lowest noise and most accurate measurements.
- **Custom:** Allows you to fine-tune the timing parameters to meet a particular need. With Custom, you can configure the integration time and the number of readings to produce a composite setting. The entries in the PLC and # Average fields control the A/D (analog-to-digital) converter integration time used to measure a signal. A short integration time for each A/D conversion results in a relatively fast measurement speed, at the expense of noise. A long integration time results in a relatively low noise reading, at the expense of speed. The integration time setting is based on the number of power line cycles (NPLCs):

  For 60 Hz line power, 1.0 PLC = 16.67 ms (1/60 of a second).

  For 50 Hz line power, 1.0 PLC = 20 ms (1/50 of a second

Mode area

There are two test modes you can select in the Mode area of the ITM Timing dialog:

- Sweeping: Applies to any ITM where one or more forced voltage or current vary with time.
- Sampling: Applies to any ITM where all forced voltages or currents are static (bias I or bias V), with measurements typically made at timed intervals. For example, sampling mode records a few static measurements, or to time-profile the charging voltage of a capacitor while forcing a constant current.

For a new ITM, the Mode allows you to select the sweeping or sampling test mode. Selecting the appropriate test mode simplifies configuration options and helps to avoid errors:

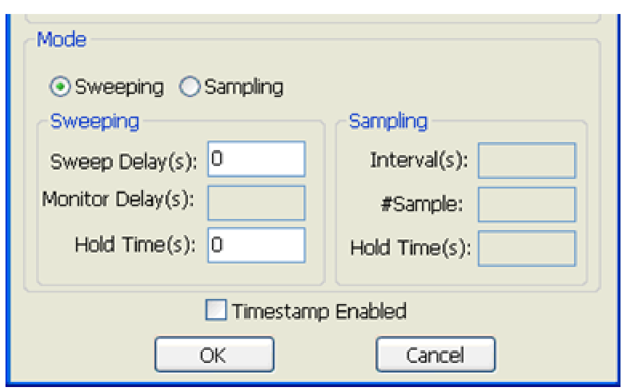
- Only the static forcing functions are configurable in the sampling mode.
- Only mode-appropriate timing options are configurable.

For an existing library ITM that is in sampling mode, the mode area allows you to change to sweeping mode if you want to change some of the static forcing functions to dynamic forcing functions.

Sweeping mode

If any terminal of the device under test is configured for a dynamic forcing function (step or sweep forcing function), the sweeping mode is automatically enabled. You can configure two sweeping mode settings in the ITM Timing dialog (Sweep Delay and Hold Time). You can configure these settings for all measurement Speed modes: Fast, Normal, Slow, and Custom.

Figure 163: ITM Sweep timing GUI

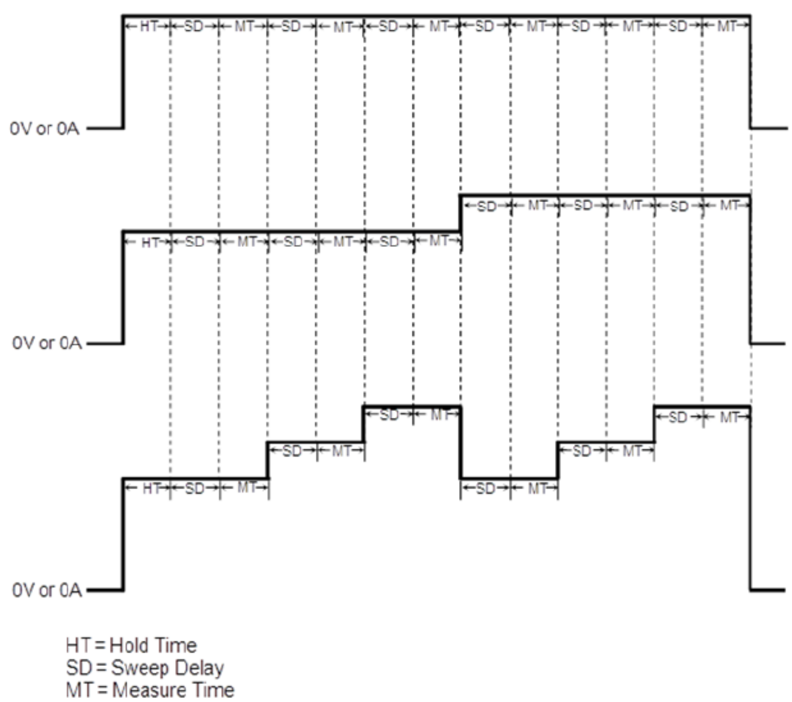


Sweep Delay: If you are using a sweep forcing function and want additional settling time before each measurement, you can specify an additional delay in the Sweep Delay box. You can specify a Sweep Delay from 0 to 1000 seconds. The default Sweep Delay is 0 s.

Hold Time: The starting voltage or current of a sweep may be substantially larger than the voltage or current increments of the sweep. Accordingly, the source settling time required to reach the starting voltage or current of a sweep may be substantially larger than the settling times required to increment the sweep. To compensate, you can specify a hold time delay to be applied only at the beginning of each sweep. You can specify a hold time delay of 0 to 1000 seconds. The default hold time is 0 s, as shown in the following figure.

When the test is started, hold time provides extra settling time for the initial step change of the source. Hold time is a global setting; therefore, it is the same for all SMUs in the test system. Note that hold time is only applied at the start of each sweep.

Figure 164: 42XX sweep mode timing diagram



Sweep Delay: Provides additional settling time for each step in the sweep. It is a global setting and is applied identically to all SMUs in the test system.

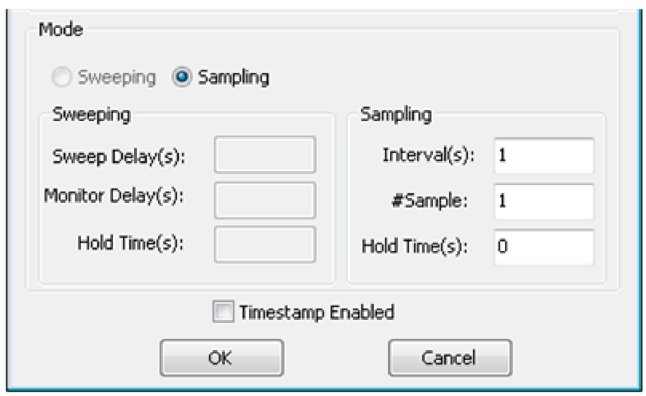
Measure Time: Determined by the number average and the A/D Integration Time (NPLC). All SMUs in the test system are synchronized. The measure time for the SMU requiring the longest measure time is the same for all SMUs in the test system.

Sampling mode

The sampling mode measures voltage and current as a function of time while forcing constant voltage and current (bias I or bias V). For example, sampling mode measures voltage while forcing a constant current. Time is measured relative to when the SMUs apply the forced voltage or current (that is, t = 0 at the step change from 0.0 V or 0.0 A to the applied voltage or current).

ACS enables the sampling mode option only when all terminals of the device under test are configured for static forcing functions: voltage bias or current bias. If any terminal of the device under test is configured for a step or sweep forcing function, the sampling mode option is unavailable.

Figure 165: ITM sample timing settings



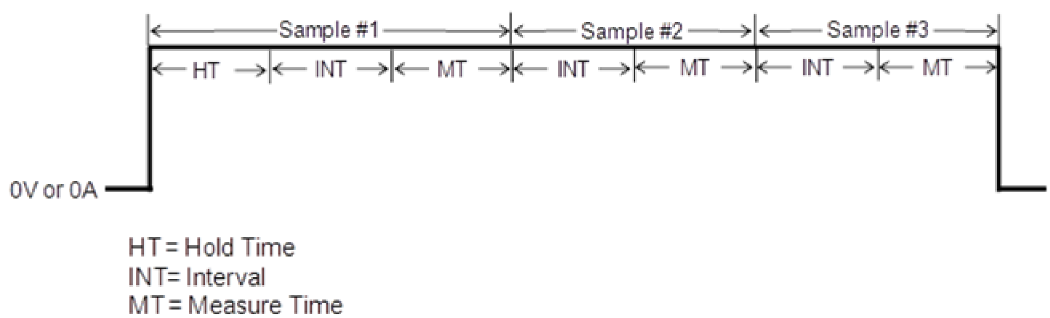
If an ITM is configured for the sampling mode, you can configure three sampling mode settings: Interval, #Sample, and Hold Time.

Interval(s): Specifies the time between measurements (also known as data points). Intervals can be set from 0 to 1000 seconds.

#Sample: Specifies the number of data points to be acquired. Samples can be set from 1 to 4096.

Hold Time(s): After initial application of voltage or current by the SMUs, the source settling times can be substantial. To allow for settling, you can specify extra hold time (delay) to be applied before making the first measurement. You can specify a hold time from 0 to 1000 seconds, as shown in the following figure. If needed, Hold Time can be used to allow for extra source settling after the initial application of voltage or current from the SMU. Hold Time is a global setting and is the same for all SMUs in the test system.

Figure 166: 42XX sampling mode timing diagram



Interval: Specifies the time between measurements (also known as data points). Interval (INT) can be set from 0 to 1000 seconds.

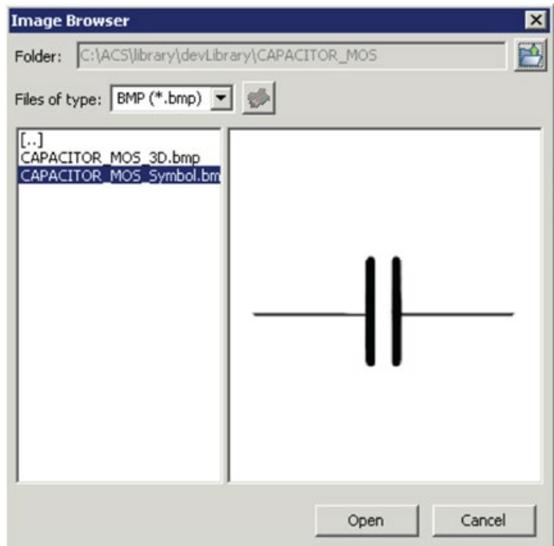
Measure Time: Determined by the A/D Integration Time. All SMUs in the test system are synchronized. The measure time for the SMU requiring the longest measure time is the same for all SMUs in the test system.

ITM test configuration

To create a test configuration:

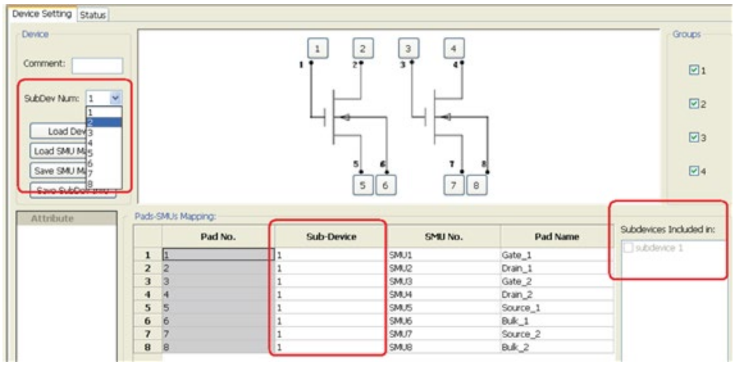
1. Create a device by selecting device function from the toolbar.
2. Select the device in the Image Browser, as shown in the following figure.

Figure 167: Device type Image Browser



- Define the number of the subdevice using the SubDev list. The created subdevices with checkbox are listed in the Subdevices Included In box.
- Select any row in the Pad SMUs Mapping box. Set the SMU as instructed in the ITM section (refer to the Device Setting tab in the following figure).
- In the Subdevices Included in section of the screen, select the checkbox of the subdevices.
- Match SMU, Pads, and Sub-device to the actual physical connection to complete the test device structure definition.

Figure 168: Multiple subdevice setting



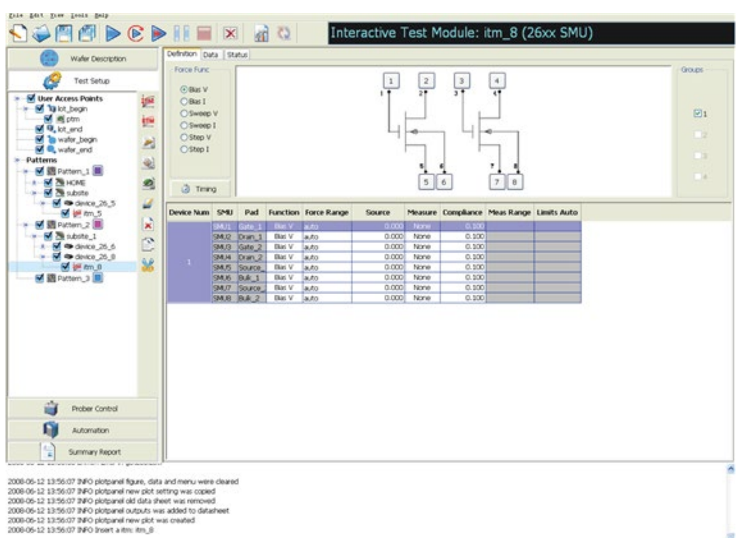
NOTE

If a Pad is set as NONE or GND, it will not appear in test set up page (users should make GND connection themselves). If a pad is set as Common, it is shared by all subdevices. The pad name cannot be the same. If the same name is used, the Invalid Pad Name error message is displayed.

Add a new ITM:

1. From the side toolbar, select the ITM function. A new ITM multi-subdevice is added to the configuration navigator, as shown in the following figure.

Figure 169: ITM GUI of multiple subdevices



2. After creating the new ITM, configure it to meet your test requirements. There are three tabs in ITM dialog that you can use for configuration: Definition tab, Data tab, and Status tab.

Definition tab

Device Num: The subdevice number is defined in the device work area based on hardware connection.

SMU: The number is defined in the device work area, based on hardware connection.

Pad: The name is defined in the device work area, based on the hardware connection.

Function: Force function type.

In the Function category, you can select the function you need performed. For example, you can choose bias V, bias I, sweep V, sweep I, step V, step I, V_meter, GND, or OPEN depending on your testing needs.

Force Range: Specifies the SMU source range used for the voltage source, per the following options:

- The Numerical Voltage Range options (200 mV, 2 V, 20 V, 200 V) allow you to manually select the SMU range for your needs.
- The Auto option commands the SMU to automatically optimize the voltage range as the step voltage progresses. This option provides the best current resolution and control when stepping several decades.

The current source specifies the type (bias I, sweep I, and step I) used to force the current, per the following options:

- The Auto option commands the SMU to automatically optimize the current range as the step current progresses. This option provides the best current resolution and control when stepping several decades.
- The Numerical Current range options (100 nA, 1 µA, 100 µA, 1 mA, 10 mA, 100 mA, 1 A, 1.5 A, 10 A) allow you to manually select a SMU range for your needs. This range must accommodate the Stop or Start value, whichever is greater.

For example, when forcing a stop at 1 mA, you typically choose the 1 mA SMU range or the autorange, which allows the SMU to pick the most appropriate range based on the current level.

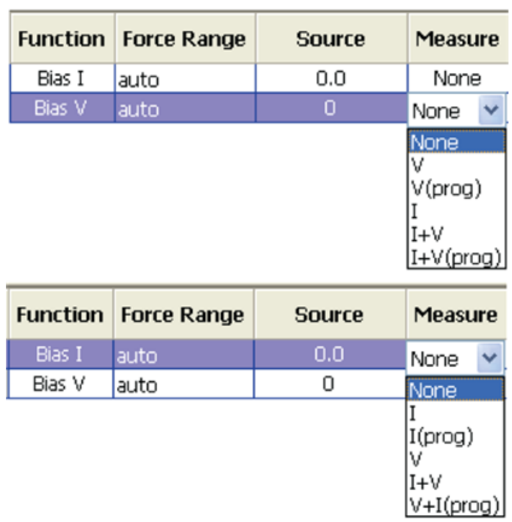
Source: Specifies any valid value that can be applied by the SMU. The unit of measure is volts or amps.

Since there are two types of tests that are supported (spot and sweep), there are two types of data that can be entered in the source dialog. For a spot test, enter a single value. For a sweep test, double-click the source cell. The Sweep V/I settings dialog opens. Enter the Start, Stop, Step and Points values. Select **OK**.

Figure 170: Sweep V settings

Measure: The unit is amps or volts with the following options, as shown in the following figure:

Figure 171: Measure Variable sub-list box



- **I:** Actual measured current.
- **V:** Actual measured voltage values.
- **V (prog):** The programmed voltage, as requested forced voltage values. For example, 2.5 V is to be forced at a transistor drain terminal. The programmed value (V (prog)) is exactly 2.5 V, even if the measured value (V) is 2.4997 V. Under the programmed option, the reported value is exactly 2.5 V, even if the measured value would be 2.4997 V.
- **I (prog):** The programmed current.
- **I+V:** Measures current and voltage.
- **I+V (prog):** Measures current and returns voltage (prog).
- **V+I (prog):** Measures voltage and returns current (prog).

Compliance: The Compliance box specifies any valid SMU current or voltage compliance limit. The current compliance is present for voltage forcing functions and the voltage compliance is used for current forcing functions. Here, forcing bias V, the compliance is current compliance. The unit is Amp.

Meas Range: The measurement range list specifies the SMU range used. The current measure range is present for voltage forcing functions and the voltage measure range is present for current forcing functions only.

If the measure variable is set to I, I+V, or I+V(prog), the measure ranges that can be used by the SMU for current are as follows:

- The Auto option commands the SMU to automatically optimize the current measurement range. This option provides the best resolution.
- The Limited Auto option is a compromise between the full Auto option and a fixed range option. Here one can specify the minimum range that the SMU uses when automatically optimizing the current measurements. This option saves time when maximum resolution at minimum currents is not needed.
- The numerical range options (100 nA, 1 µA, 10 µA, 100 µA, 1 mA, 10 mA, 100 mA, 1 A, 1.5 A, 10 A) allow you to manually select a fixed current measurement range for your needs.

If the measure variable is set to V, I+V, or V+I(prog), the measure ranges that can be used by the SMU for voltage are as follows:

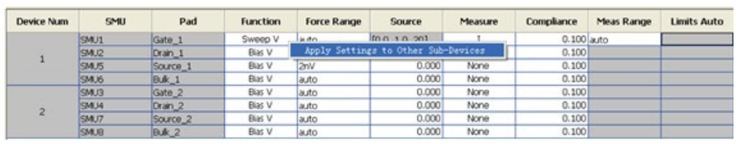
- The Auto option commands the SMU to automatically optimize the voltage measurement range as stepping and sweeping progresses. This option provides the best resolution when the measurements span several decades.

- The Numerical Range options (200 mV, 2 V, 20 V, 200 V) allow you to manually select a fixed voltage measurement range to suit your needs.

Set Force and Measure Function

Configure the forcing and measure functions for each device terminal of a subdevice. If you want to know how to set these functions, refer to the ITM single device for detailed instructions. After setting up a device, you can right-click any cell that belongs to this device, as shown in the following figure.

Figure 172: Apply Settings to Other Sub-Devices



After selecting this function, the same setting is applied to all devices. You can also edit the devices to suit the demands of the test.

Set the Timing (Optional)

Enter time setting for the entire test as instructed in the ITM single device section or use the default setting.

Data tab

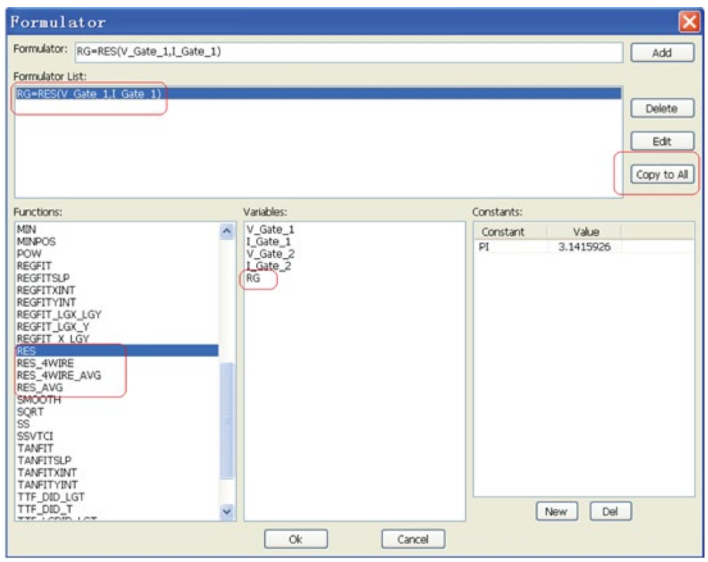
Set the Formulator (Optional)

There are some specific features designed for the ITM that are in multiple subdevices, including the "Copy to All" function and the formulas RES, RES_4WIRE, RES_4WIRE_AVG, and RES_AVG. These features are only used if the ITM is used in multiple subdevices.

To start the Formulator:

1. Open the Data tab for the test data that you want to analyze.
2. In the Data tab, select the Formulator. The Formulator dialog opens, as shown in the following figure.

Figure 173: Formulator of ITM with multisubdevice



3. Edit the function you want to analyze, then select **Copy to All**. This formula is used by all who are using the same format variables generated by different subdevices.
4. Once the RG=RES(V_GATE_1,I_GATE_1) is added, select **Copy to All**. This Formulator function applies to V_GATE_2, and I_GATE_2. And RG change to RG_1 and RG_2 at same time. RG_1 and RG_2 are listed in output data sheet of DATA tab.

These Formulator functions are used in the ITM with multi-subdevices: RES, RES_4WIRE, RES_4WIRE_AVG, RES_AVG. For more information about how to use the Formulator, refer to Formulator arguments and constants (on page 5-250).

Model 4200A ITM configuration

NOTE

When the ACS software is controlling the Model 4200A in net mode, make sure the ethernet cable is connected to the personal computer and the Model 4200A.

There are two parts you configure for a Model 4200A ITM:

- Hardware configuration
- ITM configuration

Configure Model 4200A hardware with ACS software

To configure Model 4200A-SCS hardware with ACS software:

1. Launch KCON by selecting **KXCI Settings**, as shown in the following figure.

Figure 174: Launch KCON

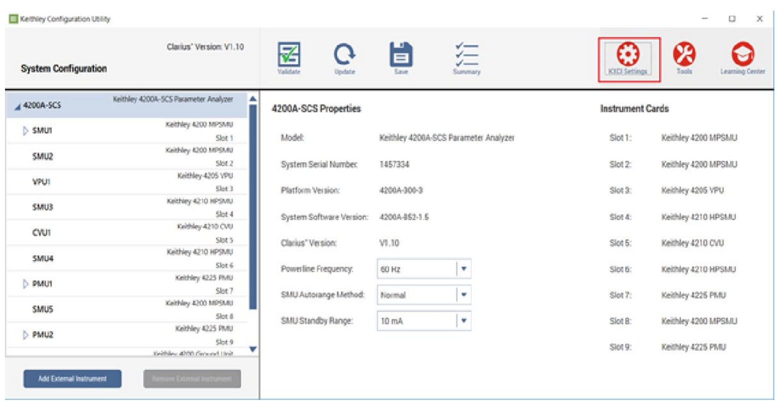
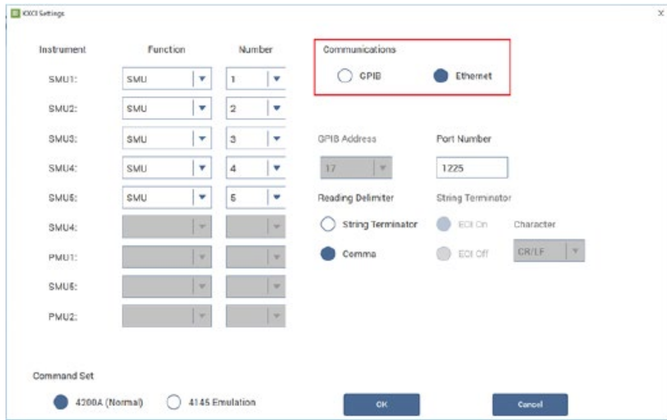
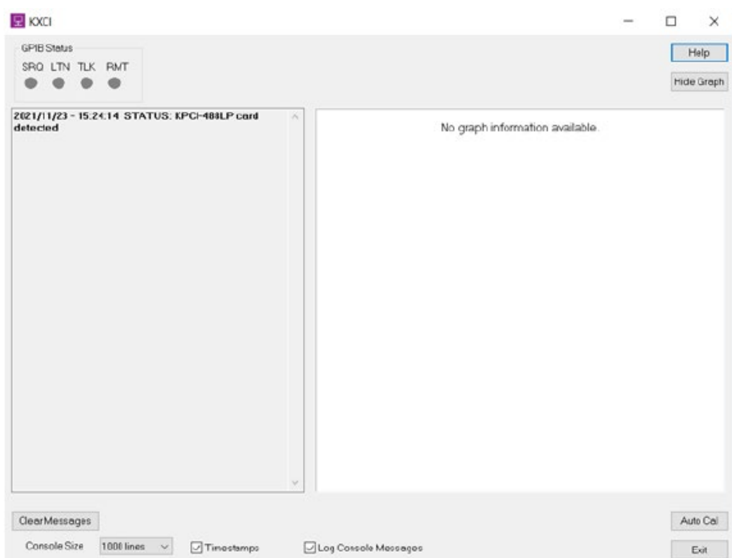


Figure 175: Set Model 4200A-SCS communication mode



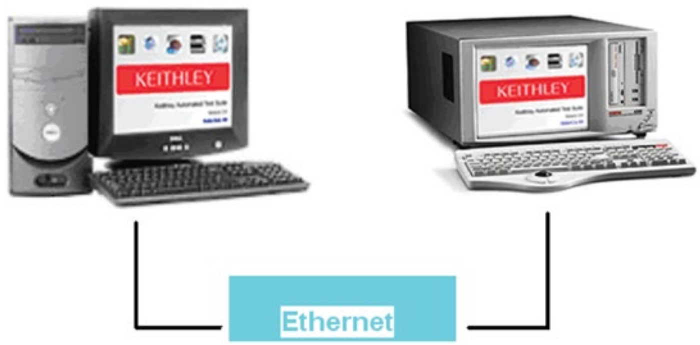
2. Under Command Set, select **4200A (Normal)** communication mode.
3. Select **OK** to launch KXCI.

Figure 176: Launch KXCI



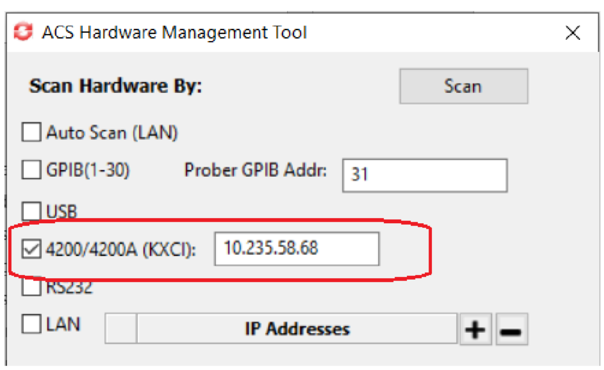
4. Connect the Model 4200A to ethernet through netlink.
5. Write down the IP address (you need this for step 9).
6. Install ACS software on a system computer or laptop and connect the system computer or laptop to ethernet through netlink.

Figure 177: ACS software on Model 4200A-SCS computers



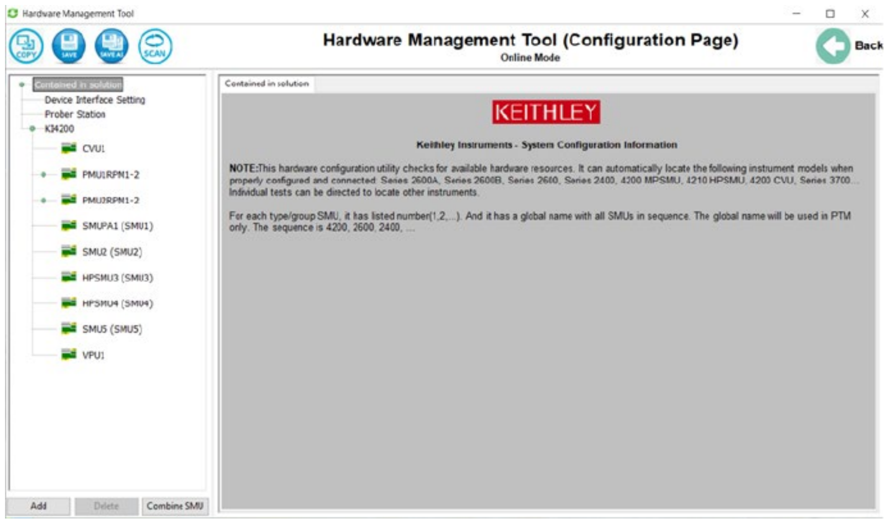
7. In the Hardware Management Tool, configure the settings for the 4200A-SCS. Make sure the Hardware Management Tool is in Online mode. Refer to "Hardware Management Tool" in the *Automated Characterization Suite (ACS) Advanced Features Reference Manual* for more information.
8. In the Hardware Management Tool, select KXCI, as shown in the following figure.
9. Add the IP address by selecting the plus symbol (**+**) and enter the IP address (saved from step 5).

Figure 178: Set the 4200A-SCS IP address



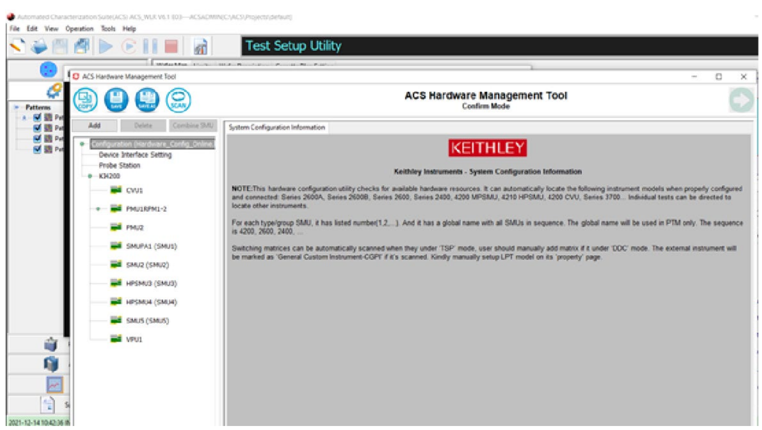
10. Review the Hardware Management Tool (Configuration Page), as shown in the following figure. Make sure the hardware information matches the physical configuration and save the configuration.

Figure 179: 4200A-SCS configuration in the Hardware Management Tool



11. Start ACS and select **Configure Hardware** from the **Tools** menu.
12. Confirm that the hardware in the Hardware Management Tool matches your physical configuration. ACS software communicates with the 4200A-SCS instrument.

Figure 180: Confirm the hardware matches

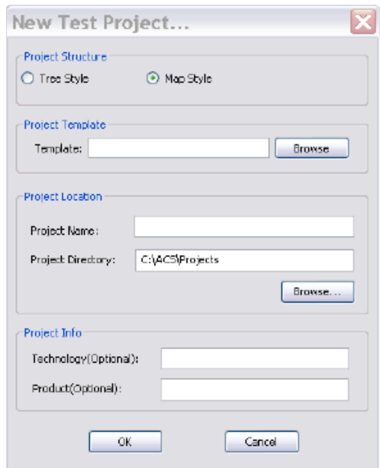


Configure a Model 4200A-SCS ITM test module

To Configure a Model 4200A-SCS ITM test module:

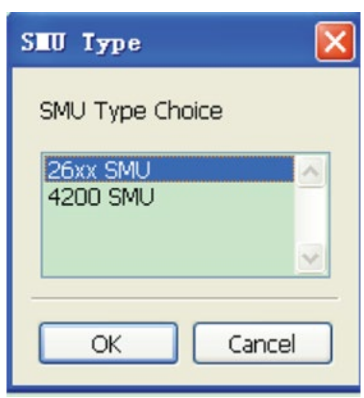
1. Select **New** from the **File** menu.
2. Select **Tree Style** in the Project Structure and input a **Project Name**.

Figure 181: Define New Test Project



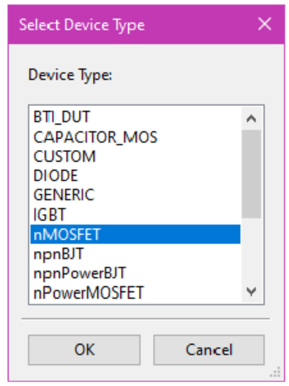
3. Select the **New Device** icon. The SMU Type dialog opens.
4. Select the **4200A SMU** and select **OK**.

Figure 182: SMU Type



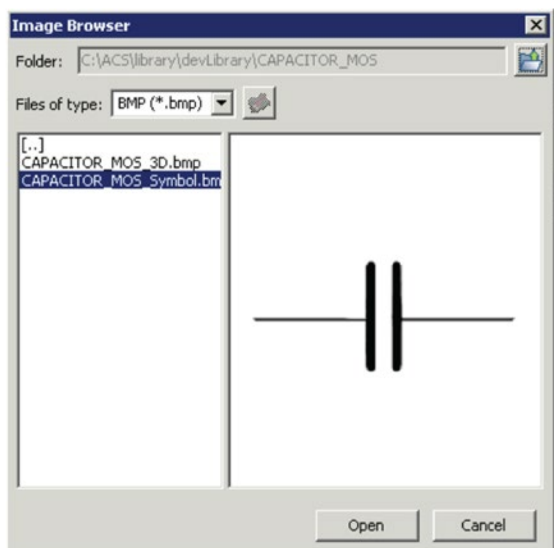
5. The Select Device Type dialog opens, as shown in the following figure.

Figure 183: Select Device Type



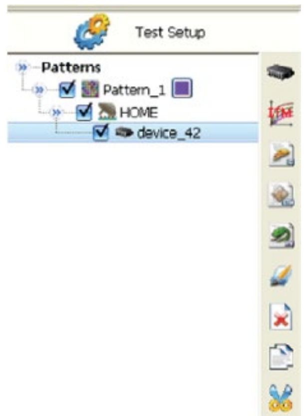
6. Select a device. An image browser dialog opens.
7. Choose a graphic and select **OK**. A new device is created.

Figure 184: Device type Image Browser



8. Select the new device.

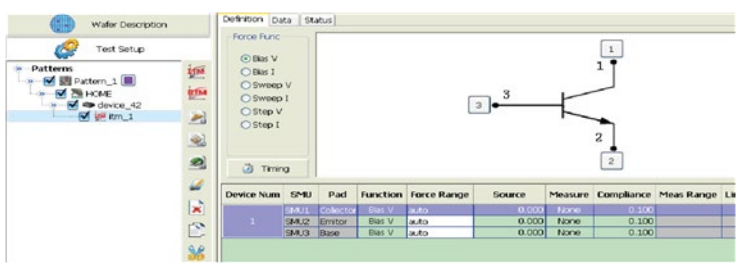
Figure 185: Select a new device



9. Select the ITM icon in the vertical toolbar to insert a new test module for this device. An ITM test module is created for the device, as shown in the following figure.

Follow the steps in Match the physical and virtual connections manually (on page 5-116) to configure the 4200A ITM parameters.

Figure 186: New Model 4200A-SCS ITM



ITM test configuration summary

The following topics show you how to configure your specific ITM test.

Match the physical and virtual connections manually

To match the physical and virtual connection:

1. In the Test Setup panel, select the ITM that needs configured or reconfigured. The Definition tab of the ITM work area opens by default.
2. In the Definition tab, review the virtual connections for each terminal.
3. Make sure that the physical device connections match the virtual (Definition tab) device connections. If necessary, turn off the instruments and correct the physical connections.

CAUTION

The physical device-terminal connections must accurately match virtual connections to avoid test error results and potential device damage.

Configure forcing for each device terminal

In the force function area, select the cell and select the SMU terminal. The SMU is assigned to the corresponding table and is highlighted.

Configure the bias function

To configure the bias function:

1. Enter the current or voltage value and range to be forced for a static forcing function.
2. Select the measure variable and corresponding measure range and default or configured limited auto.
3. Enter the default or the configured compliance.

Configure the step and sweep function

To configure a step or sweep function:

1. Enter the start, stop, step, and sweep points for a sweep forcing function.
2. Enter start, stop, and step points for a step forcing function.
3. Enter the default or configured force range.
4. Select the measure variable, and the corresponding measure range, and limited auto, if enabled.
5. Enter the default or configured compliance.
6. Repeat all the steps above for the remaining device terminals.

Enter the time setting

To enter the time:

1. Select the test mode before configuring a new ITM.
2. Add delays for sweeping and hold time in the test-mode sweeping tests or configure the timing for sampling test-mode tests.
3. Enable a timestamp to record each measurement.

NOTE

Make sure a device is configured correctly in each step. If you create a multigroup test, configure every device as indicated previously, following it step-by-step, making sure that all the devices are configured correctly.

Script Test Module (STM) configuration

An STM is a script that refers to a file or program that can be created using the Test Script Builder.

Creating a script using Test Script Builder is discussed in the Writing or modifying a script (on page 5-126) section.

After STMs are inserted into a project plan, they must be configured to meet the test requirements. This section describes the connection of project plan STMs to your modules and libraries, and how to enter and edit module parameters.

Project Plan STM configuration

Open an STM. The STM displays in the STM work area on the Script tab. You can connect or reconnect the name of an STM to an existing ACS-created user module using the STM Script tab of the STM work area.

In the STM Script tab, check the SMU to match the physical connection. Change the inputs and configure or change the script, if necessary. Save the configuration using Save on the toolbar.

Open a STM

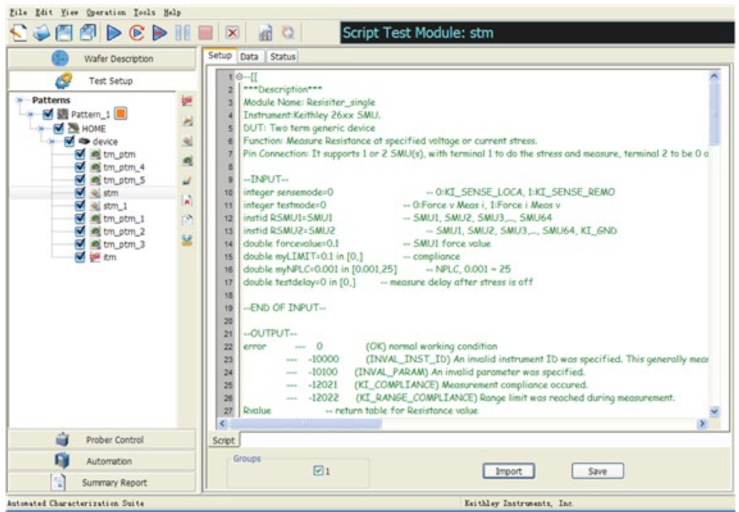
An STM work area allows you to enter information that defines a given STM. An STM work area also allows you to view and analyze test data created by the STM, both numerically and graphically. The tabs associated with the STM work area are:

- The Data tab displays the STM results in its Data worksheet in a spreadsheet format. This spreadsheet data sheet allows you to perform data analysis. The spreadsheet, displays the same output information as the Script tab. Cells in the spreadsheet may be hot-linked to cells in the Script tab. The Data tab also allows you to create and export graphs of the test and test-analysis results. The Data tab provides flexible plot-data selection, formatting, annotation, and numerical coordinate display (using precision cursors).
- The Status tab monitors the current configuration status of the STM, reporting its readiness for use and recommending additional preparations if necessary. A test-ready report for a STM is the same as for an ITM. To view example Status tab reports for ITMs, refer to ITM for status tab information.
- The Script tab allows you to specify a TSP-created user library and user module and allows you to specify a limited number of test parameters. To configure or reconfigure an STM, you need to configure the features of the Script tab, which is the default tab of the STM work area.

Open a STM work area

In the Test Setup panel select the STM to configure. For this example, the Resistor_ single STM was selected from the Test Setup panel.

Figure 187: Resistor single STM



If you inserted a completely new STM into the project plan that is not yet named or connected to a module, a blank STM Definition tab displays.

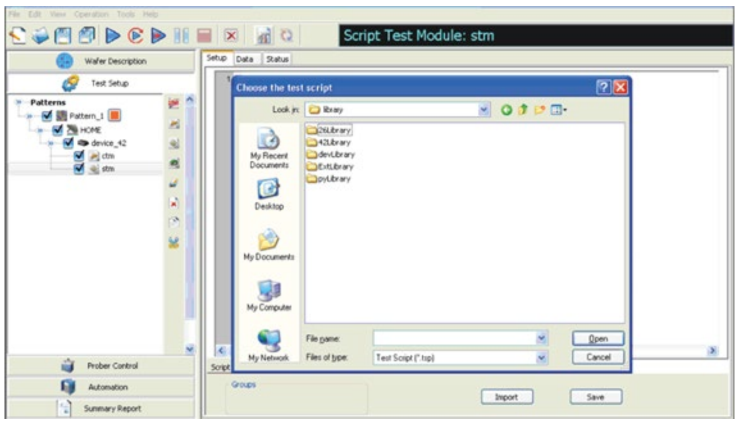
Make configuration changes

Make configuration changes in the Script tab for the blank STM.

To make configuration changes:

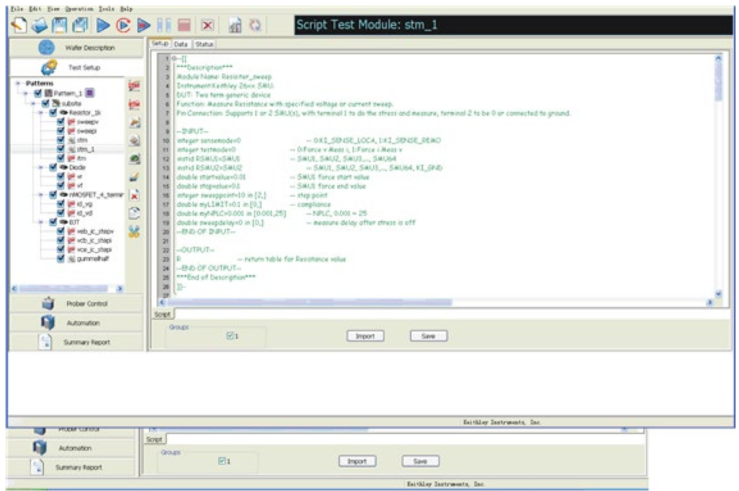
1. Right-click the STM and select the **Rename** command.
2. Type the name for this STM.
3. Select **OK**.
4. Select **Import** at the bottom of the STM work area to open a test script. The Choose the Test Script dialog displays, as shown in the following figure.

Figure 188: Import script



5. Choose `Resistor_sweep.tsp` in the Resistor folder.
6. Select **Open**.

Figure 189: Resistor sweep STM



NOTE

The imported file can be edited in Script Editor, which is a tool embedded in ACS. You can also write it in another tool, but it must follow the ACS STM library rules.

STM Script tab

The STM Script tab contains:

- Module Type and device information area.
- Group's information area.
- TSP Script area that displays the blank modules and allows you to import one into the user library or into your own library.

Understanding a STM script

A script is a collection of instrument control commands and programming statements. A script is made up of a chunk of programming code that is framed by shell commands. Program statements control script execution and provide services such as variables, functions, branching, and loop control. Because scripts are programs, they are written using a programming language. This language is called the Test Script Processing Language or TSP. TSP is derived from the Lua scripting language.

Script examples

Scripts can be written using the Test Script Builder (TSB). Scripts are loaded into the Series 2600B instrument and can be saved in nonvolatile memory. Scripts not saved in nonvolatile memory will be lost when the instrument is turned off.

The following example script is for a resistance measurement.

```
local  v_value = {}
local  i_value = {}
local  sensemode = 0
local  testmode = 0
local  RSMU1 = SMU1
local  RSMU2 = SMU2
local  forcevalue = 1
local  myLIMIT = 0.1
local  myNPLC = 1
local  testdelay = 0.1
local  resetflag = 1
local  Rvalue = {}

setmode(RSMU1, KI_INTGPLC, myNPLC)
   setmode(RSMU1, KI_SENSE, sensemode)

   if RSMU2 ~= KI_GND then
      setmode(RSMU2, KI_SENSE, sensemode)
      limiti(RSMU2, 1)
   end --if

   if testmode == 0 then
      limiti(RSMU1, myLIMIT)
      forcev(RSMU1, forcevalue)
   elseif testmode == 1 then
      limitv(RSMU1, myLIMIT)
      forcei(RSMU1, forcevalue)
   end --if

   if RSMU2 ~= KI_GND then
      forcev(RSMU2, 0)
   end --if

   delay(testdelay)
   intgv(RSMU1, v_value)
   intgi(RSMU1, i_value)
   Rvalue[1] = v_value[1]/i_value[1]

postdata("R_single", Rvalue[1])

if resetflag == 1 then
   devint()
end --if
```

Script using a function

TSP facilitates grouping commands and statements using the function keyword. Therefore, a script can also consist of one or more functions. Once a script has been run, the host computer can call a function in the script directly. TSP allows you to define functions. A function can take a predefined number of parameters and return multiple parameters.

To define a function and call that function, use the example below.

```
function add_two(parameter1, parameter2)
return(parameter1 + parameter2)
end
print(add_two(3, 4))
```

Below is a function that returns multiple parameters. It is used for resistance measurement.

```
function R_single(sensemode, testmode, RSMU1, RSMU2, forcevalue, myLIMIT, myNPLC,
    testdelay, resetflag, Rvalue)
   local v_value = {}
   local i_value = {}

setmode(RSMU1, KI_INTGPLC, myNPLC)
setmode(RSMU1, KI_SENSE, sensemode)
   if RSMU2 ~= KI_GND then
      setmode(RSMU2, KI_SENSE, sensemode)
      limiti(RSMU2, 1)
   end --if
   if testmode == 0 then
      limiti(RSMU1, myLIMIT)
      forcev(RSMU1, forcevalue)
   elseif testmode == 1 then
      limitv(RSMU1, myLIMIT)
      forcei(RSMU1, forcevalue)
   end --if
   if RSMU2 ~= KI_GND then
      forcev(RSMU2, 0)
   end --if
   delay(testdelay)
   intgv(RSMU1, v_value)
   intgi(RSMU1, i_value)
   Rvalue[1] = v_value[1]/i_value[1]
   postdata("R_single", Rvalue[1])
   if resetflag == 1 then
      devint()
   end --if
end --function
```

You can call a function using the following example:

```
Local Sensemode = 0
local testmode = 0
```

```
local RSMU1 = SMU1
local RSMU2 = SMU2
local forcevalue = 1
local myLIMIT = 0.1
local myNPLC = 1
local testdelay = 0.1
local resetflag = 1
local Rvalue = {}
R_single(sensemode, testmode, RSMU1, RSMU2, forcevalue, myLIMIT, myNPLC, testdelay,
   resetflag, Rvalue)
```

Script header

It is recommended that you add an output list header at the top of each STM, which is already functional in ACS. Make sure that this header is at the beginning of the script.

```
--[[
--OUTPUT--
I_drain
V_gate
--End of OUTPUT--
]]--
```

In the Data tab of a test module, in Test Setup, you will see with the variable sequence showing the same as in the output list. This output list is high priority compared to "postdata." If you add the output list, "postdata" will not be scanned and used.

If a parameter is not in the output list, but you still post it, it will be added in the data sheet as well.

The PLC test speed is set to 0.001 by default in ACS. If you want to change the test speed in the STM, you should change the test speed in the PLC at the beginning of the module.

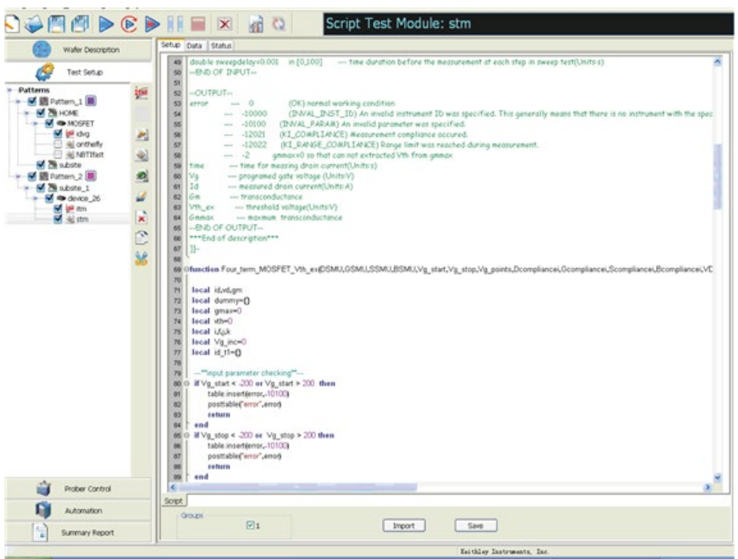
Writing or modifying a script

A script is a list of TSP commands . A script file is created using Test Script Builder (TSB). TSB is a Keithley Instruments product that can be downloaded on the tek.com/keithley.

To modify a script file in the STM work area:

1. Select the script file name in the Test Setup panel of ACS. The script file opens in the STM work area and contains tabs depending on the contents of the script file, as shown in the following figure.

Figure 190: Importing a Script



2. Modify the script file, as necessary.
3. Select **Save** at the bottom of Script tab in the STM work area.

You can also access the Test Script Builder by selecting Edit. In the Test Script Builder, the function is at the bottom of the STM work area.

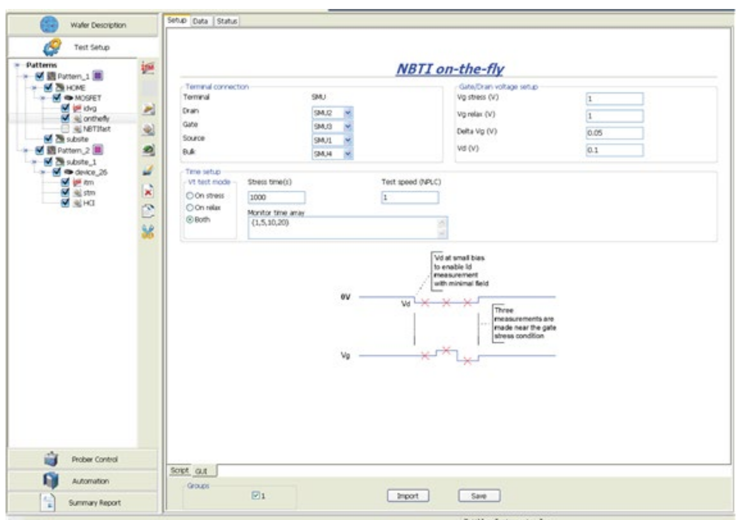
STM with the .xrc GUI File

If the TSP file imported has a corresponding `.xrc` user interface file, ACS automatically loads and displays the user interface. You can set parameters on the user interface so that you do not need to edit the parameters (see the next several figures). You can generate the `.xrc` file yourself.

NOTE

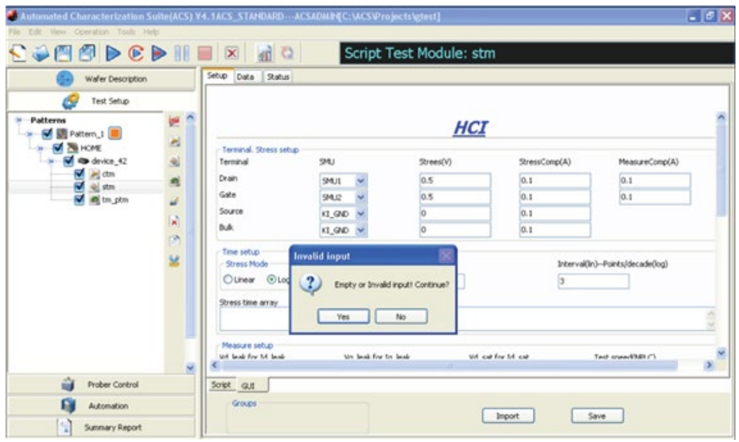
It is recommended that only people who are familiar with XRCed software try to generate their own `.xrc` files. This process will require extensive knowledge and understanding to accomplish.

Figure 191: TSP GUI example HCI



If you input empty or invalid parameters in the user interface tab, you will receive a message indicating that the input is not correct, as shown in the following figure.

Figure 192: Empty or Invalid input



After the correct input is entered, open the Script tab, as shown in the following figure.

Figure 193: STM Script tab

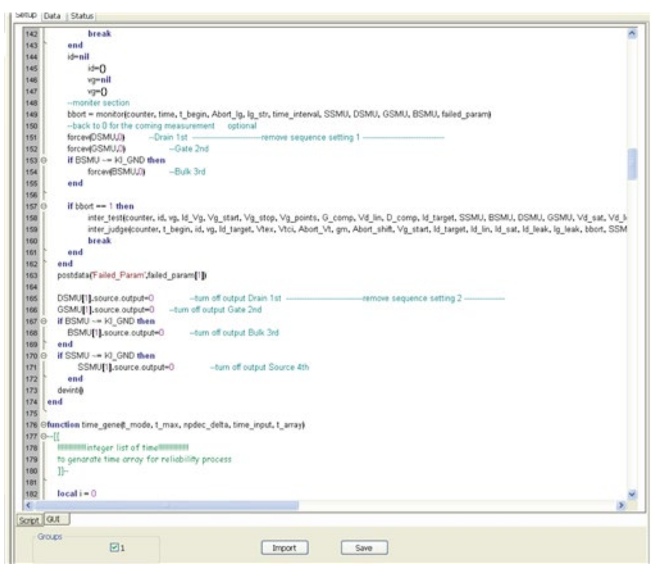
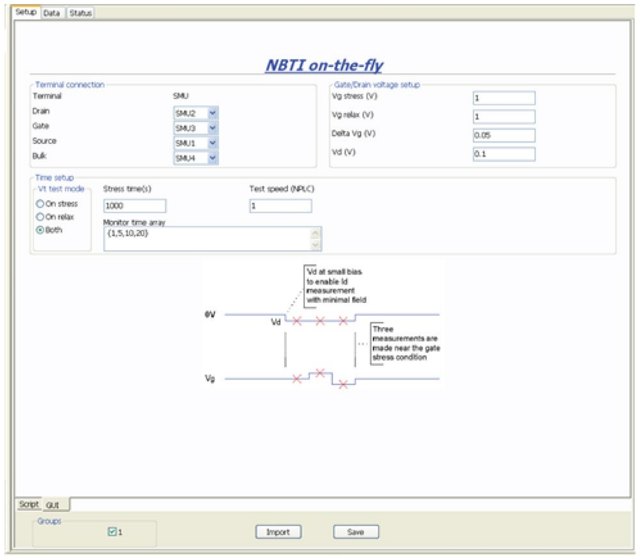


Figure 194: TSP GUI example NBTI



The user interface (`.xrc` file) editor is accessed by selecting Tools > Custom Test GUI Designer(XRC).

Create a TSP GUI file using XRC

NOTE

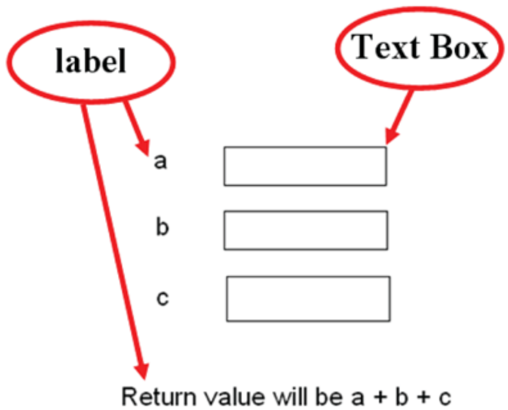
Keithley Instruments recommends that you use the Script Editor Tool in the ACS software package rather than the XRC software user interface creation tool described in this topic. See Script Editor tool (on page 5-197) and Script Editor tutorials (on page 5-213) for more information.

If you are familiar with XRCed software, including wxWindows class names and the XRC format, you can use the following information to assist in creating a TSP user interface file. This process requires extensive knowledge and understanding to accomplish. Also, note that the following information is not a step-by-step process on how to create a TSP user interface.

If you do not have experience or knowledge with XRCed software, it is strongly recommended that you use Script Editor.

Before creating a TSP user interface, create a draft first so that you know how you want the user interface to look, depending on the functionality that is needed for your testing. First, we will show you an image of a simple user interface and show you how to create a TSP user interface based on a few simple parameters. For example, the following figure shows a simple user interface with three labels and three text boxes, as shown in the following figure.

Figure 195: Simple GUI image

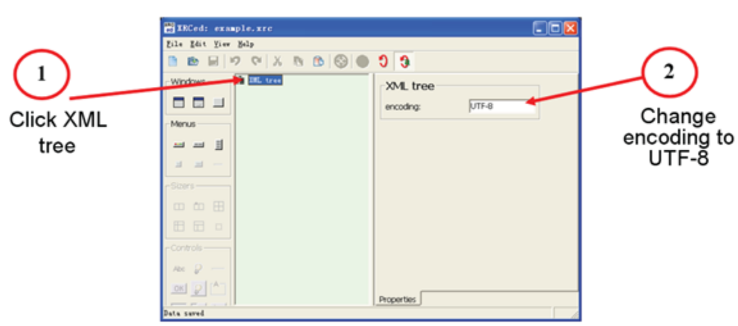


The purpose of this simple user interface is to take three values (a, b, and c), add them together, and return a value. For example, the output will equal a + b + c.

To create a TSP GUI file:

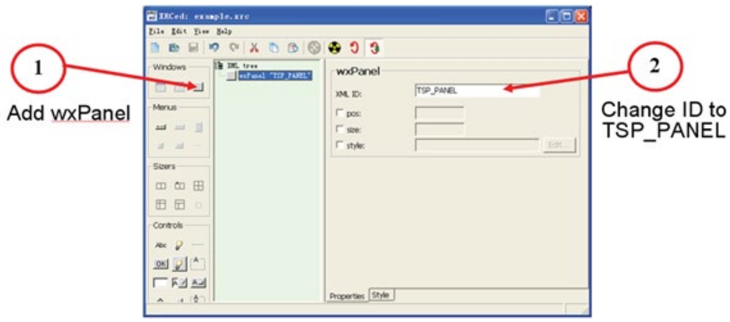
1. Go to **Tools**, then select **Custom Test GUI Designer (XRC)**.
2. From the **File** menu, select **New**.
3. Change the XML configuration navigator encoding, as shown in the following figure.

Figure 196: Change the .xml tree encoding



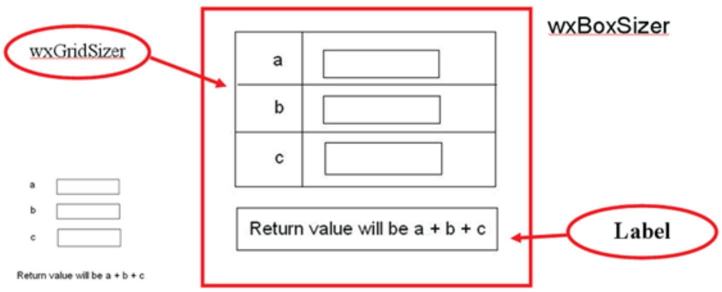
4. Add a panel to the XML configuration navigator (only wxPanel is supported in ACS), as shown in the following figure.

Figure 197: Add panel for new object



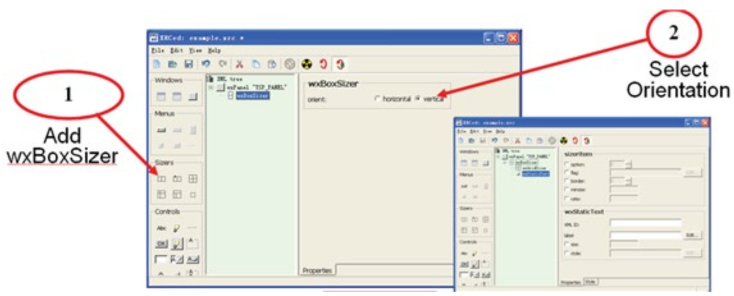
5. Change the panel ID to TSP_PANEL (only this ID script is identified in ACS).
6. In the following figure for the design user interface structure, notice the use of the GridSizer and BoxSizer.

Figure 198: Design GUI structure



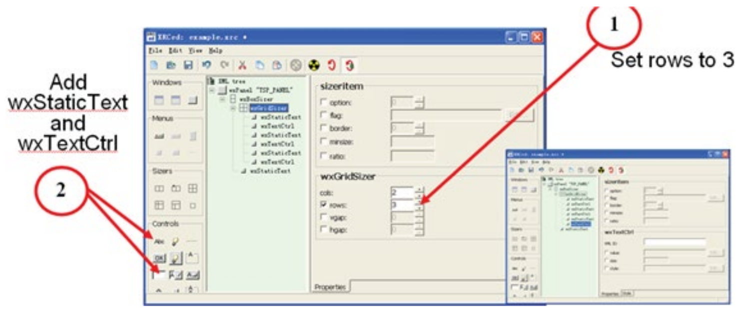
7. Select the BoxSizer size for your project user interface and choose either vertical or horizontal, as shown in the following figure.

Figure 199: Add sizers under the TSP panel



8. Select the GridSizer size, add your StaticText (the label) and the TextCtrl (the text box) to your project user interface, as shown in the following figure.

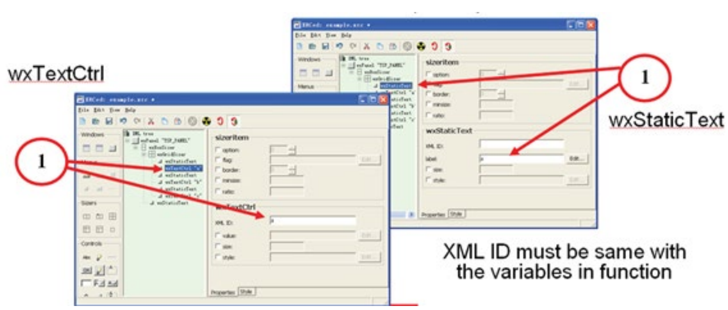
Figure 200: Add labels and text box under wxGridSizer



Modify the information in the StaticText (label) and the TextCtrl (text box):

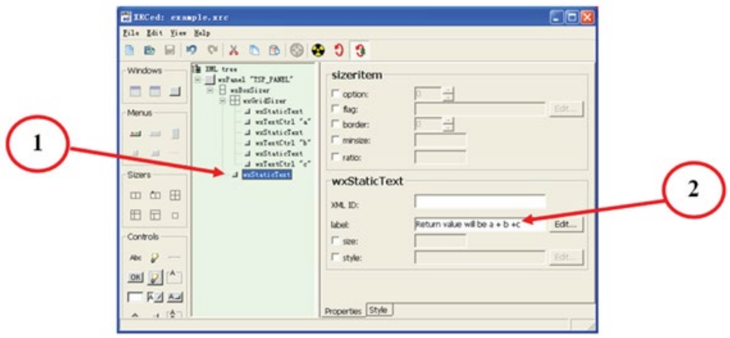
1. Select the StaticText node on the XML configuration navigator.
2. Set the XML ID and label on the right.
3. Select TextCtrl, as shown in the following figure.

Figure 201: Modify information



4. Set the label for the StaticText (return information label, see following figure).
5. Select wxStaticText.
6. Save your `.xrc` file.

Figure 202: Set label for wxStatic Text



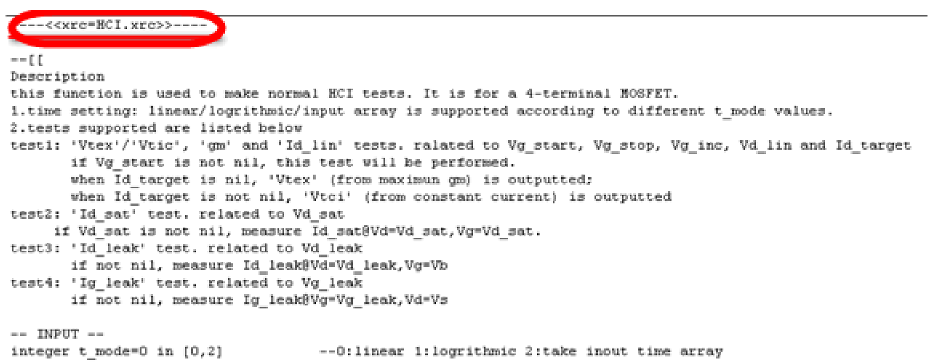
Once you have completed these steps, you can save this as a `.xrc` user interface that can be used in the Script Editor Tool. For more information about how to use the Script Editor and create a STM with the `.xrc` user interface, see Script Editor tool (on page 5-197) and Script Editor tutorials (on page 5-213) for more information.

Helpful tips

Helpful tips:

- Save your TSP script file to this folder: `\\ACS\library\26Library\TSPLib`.
- Save your `.xrc` user interface file to this location: `\\ACS\library\26Library\TSPLib\xrc`.
- Add a specification line with a recognizable format and name to the top of the TSP script to help identify it with the corresponding `.xrc` file. The ACS parsing engine will parse this format and determine the `.xrc` file that is related to the TSP script and will load the `.xrc` file. In other words, the TSP text and TSP user interface are linked by the `.xrc` file name that appears at the top of the text file. The suggested format and name is `----<<xrc=HCI.xrc>>----`, as shown in the following figure.

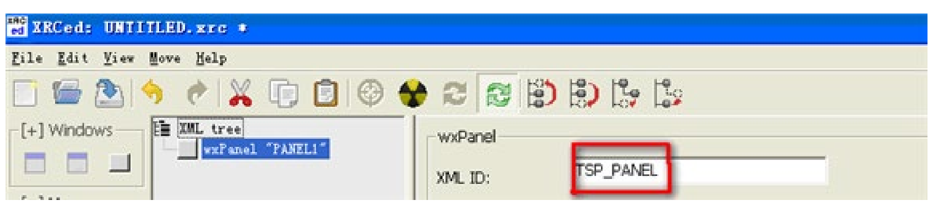
Figure 203: XRC specification line name



```
----<<xrc=HCI.xrc>>----

--[[
Description
this function is used to make normal HCI tests. It is for a 4-terminal MOSFET.
1.time setting: linear/logrithmic/input array is supported according to different t_mode values.
2.tests supported are listed below
test1: 'Vtex'/'Vtic', 'gm' and 'Id_lin' tests. ralated to Vg_start, Vg_stop, Vg_inc, Vd_lin and Id_target
        if Vg_start is not nil, this test will be performed.
        when Id_target is nil, 'Vtex' (from maximun gm) is outputted;
        when Id_target is not nil, 'Vtci' (from constant current) is outputted
test2: 'Id_sat' test. related to Vd_sat
      if Vd_sat is not nil, measure Id_sat@Vd=Vd_sat,Vg=Vd_sat.
test3: 'Id_leak' test. related to Vd_leak
        if not nil, measure Id_leak@Vd=Vd_leak,Vg=Vb
test4: 'Ig_leak' test. related to Vg_leak
        if not nil, measure Ig_leak@Vg=Vg_leak,Vd=Vs

-- INPUT --
integer t_mode=0 in [0,2]          --0:linear 1:logrithmic 2:take inout time array
```

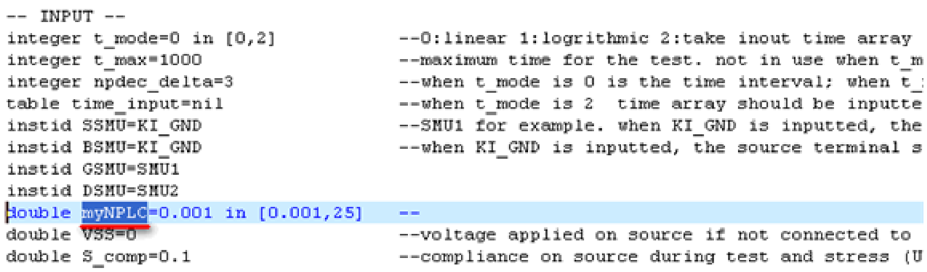
- The Name of the wxPanel must be called `TSP_PANEL`, as shown in the following figure.

Figure 204: wxPanel named TSP_PANEL



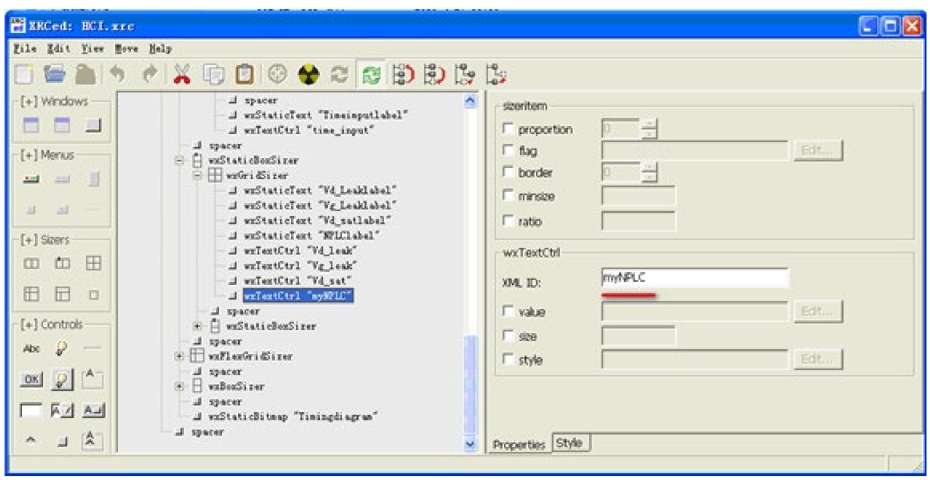
- The input XML ID parameter in the test module must have a corresponding control in the XRC user interface file so that ACS will use all the correct input values that correspond to the input parameters. Therefore, the controls in the `.xrc` file must have the same name as the input (see next two figures)

Figure 205: XML ID of the wxTextCtrl



```
-- INPUT --
integer t_mode=0 in [0,2]            --0:linear 1:logrithmic 2:take inout time array
integer t_max=1000                   --maximum time for the test. not in use when t_m
integer npdec_delta=3                --when t_mode is 0 is the time interval; when t_
table time_input=nil                 --when t_mode is 2  time array should be inputte
instid SSMU=KI_GND                   --SMU1 for example. when KI_GND is inputted, the
instid BSMU=KI_GND                   --when KI_GND is inputted, the source terminal s
instid GSMU=SMU1
instid DSMU=SMU2
double myNPLC=0.001 in [0.001,25]    --
double VSS=0                         --voltage applied on source if not connected to
double S_comp=0.1                    --compliance on source during test and stress (U
```

Figure 206: wxTextCtrl and XML ID



- You must make sure that your script contains a session code at the end of the script that tells the engine to execute a function with the given input parameters that is named `--CALL--`, as shown in the following figure.

Figure 207: Session named CALL

```
        return
      end
      r[1] = top/bot
end

--------------------------------------------------

--CALL--

---- the following is auto-generated by KATS ----

local SMU_id=SMU1
local Isrc=1e-3
local Delay_time=400
local myPLC=1
local CompV=20
local Temp_env=30
local Temp_start=30
local Temp_stop=180
local Temp_point=6
local Temp_ref=20
```

- Make sure that you place your input parameters between the key words `--INPUT--` and `--End of INPUT--`. ACS uses this to parse the input parameters, as shown in the following figure.

Figure 208: INPUT and End of INPUT parameters

```
--[[
this function is used to determine TCR at Reference Temperature for selecte
Resistance is measured at several temperature. LLSQ is used to fit the rela

--INPUT--
SMU_id=SMU1             ---4-wire structure is used here. only one SMU per test
Isrc=1e-3               ---current value used to measure resistor. this voltage
Delay_time=400          ---time for chuck to settle at each temperature. measur
myPLC=1                 ---PLC value for the measurement
CompV=0.1               ---compliance voltage
Temp_env=25             ----enviroment temperature
Temp_start=25           ---the first temperature
Temp_stop=180           ---the last temperature
Temp_point=5            ---total point of temperature need measurement
Temp_ref=20             ---reference temperature at which TCR is calculated
--End of INPUT--
--OUTPUT--
Temp                    ---array of temperature
R                       ---array of resistance
slope                   ---slope of the fitting line of R vs. Temp
R0                      ---resistance at 0 degree C
```

- Make sure that you place your output parameters between the key words `--OUTPUT--` and `--End of OUTPUT--`. ACS uses this to parse the output parameters, as shown in the following figure.

Figure 209: OUTPUT and End of OUTPUT parameters

```
--INPUT--
SMU_id=SMU1          ---4-wire structure is used here. only one SMU per test
Isrc=1e-3            ---current value used to measure resistor. this voltage
Delay_time=400       ---time for chuck to settle at each temperature. measur
myPLC=1              ---PLC value for the measurement
CompV=0.1            ---compliance voltage
Temp_env=25          ----enviroment temperature
Temp_start=25        ---the first temperature
Temp_stop=180        ---the last temperature
Temp_point=5         ---total point of temperature need measurement
Temp_ref=20          ---reference temperature at which TCR is calculated
--End of INPUT--
--OUTPUT--
Temp                 ---array of temperature
R                    ---array of resistance
slope                ---slope of the fitting line of R vs. Temp
R0                   ---resistance at 0 degree C
TCR_ref              ---calculated TCR at reference temperature
R_env                ---resistance value at enviroment temperature
R_ref                ---resistance value at reference temperature
--END OF OUTPUT--
]]--
```

You must make sure that your TSP script contains a session code so that it executes the selected function. For example, every TSP script should have a main function. The following figure indicates the main function (`TCR_EM` is the main function) of this script.

Figure 210: Session code for executing scripts

```
--CALL--

---- the following is auto-generated by KATS ----

local SMU_id=SMU1
local Isrc=1e-3
local Delay_time=400
local myPLC=1
local CompV=20
local Temp_env=30
local Temp_start=30
local Temp_stop=180
local Temp_point=6
local Temp_ref=20

TCR_EM(SMU_id, Isrc, Delay_time, myPLC, CompV, Temp_env, Temp_start, Temp_stop, Temp_point, Temp_ref)
```

Modify script for GUI import

After the script with the `.xrc` user interface is imported to ACS, the corresponding information of the user interface and Script can be changed. If you want to modify information, use the user interface script that you imported.

In the script, there must be a `.xrc` file name, input variable, function, and function call so that you can modify various parts to meet your requirements (see next two figures).

Figure 211: Modify script for import GUI

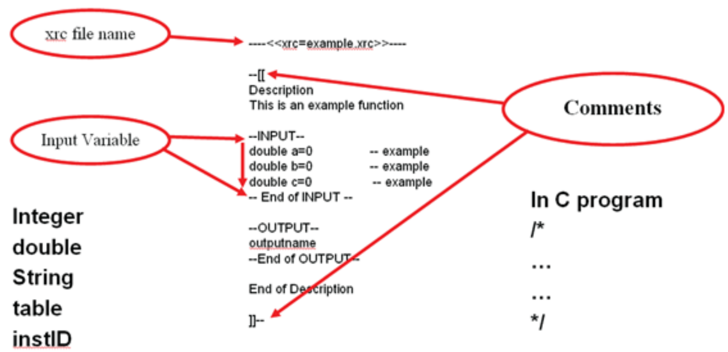
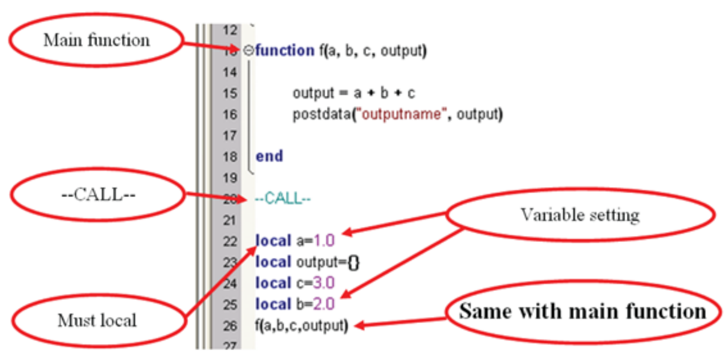


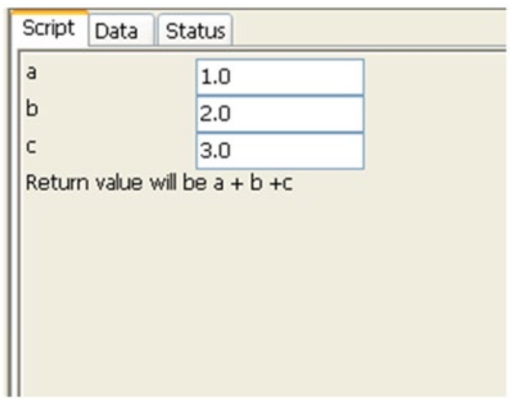
Figure 212: Modify script for import GUI (2)



In the test script program, the comments are denoted as `[[ ......]]`. In the C language, program comments are denoted as `/* ......*/`. Input variables can be integer, double, string, table or instID. Additionally, in the `-CALL-` part, the call function name must be the same as the name of the main functions. The following of `--CALL --` is automatically generated by ACS.

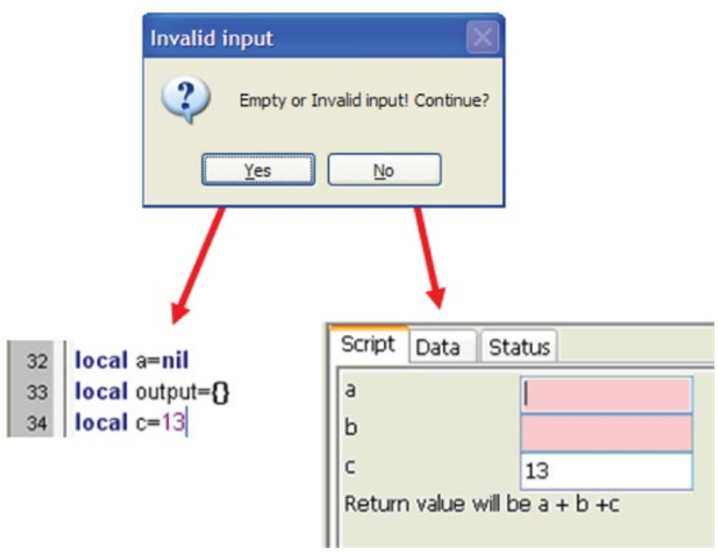
You can set the limit range for input variables. For example: double a=0.0 in [-40,40]; double b=0.0 in [-40,]; integer a=0 in [,40], as shown in the following figure.

Figure 213: TSP GUI for STM



On the TSP user interface, you can input values according to what you want. If input is empty, you receive an invalid input message, as shown in the following figure.

Figure 214: Invalid input



C language Test Module (CTM) configuration

A CTM is a test module defined primarily by programming a C-language user module using KULT of 4200A. It is one in which you typically configure key test parameters with the user interface.

After inserting a CTM into a project, it must be configured to meet the test requirements. This section describes the connection of CTMs to user modules, user libraries, and how to enter or edit module parameters.

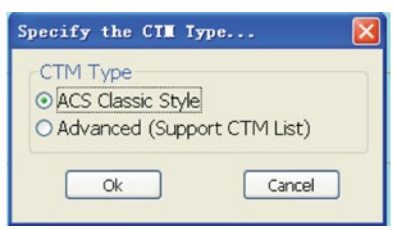
Connect or reconnect the selected CTM to an existing KULT-created user module using the CTM Setup tab of the CTM work area.

There are two kinds of CTM: ACS Classic Style and Advanced Style (Support CTM List). Classic Style CTM only contains one test module, but the advanced style CTM contains many test modules.

Classic CTM configuration

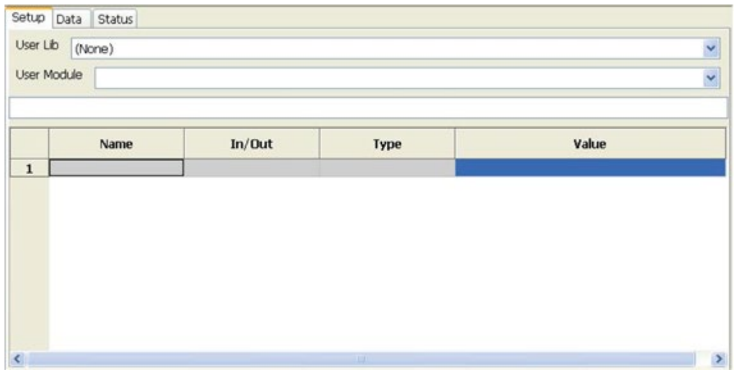
1. Select the New CTM icon in the vertical edit toolbar. The Specify the CTM Type dialog opens, as shown in the following figure.

Figure 215: Specify the CTM Type



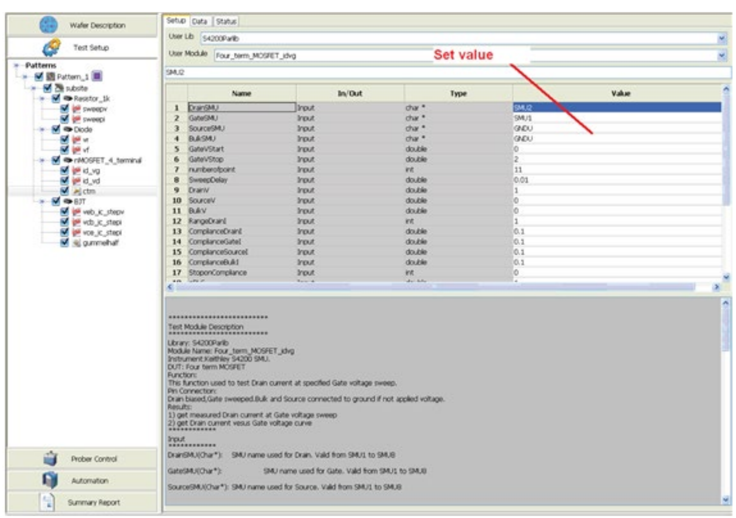
2. Select the **ACS Classic Style** and select **OK**. The CTM Setup Tab displays, as shown in the following figure.

Figure 216: CTM Setup tab



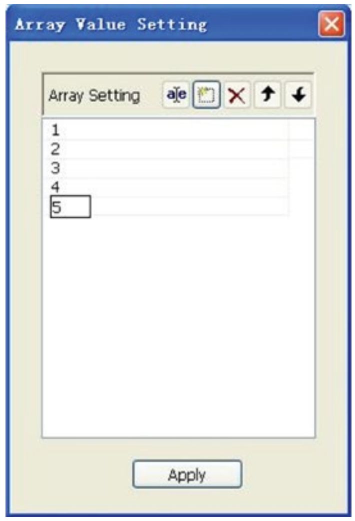
3. In the User Lib menu of the Setup tab, select the user library that contains the user module to test. For this example, the ACS_DEMO library is selected.
4. In the User Module menu of the Setup tab, select the user module to test. The CTM displays the configuration parameters for the selected user module. In the following figure, idvd_demo is selected as the user module.

Figure 217: CTM Setup example



5. Accept the default parameter values or change them if needed. For an array input, select Set Value on the active sheet. The Array Value Setting dialog opens, as shown in the following figure.

Figure 218: CTM Array Value Setting



6. Choose the Array, then select **Apply**. This returns you to the Setup tab.
7. Save the CTM and the project by selecting the Save or Save All function on the toolbar.
8. Run the CTM.
9. View the result in the Data tab.

CTM work area

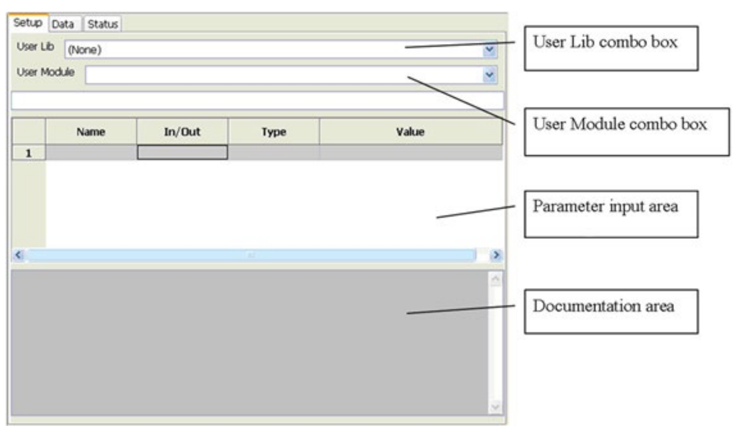
A CTM work area allows you to enter information that defines a given CTM. A CTM work area also allows you to view and analyze test data created by the CTM, both numerically and graphically. Tabs available in the CTM work area are discussed in the following.

The Setup tab of the CTM work area allows you to specify a KULT-created user library and user module. In addition, it allows you to specify a limited number of test parameters.

The Data tab displays the CTM results and graph in its Data worksheet. The Data worksheet allows you to create and plot graphs of the test results. Refer to Data plot (on page 5-233) for more information.

The Status tab monitors the current configuration status of the CTM, as shown in the following figure.

Figure 219: CTM work area



- **User Lib menu:** Displays the available user libraries for selection.
- **User Module menu:** Displays the user modules in the user library for selection.
- **Parameter input area:** Contains the following:
  - **Parameter cells:** Spreadsheet-like cells that receive inputs directly or display parameter titles. As in a standard spreadsheet, each cell is designated by a column and row (for example, the fourth column, second row is designated as D2). Any of the white cells next to parameter titles and descriptions are valid user-input cells.
  - **Present-cell list box:** Displays the contents of the presently selected cell and provides an alternative, more spacious place to input or change the parameters of the currently selected user-input cell.
  - **Documentation area:** Typically displays descriptions, examples, and requirements for the connected user module.

NOTE

The CTM LPT commands only work when the ACS software is installed on the 4200A to control the 4200A hardware, such as the SMU, CVU, SCP, or VPU. Also, if you are controlling the 4200A hardware using the KXCI software and an ethernet cable, you cannot use the KITE LPT commands at the same time. For more information about CTMs that are used on the 4200A, refer to the documentation in the ACS software Help menu.

Post data function in CTM

In KITE UTM, the data post functions are PostDataInt (char *, int *), PostDataDouble (char *, double *) and PostDataString (char *, char*). But in ACS CTM, those functions can no longer be used if the KITE version is v7.1 or older. However, if the KITE version is v7.2 or newer, all the KITE UTM data post functions can be used in ACS, and the KITE user library can be used in ACS directly without any modifications. Note that the following functions have been added.

- int ACSPostDataInt (char *data_name, int data_val);
- int ACSPostDataFloat (char *data_name, float data_val);
- int ACSPostDataDouble (char *data_name, double data_val);
- int ACSPostArrayInt (char *data_name, int *data_array, int data_num);
- int ACSPostArrayFloat (char *data_name, float *data_array, int data_num);
- int ACSPostArrayDouble (char *data_name, double *data_array, int data_num);

NOTE

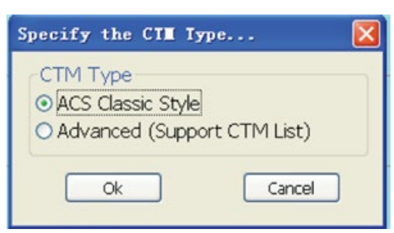
To use these functions in ACS, you should add `#include acsenv.h` in the C source code.

Advanced CTM configuration

To configure the CTM:

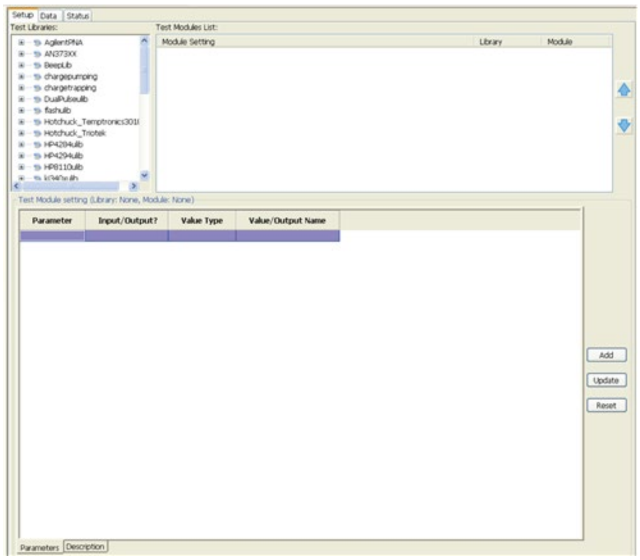
1. Select the New CTM icon. The Specify the CTM Type dialog opens, as shown in the following figure.

Figure 220: Specify the CTM Type



2. Select the Advanced (Support CTM List) and select **OK**. The CTM Setup Tab of this Advanced CTM is displayed by default, as shown in the following figure.

Figure 221: CTM of Advanced Setup tab

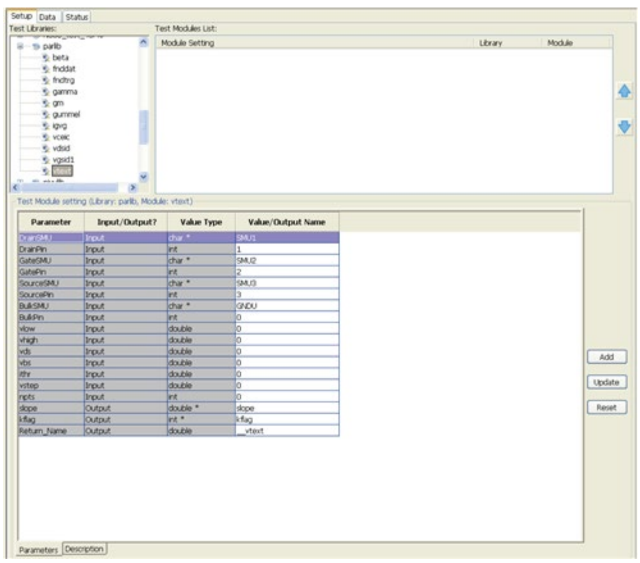


The main parts of the Setup tab are:

- Test libraries
- Test module list
- Test module setting

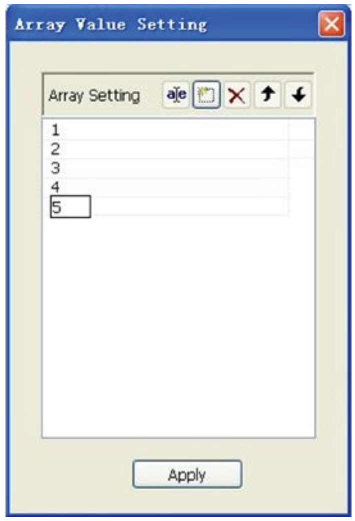
3. In the Test Libraries box, select the user library that contains the user modules. For this example, the parlib library is selected.
4. After expanding the library, you see all the test modules in the library. Select the module to test. For example, in the Vtext module you see the configuration parameters for this CTM, as shown in the following figure.

Figure 222: CTM Setup tab example



5. For an array input, select the active sheet. The Array Value Setting dialog opens, as shown in the following figure.

Figure 223: CTM Array Setting



6. You can select the Description tab. You see descriptions, examples, and requirements for this user module (see the following two figures).

Figure 224: Description function

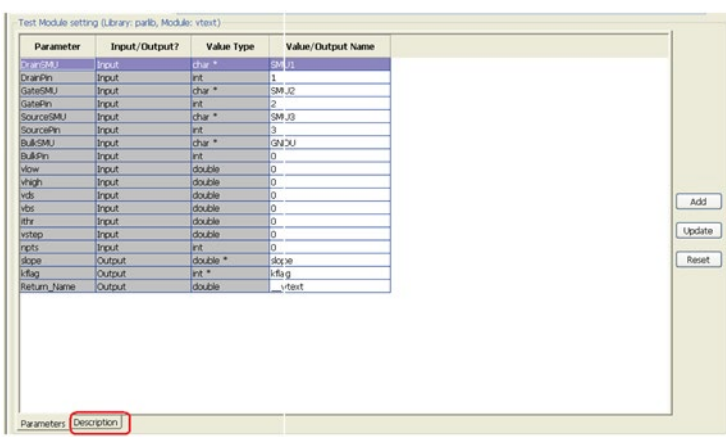
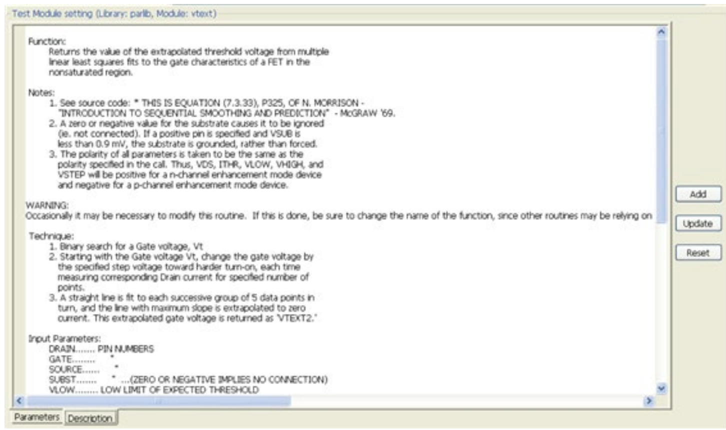
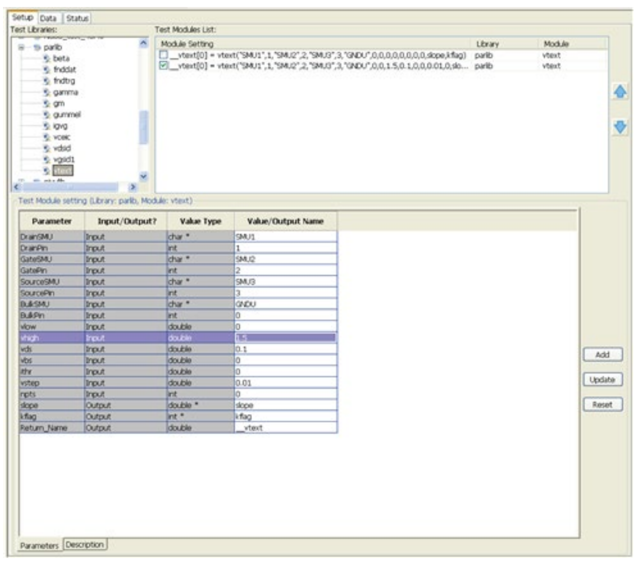


Figure 225: Description of CTM



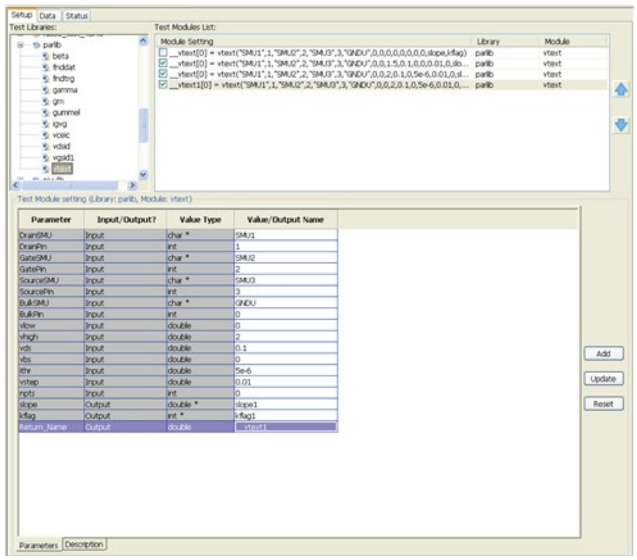
7. Select **Add** to add this module to the Test Modules List, as shown in the following figure.

Figure 226: Add a test module to the Test Modules List



8. In the Test Module List, you can see the following figure _vtext[0] in the Test Module List. Here you can change the parameters as shown in the step 5, and select **Add** again, which means you can add this module with different parameters for each module, as shown in the following figure.

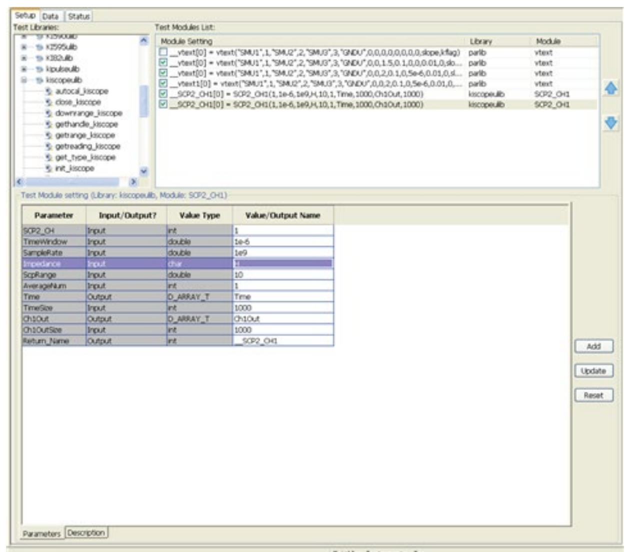
Figure 227: Add a test module twice to the Test Modules List



- Select **Reset** to set all the parameters of the module in the Test Module setting to their original values.
- If you want to change the parameters of a module in the Test Module List, you can select this module in the Test Module List, then this module appears in the Test Module Setting. You can change the parameters as you want and select **Update**.

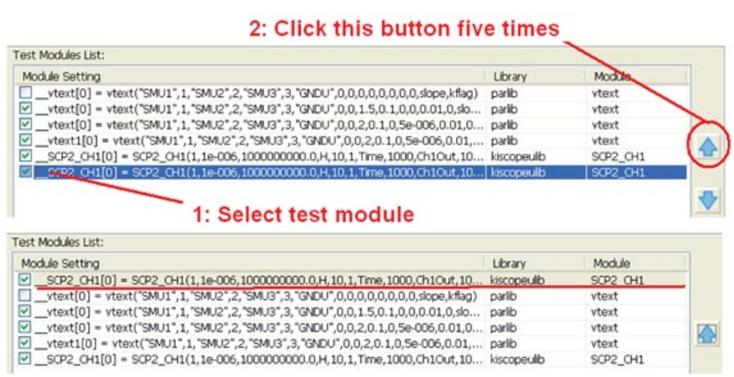
9. You can also add modules to other libraries for Test Modules. For example, select the kiscopeulib library and double-click the SCP2_CH1 module and select. This new module is added to the Test Module List, as shown in the following figure.

Figure 228: Add another test module to the list



10. You can change the module's test sequence in the Test Modules List with the upper function and forward function. For example, if you want to test the _SCP2_CH1[0] module first, select this module and select the up arrow.

Figure 229: Change the module test sequence



11. In the Test Modules List, right-click and there are four items: **Select All**, **Unselect All**, **Delete Selected Macro**, and **Clear All**. Select **Delete Selected Macro** to delete the selected the module and **Clear All** to delete all the modules in the Test Module List.
12. Save the CTM and the project by selecting the Save or Save All function on the toolbar.
13. Run the CTM.
14. View the result in the Data tab.

The Data tab displays the CTM results and graph in its Data worksheet. The Data worksheet allows you to create and plot graphs of the test results. Refer to Data Plot (on page 5-233).

The Status tab shows the current configuration status of the CTM.

Python Language Test Module (PTM) configuration

A PTM is a test module defined primarily by programming a Python language user module. It can be created by using the Script Editor in ACS software.

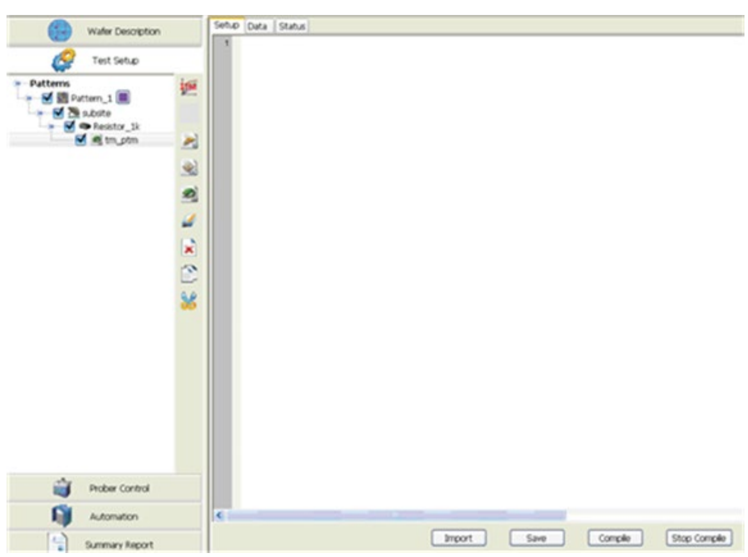
The PTM can be used as a test module. PTMs can also be used to control instruments and process data.

After inserting a PTM into a project configuration navigator, it must be configured to meet the test requirements.

To configure a PTM:

1. Select the New PTM icon. The Setup tab is displayed, as shown in the following figure.

Figure 230: Add a PTM

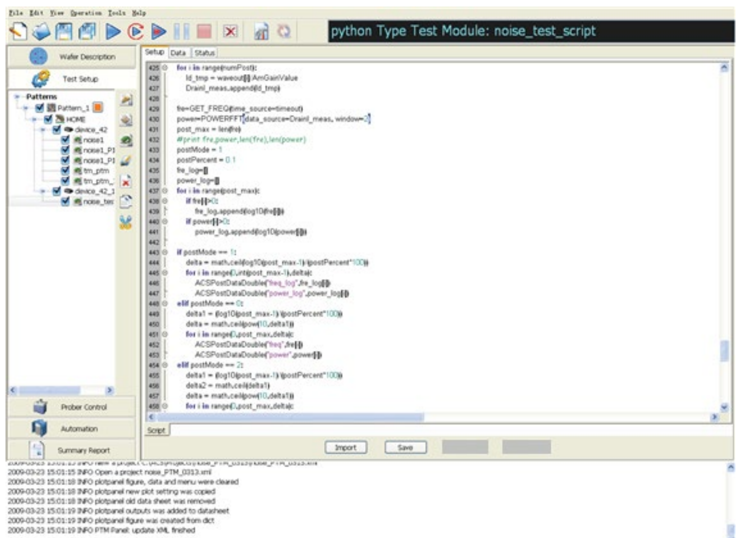


2. You can write a PTM script or import an existing test module.

NOTE

The imported file can be edited in Script Editor, a tool embedded in ACS, or you can write the code yourself. The code will have to follow the ACS STM library rules. For more information about the Script Editor, you can refer to the Script Editor tutorials (on page 5-213).

3. Select **Import**.
4. Select **Save**.



The Data tab displays the PTM results and the graph in the Data worksheet. The Data worksheet allows you to create and plot graphs of the test results. Refer to the Data Plot of the Status tab where it monitors the current configuration status of the PTM.

Understanding a Python script

A Python script is a script written in Python® programming language. A script is a collection of instrument control commands and programming statements. Program statements control script execution and provide services such as variables, functions, branching, and loop control.

Script example

The example Python script below makes a diode measurement.

```
from ACSLPT import *
from ptmlpt.constantlpt import *
from ACS_PostData import *
import kimisc as misc
from ACS_Global import *

def Diode_DynamicZ_I1I2(PSMU='SMU1', NSMU='GNDU', I1=0.01, I2=0.02, RangeV=1,
    ComplianceV=20, nPLC=1.0, Holdtime=0.01, output_error='error',
    output_DynamicZ='DynamicZ', output_I1='I1', output_I2='I2', output_V1='V1',
    output_V2='V2'):
   """
   VISIBILITY: NOT_HIDDEN
   TYPE : standard
   INPUTLIST
      #Name, Type, Default Value, Min Value, Max Value, Unit
PSMU,enum,'SMU1',('SMU1','SMU2','SMU3','SMU4','SMU5','SMU6','SMU7','SMU8'),,A
NSMU,enum,'GNDU',('SMU1','SMU2','SMU3','SMU4','SMU5','SMU6','SMU7','SMU8','GNDU'),,
    A
      I1,float,0.01,-0.1,0.1,A
      I2,float,0.02,-0.1,0.1,A
      RangeV,int,1,1,5,A
      ComplianceV,float,20,-1100,1100,A
      nPLC,float,1.0,0.01,10,A
      Holdtime,float,0.01,0,100,A
END INPUTLIST
OUTPUTLIST
      #Name, Type, DefaultValue, Unit
      output_error,str,'error',A
      output_DynamicZ,str,'DynamicZ',A
      output_I1,str,'I1',A
      output_I2,str,'I2',A
      output_V1,str,'V1',A
      output_V2,str,'V2',A
END OUTPUTLIST
DESCRIPTION
      Module Name: Diode_DynamicZ_I1I2
      Instrument:Keithley S4200 SMU.
      DUT: Diode
      Function: Calculates the Dynamic Impedance based on two forward voltage or
    two reverse voltage measurements.
      DynamicZ = (v2 - v1) / (I2 - I1)
      Pin Connection: It uses one SMU to force forward current, while the other
    terminal is grounded.
      ---PSMU:SMU name of P terminal. Valid from SMU1 to SMU8.
      ---NSMU:SMU name of N terminal. Valid from SMU1 to SMU8, GNDU.
      ---I1:P terminal forced current. Valid from -0.1 to 0.1. [A]
      ---I2:P terminal forced current. Valid from -0.1 to 0.1. [A]
      ---RangeV: Limit autorange of voltage measurement. Valid from 1 to 5. For
    some instrument without lorange, it will be set to autorange.
            Input           Range
```

```
         ======================
         1                Full Auto
         2                Limited Auto 0.2V
         3                Limited Auto 2V
         4                Limited Auto 20V
         5                Limited Auto 200V
   ---ComplianceV:Compliance of current force.
   ---Valid from -1100 to 1100. [V]
   ---nPLC:Number of power line cycles for integration.
   ---Valid input from 0.01 to 10.
   ---Holdtime:Holdtime before measurement.
   ---Valid input from 0 to 100.  [s]
   ---output_error : error value list
0 (OK) normal working condition;
-10000 (INVAL_INST_ID)An invalid instrument ID was specified. This generally
means that there is no instrument with the specified ID in your configuration.
-10100 (INVAL_PARAM) An invalid parameter was specified.
-12021 (KI_COMPLIANCE) Measurement compliance occurred.
-12022 (KI_RANGE_COMPLIANCE) Range limit was reached during measurement.
   ---output_DynamicZ : Dynamic impendence of diode.
   ---output_I1 :
   ---output_I2 :
   ---output_V1 :
   ---output_V2 :
END DESCRIPTION
"""

Get4200HWCtrl()

error = [ ]
DynamicZ = 0.0
#Some input checking is needed
if  abs(I1) > 0.1:
   error.append(INVAL_PARAM)
   ACSPostArrayInt(output_error, error, len(error))
   return INVAL_PARAM
if  abs(I2) > 0.1:
   error.append(INVAL_PARAM)
   ACSPostArrayInt(output_error, error, len(error))
   return INVAL_PARAM
if  RangeV < 1 or RangeV > 5:
   error.append(INVAL_PARAM)
   ACSPostArrayInt(output_error, error, len(error))
   return INVAL_PARAM
if  abs(ComplianceV) > 1100:
   error.append(INVAL_PARAM)
   ACSPostArrayInt(output_error, error, len(error))
   return INVAL_PARAM
if  nPLC < 0.01 or nPLC > 10:
   error.append(INVAL_PARAM)
   ACSPostArrayInt(output_error, error, len(error))
   return INVAL_PARAM
if  Holdtime < 0 or Holdtime > 100:
   error.append(INVAL_PARAM)
   ACSPostArrayInt(output_error, error, len(error))
   return INVAL_PARAM
```

```
#Map the SMUs with DUT terminals
PSMUId = getinstid(PSMU)
if not PSMUId:
   error.append(INVAL_INST_ID)
   ACSPostArrayInt(output_error, error, len(error))
   return INVAL_INST_ID
NSMUId = getinstid(NSMU)
if not NSMUId:
   error.append(INVAL_INST_ID)
   ACSPostArrayInt(output_error, error, len(error))
   return INVAL_INST_ID
#Set range and compliance
limitv(PSMU, ComplianceV)
if NSMU != "GNDU":
   limitv(NSMU, ComplianceV)
if RangeV == 1:
   setauto(PSMU)
elif RangeV == 2:
   lorangev(PSMU, 0.2)
elif RangeV == 3:
   lorangev(PSMU, 2)
elif RangeV == 4:
   lorangev(PSMU, 20)
elif RangeV == 5:
   lorangev(PSMU, 200)
else:
   lorangev(PSMU, 2)
#Set instrument config
setmode(PSMU, KI_INTGPLC, nPLC)
htime = int(Holdtime * 1000)
#Set stress
if NSMU != "GNDU":
   forcev(NSMU, 0)
forcei(PSMU, I1)
delay(htime)
V1 = intgv(PSMU)
forcei(PSMU, I2)
delay(htime)
V2 = intgv(PSMU)
#check compliance
cstatus = getstatus(PSMU, KI_COMPLNC)
if cstatus == 2:
   error.append(KI_RANGE_COMPLIANCE)
   ACSPostArrayInt(output_error, error, len(error))
   return error[len(error)-1]
if cstatus == 4:
   error.append(KI_COMPLIANCE)
   ACSPostArrayInt(output_error, error, len(error))
   return error[len(error)-1]
#test complete
devint()
Idelta = I1 - I2
if Idelta == 0:
   Idelta = 1e-37
DynamicZ = (V1 -V2) / Idelta
```

```
    error.append(0)
    ACSPostDataDouble(output_DynamicZ, DynamicZ)
    ACSPostDataDouble(output_I1, I1)
    ACSPostDataDouble(output_I2, I2)
    ACSPostDataDouble(output_V1, V1)
    ACSPostDataDouble(output_V2, V2)
    ACSPostArrayInt(output_error, error, len(error))
    return  DynamicZ

#--CALL--

PSMU = "SMU1"
NSMU = "GNDU"
I1 = 0.01
I2 = 0.02
RangeV = 1
ComplianceV = 20
nPLC = 1.0
Holdtime = 0.01
output_error = "error"
output_DynamicZ = "DynamicZ"
output_I1 = "I1"
output_I2 = "I2"
output_V1 = "V1"
output_V2 = "V2"

Diode_DynamicZ_I1I2(PSMU,NSMU,I1,I2,RangeV,ComplianceV,nPLC,Holdtime,output_error,o
    utput_DynamicZ,output_I1,output_I2,output_V1,output_V2)
```

Code explanation

NOTE

It may be necessary for you to learn some basic knowledge of the Python® programming language before you continue. You can download the Python IDE (Python Integrated Development Environment) to learn more about basic syntax and the environment. To download the Python IDE, go to https://www.python.org/downloads/ (python.org/downloads/).

Here is an explanation of the code presented previously:

Lines 1 to 4: Import declarations

Lines 6 to 9: Declaring functions

The declare function must follow the rule of indentation plus a colon.

NOTE

Declaration function rule (Python syntax rule): The keyword def starts the function declaration, followed by the function name, and the arguments in parentheses. Multiple arguments are separated with commas.

Declaration variable rule (Python syntax rule): Parameters do not specify a data type. In Python, variables are never explicitly typed. Python determines what type a variable is and keeps track of it internally.

Lines 10 to 74: Function descriptions and details

The document string of the function is between the triple quotes. You can add other comments about the functions.

NOTE

Paragraph rule (Python syntax rule): Python functions have no explicit beginning or end, and no curly braces to mark where the function code starts and stops. The only delimiter is a colon (:) and the indentation of the code itself. Indenting starts a block and not indenting ends it.

This means that whitespace is significant and must be consistent. In this example, the function code, including the doc string, is indented. It does not need to be four spaces; it just needs to be consistent. The first line that is not indented is outside the function.

The following example shows indentation errors:

```
def perm(l):                          # error: first line indented
for i in range(len(l)):                 # error: not indented
    s = l[:i] + l[i+1:]
        p = perm(l[:i] + l[i+1:])     # error: unexpected indent
        for x in p:
              r.append(l[i:i+1] + x)
            return r                    # error: inconsistent indent
```

The correct code would look like this:

```
def perm(l):
   for i in range(len(l)):
          s = l[:i] + l[i+1:]
          p = perm(l[:i] + l[i+1:])
          for x in p:
                 r.append(l[i:i+1] + x)
   return r
```

For more information about the Python resource, go the following web addresses:

- www.python.org/downloads/
- www.activestate.com/Products/ActivePython/
- www.python.org/doc/

Post data function in PTM

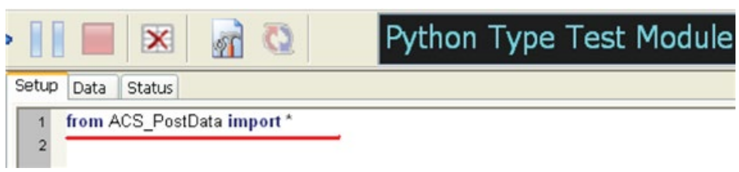
In CTM, the data post functions are ACSPostDataInt(char *, int *), ACSPostDataDouble(char *, double *) and ACSPostDataString(char *, char*). But in an ACS PTM, those functions can no longer be used and the following functions are added to take their place.

- ACSPostDataInt(data_name, data_val);
- ACSPostDataDouble(data_name, data_val);
- ACSPostArrayInt(data_name, data_array, data_num);
- ACSPostArrayDouble(data_name, data_array, data_num);

NOTE

To use these functions in ACS, you should import the ACS_PostData at the beginning of the Python source code, as shown in the following figure.

Figure 231: Import the ACS_PostData line in PTM header



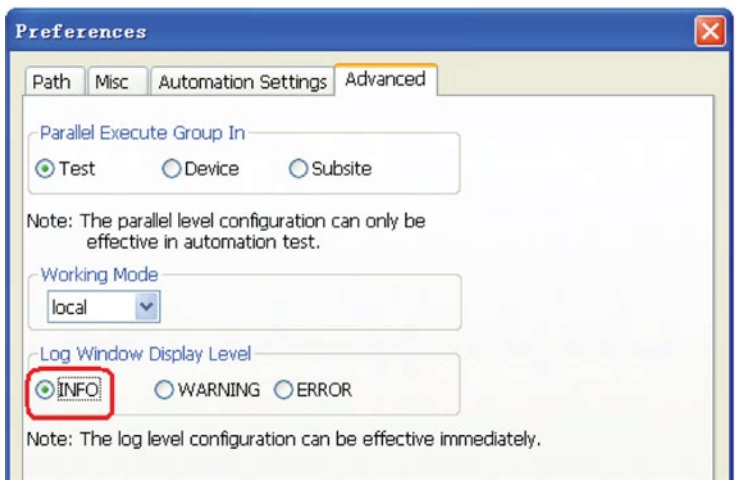
Using Log Manager in PTM

If you want to debug the program of a PTM, use the LogManager. In the Preferences setting, you can change the log level for debug or production run.

For the PTM Test module, you can use logInfo, logError, and logCritical to print log information in the ACS log window if this line is added in the header “from ACS_PostData import.*” The logCritical is the highest level and cannot be confined by all level settings.

For example, in the ACS Preferences dialog, set the info level as INFO, as shown in the following figure.

Figure 232: Select INFO Level in Advanced tab



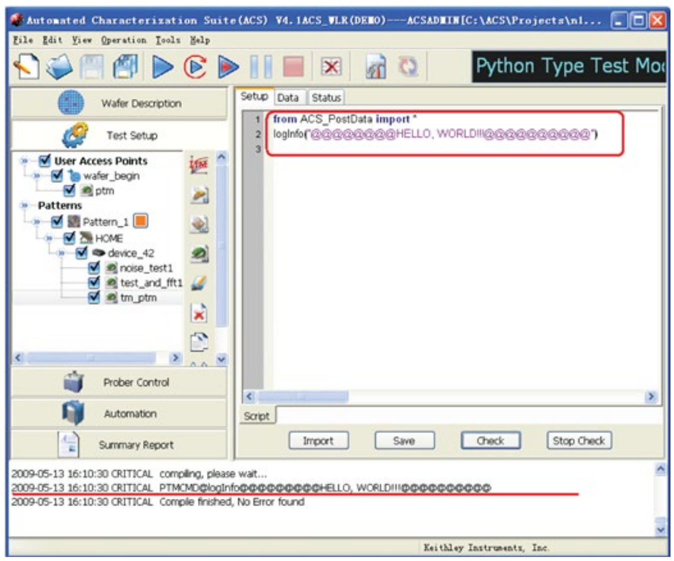
Then add a new PTM test module and add the following lines:

```
from ACS_PostData import *
logInfo("@@@@@@@@HELLO, WORLD!!!@@@@@@@@@@")
```

Select **Check** to compile or **Run** to run this PTM. The string is printed in the log window, as shown in the following figure. In info level, the logError and logCritical can also be used.

It will not print a log message after setting the log level to Warning or Error. If the log level is set to Error, logError, and logCritical are used. The logCritical is the highest level and is not confined by all level settings, as shown in the following figure.

Figure 233: Print line in log window



PTM with the .xrc GUI file

The `.xrc` is a tool used for creating a user interface. As with a STM in ACS, a PTM can include a `.xrc` file for a test module.

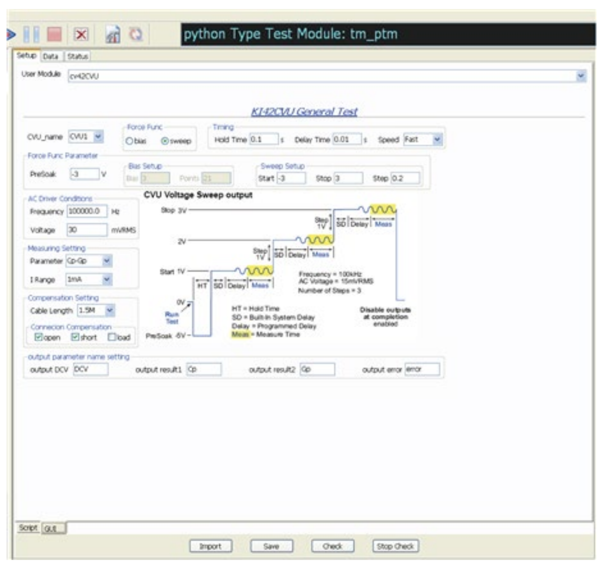
If the file imported has a corresponding `.xrc` user interface file, ACS automatically loads and displays the user interface. You can set parameters on the user interface so that you do not need to edit parameters on the Script tab.

You can generate the `.xrc` file. For instructions on how to create a `.xrc` user interface, refer to Create a TSP GUI file using XRC (on page 5-129).

To configure a PTM with .xrc:

1. After the `.xrc` PTM is imported, select the test name in the configuration navigator. The module opens and the user interface is shown by default. The module is in the Script tab of PTM work area. The following figure shows a user interface related to the module.

Figure 234: Importing a XRC PTM

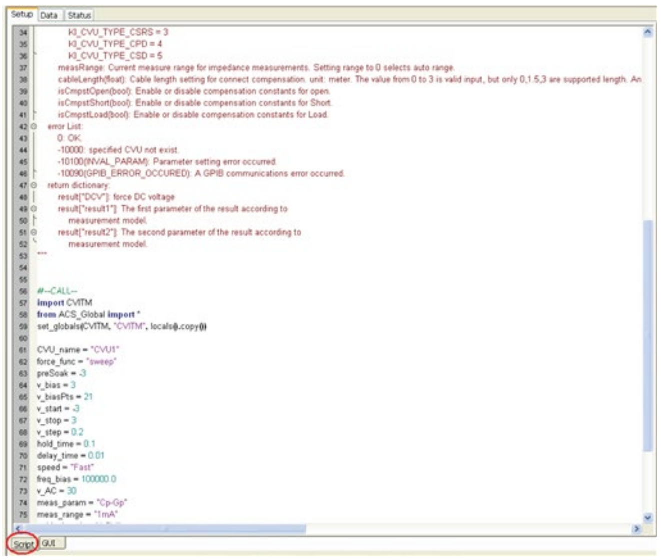


2. Set the input parameters.
3. Save the configuration using the Save function on the toolbar.

The inputs in user interface are related to the function call parameters in the script tab.

To open the corresponding script tab, select Script tab, as shown in the following figure.

Figure 235: Script tab of XRC PTM



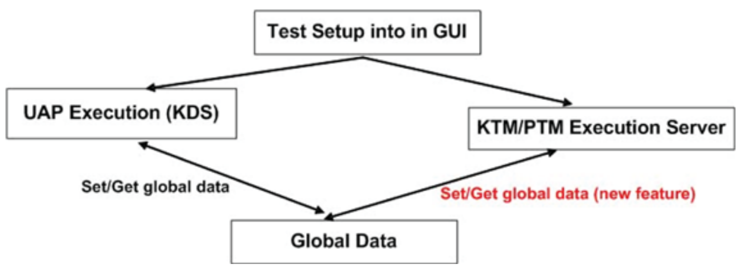
If one input parameter in the user interface is empty, the value of this input parameter is not valid in the function call of the script. After the correct input is entered, the function call of the script tab of this test module will change at the same time.

Global data support in PTM and UAP

In a PTM, ACS supports two-sided data flow:

- Global data can be used in an UAP.
- A PTM can change or set new global data, which can be used in any UAP or any other PTM after the PTM settings run over. The usage is no different than using the variable in global data, as shown in the following figure.

Figure 236: Data flow

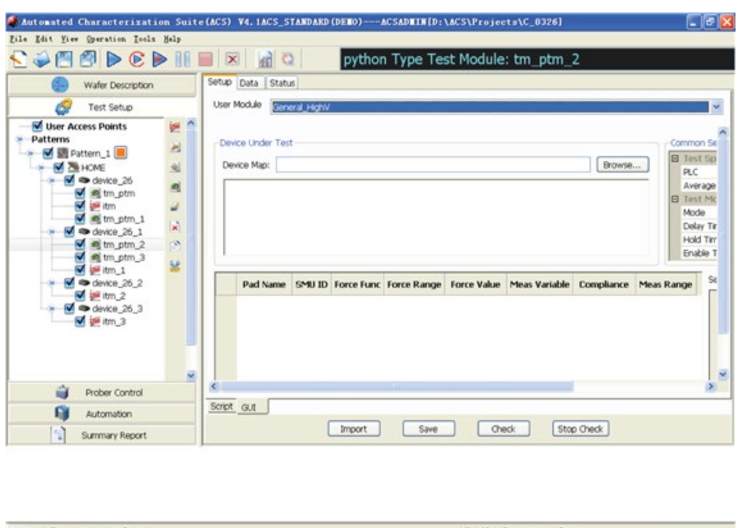


General module test in a PTM

To configure the device under test:

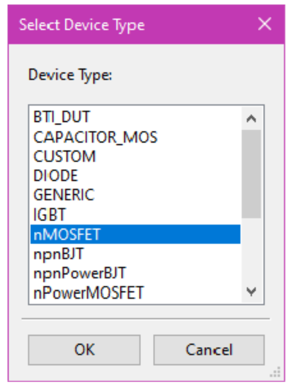
1. Import the PTM module GeneralTestLib.py from PTMlib and choose the General_HighV Module from the User Module list, as shown in the following figure.

Figure 237: General module test



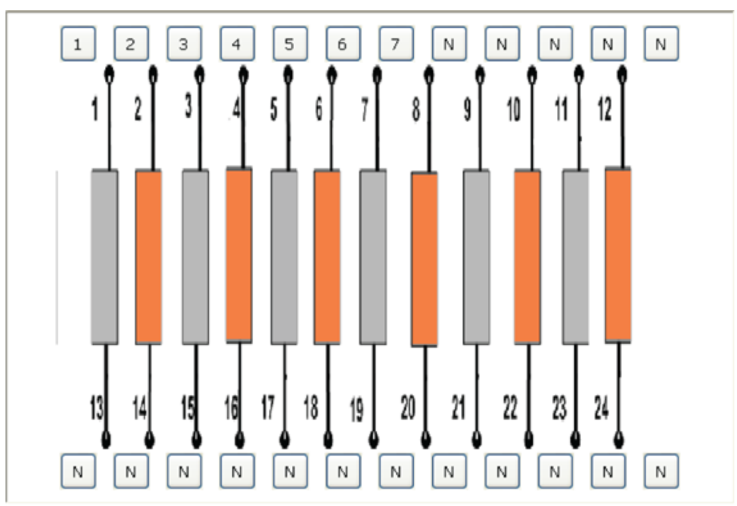
2. Select **Browse**. The Select Device Type dialog opens showing the device types, as shown in the following figure.

Figure 238: Select Device Type



3. Configure the device pads with different SMUs, as shown in the following figure.

Figure 239: Device map example



NOTE

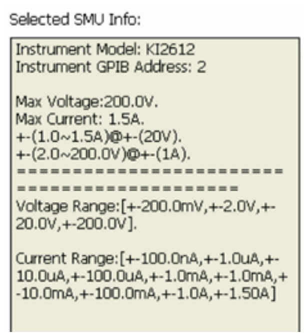
In the general module test, several types of SMUs are available for devices to perform the test. If ACS software is installed on a computer, the Model-4200A SMU cannot be used in this general test. This means that the Model-4200A SMU is not an option in the SMU box. Only when the ACS software is installed on the Model-4200A is the Model-4200A SMU used, as shown in the following figure.

This dialog shows the SMU information.

NOTE

If the Series 2400 SourceMeter is used in the general test module test, it must have a GPIB, LAN, or USB address and its command set must be set to SCPI.

Figure 240: Selected SMU information



Configuring the SMUs Force and Measure table

The Forcing Functions and Measure Options table are associated with an instrument object that is assigned to a device terminal. The Forcing Functions and Measure Options table items configure the forcing function and measurements implemented by the instrument. This section explains the following:

- Understanding and configuring each available forcing function.
- Understanding and configuring the measurement options that are associated with a forcing function. These are discussed generically, for all forcing functions, at the end of this section.
- Understanding and configuring the other controls on the Forcing Functions and Measure Options dialog.

SMUs Bias and Sweep and Step Setting Table

There are three parts in Forcing Functions/Measure options tables in the general module setup tab:

- The Pad name and the SMU ID.
- The force functions settings.
- The measure options settings, as shown in the following figure.

Figure 241: Forcing functions measure options table

Pad Name	SMU ID	Force Func	Force Range	Force Value	Meas Variable	Compliance	Meas Range
ForceHi_1	SMU1	Bias I	auto	0.000000	None	0.100000	
HeatHi_1	SMU2	Bias V	auto	0.000000	None	0.100000	
ForceHi_2	SMU3	Sweep V	auto	[0, 0, 1]	None	0.100000	
HeatHi_2	SMU4	Sweep I	auto	[0, 0, 1]	None	0.100000	
ForceHi_3	SMU5	Step V	auto	[0, 0, 1]	None	0.100000	
HeatHi_3	SMU6	Step I	auto	[0, 0, 1]	None	0.100000	
ForceHi_4	SMU7	Open					

Configure the Common Settings

The Common Settings configure the general module timing settings. There are three areas in the Common Setting: Test Speed; Test Mode; Advanced, as shown in the following figure.

Figure 242: Common Settings

Common Settings

Test Speed	
PLC	1
Average	0
Test Mode	
Mode	Sampling
Delay Time(s)	0
Sample Number	1
Hold Time(s)	0
Enable Timestamp	False
Advanced	
Test End Reset	False
Enable Trigger	False

Understanding test speed

The entries in the PLC and #Average list boxes control the A/D (analog-to-digital) converter integration time used to measure a signal. Each measured reading by a SMU is the result of one or more A/D conversions. A short integration time for each A/D conversion results in a relatively fast measurement speed, at the expense of noise. A long integration time results in a relatively low-noise reading, at the expense of speed. The integration time setting is based on the number of power line cycles (NPLCs). For example:

- For 60 Hz line power, 1.0 PLC = 16.67 ms (1/60).
- For 50 Hz line power, 1.0 PLC = 20 ms (1/50).

Understanding test modes

In the mode area of the Common Settings dialog, you can select the sweeping test mode or the sampling test mode.

The sweeping test mode applies to any general module in which one or more forced voltages or currents vary with time.

The sampling test mode applies to any general module in which all forced voltages or currents are static (bias I or bias V), with measurements typically being made at timed intervals. Refer to the Forcing Function Summary Table discussed earlier. For example, sampling mode would be used to record a few static measurements, or to time-profile the charging voltage of a capacitor while forcing a constant current.

The mode observes and changes the test mode as follows:

- For a completely new general module, the mode allows you to choose the sweeping or sampling test mode. Selecting the appropriate test mode simplifies configuration options and helps to avoid errors:
  - Only the static forcing functions are configurable in the sampling mode.
  - Only mode-appropriate timing options are configurable.
- For an existing library PTM that is in the sampling mode, the mode area allows you to change to the sweeping mode if you want to change some of the static forcing functions to sweeping forcing functions.

Sweeping mode

If any terminal of the device under test is configured for a step or sweep forcing function, sweeping mode is automatically enabled. You can configure sweeping mode Delay Time and Hold Time settings in the Common Settings dialog, as shown in the following figure.

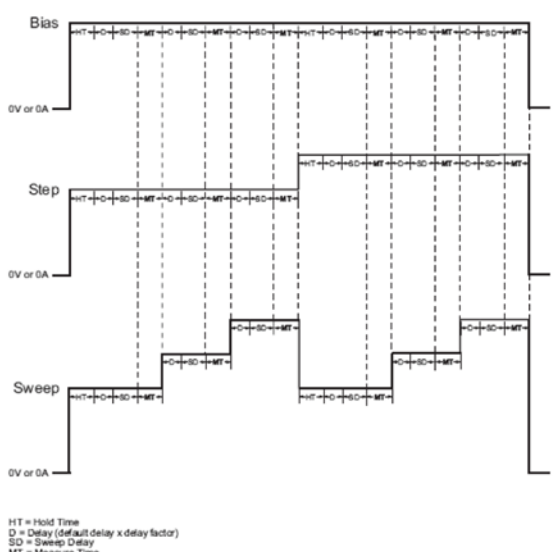
Figure 243: Sweeping mode setting

Test Mode	
Mode	Sweeping
Delay Time(s)	0
Hold Time(s)	0
Enable Timestamp	False

The starting voltages or currents of a sweep may be substantially larger than the voltage or current increments of the sweep. Accordingly, the source settling time required to reach the starting voltages or currents of a sweep may be substantially larger than the settling times required to increment the sweep. To compensate, you can specify a Hold Time delay to be applied only at the beginning of each sweep. You can specify a Hold Time delay of 0 s to 1000 s. The default Hold Time is 0 s.

The following figure shows an example sweeping mode timing diagram.

Figure 244: Sweeping mode timing diagram



When the test is started, the Hold Time (HT) provides extra settling time for the initial step change of the source. The Hold Time is a global setting. Note that the Hold Time is applied at the start (only) of each sweep; therefore, it is the same for all SMUs that are part of the test (as in the previous figure that includes sweep, step, and bias).

SMU will Delay Time (D) before making a measurement after forcing.

Sampling mode

The sampling mode measures voltage and current as a function of time while forcing constant voltage and current (bias I or bias V). For example, the sampling mode would be used to measure voltage while forcing a constant current. Time is measured relative to when the SMUs apply the forced voltage or current (that is, t = 0 at the step change from 0.0 V or 0.0 A to the applied voltage or current).

ACS enables the sampling mode option only when all terminals of the device under test are configured for static forcing functions: voltage bias, or current bias. If any terminal of the device under test is configured for a step or sweep forcing function, the sampling mode option is unavailable, as shown in the following figure.

Figure 245: Sampling mode setting

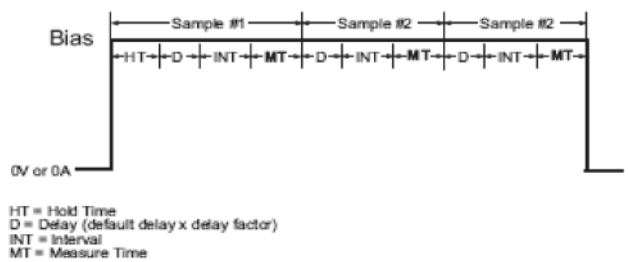
Test Mode	
Mode	Sampling
Delay Time(s)	0
Sample Number	1
Hold Time(s)	0
Enable Timestamp	False

You can configure three sampling mode settings: Delay Time, Sample Number, and Hold Time.

The Delay Time, Sample Number, and Hold Time settings control the sampling mode, as follows:

- Delay Time—Specifies the time between measurements (data points). Delay Time can be set from 0 to1000 s.
- Sample Number—Specifies the number of data points to be acquired. Sample Number can be set from 1 to 4096.
- Hold Time—After initial application of voltage or current by the SMUs, the source settling times can be substantial. To allow for settling, you can specify an extra Hold Time delay to be applied before making the first measurement. You can specify a Hold Time from 0 to 1000 s.

Figure 246: Sampling mode timing diagram

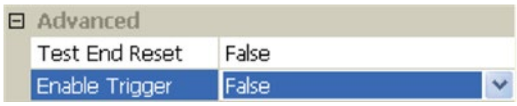


- If needed, a Hold Time (HT) can be used to allow for extra source settling after the initial application of voltage or current by the SMUs. Hold Time is a global setting and is therefore the same for all SMUs in the test system.
- A range-dependent Delay (D) is automatically applied by a SMU before each measurement, to allow for source settling. All SMUs in the test system are synchronized. Therefore, the Delay time applied by the most delayed SMU is the delay time applied to all SMUs.
- SMU will delay Measure Time (MT) before taking a measurement after forcing.

Understanding Advanced

In the Advanced area, there are two lists for you to configure for your requirement. They are Test End Reset and Enable Trigger, as shown in the following figure.

Figure 247: Advanced common settings



For the Test End Reset configuration, there are two choices for you to configure. If you choose False, that means your test result will not be cleared. If you choose True, after the test is finished, it will clear the test results.

For the Enable Trigger configuration, there are two choices. If you want to use trigger in your test configuration, you should choose True, otherwise choose False.

NOTE

For the most SMUs, choose the Enable Trigger with False is generally accepted.

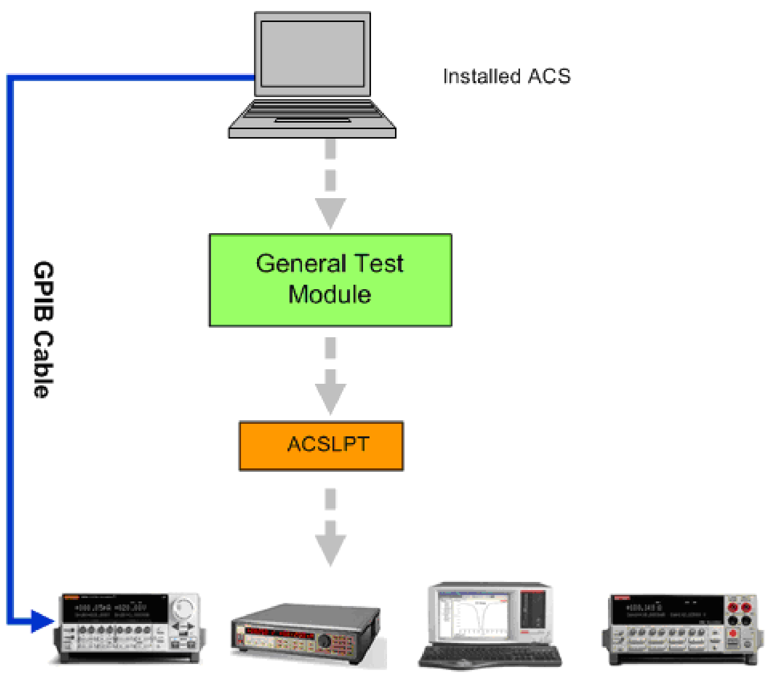
Configure a general-purpose PTM

The general-purpose PTM performs measurements with SMUs from different Keithley product lines. The user interface provides an interface like the standard interactive test modules (ITM) for the Series 2600 and 2600B, therefore, no programming is required.

You can see in the following figure that the general-purpose test module can be used with any combination of the following SMUs: Series 2400, Series 2600, Series 2600B, 4200A, and Model 237.

For more information on how to configure different GPIB instruments, refer to ACS Example hardware configurations (on page 11-1).

Figure 248: General purpose module with multiple SMUs



The general-purpose module in ACS can provide a total solution for component characterization tests with high power requirements. Because the general-purpose module will work with a mix of several types of SMUs, it can also be used in other applications, such as solar cell and LED testing.

Due to hardware platform differences in the module-compatible SMUs, the general-purpose module interface only supports the following forcing functions: bias V/I, sweep V/I, and step V/I. Unlike the Series 2600 and 2600B ITM, the general-purpose module does not support log sweeps, dual sweeps, or list sweeps, and has limited autoranging.

The external trigger feature in the Common Settings is only used when using multiple Model 237 SMUs with a Model 2361 Trigger Controller. To use this feature, you must configure the Model 2361 as GPI1 (general-purpose instrument 1) in the ACS hardware configuration.

The other Common Settings are like those in the timing dialog of the standard Series 2600 and 2600B ITM timing settings. The timing and speed of these tests is limited if you combine different generations of hardware. For the fastest test times, you should always use the Series 2600B SourceMeters in the standard tests associated with those instruments.

NOTE

When using only Series 2600 or 2600B SourceMeter instruments for a particular test, it is recommended that you use either an ITM or STM for optimum SMU performance.

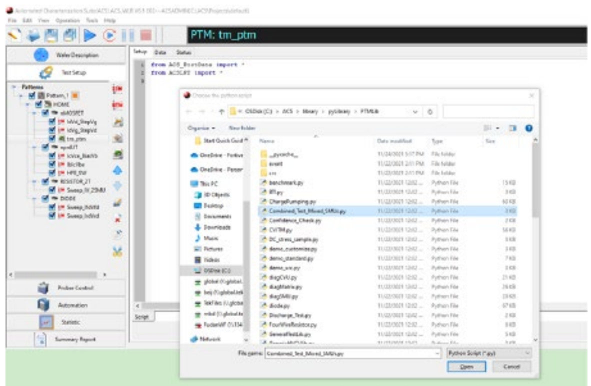
Configure the hardware system

Before you configure a general-purpose test module in the system, you need to configure and check the hardware first. For more information on how to configure different GPIB, ethernet, and TSP instruments through ACS, refer to ACS Hardware Configurations (on page 10-1).

Add a new general-purpose module

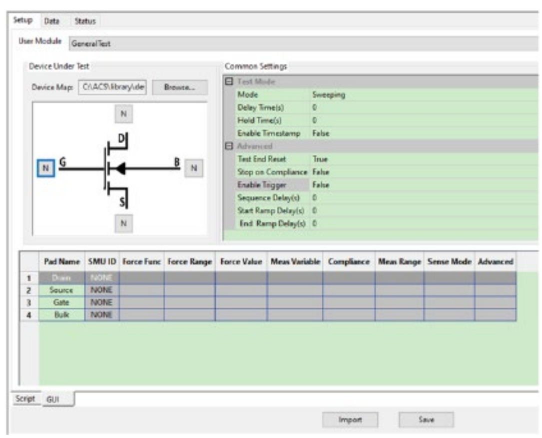
Select the **New PTM** icon and then select **Import**. In the dialog that opens, select the `Combined_Test_Mixed_SMUs.py` script, as shown in the following figure.

Figure 249: Import the General PTM



The general-purpose PTM dialog opens on the user interface, as shown in the following figure.

Figure 250: General purpose PTM GUI



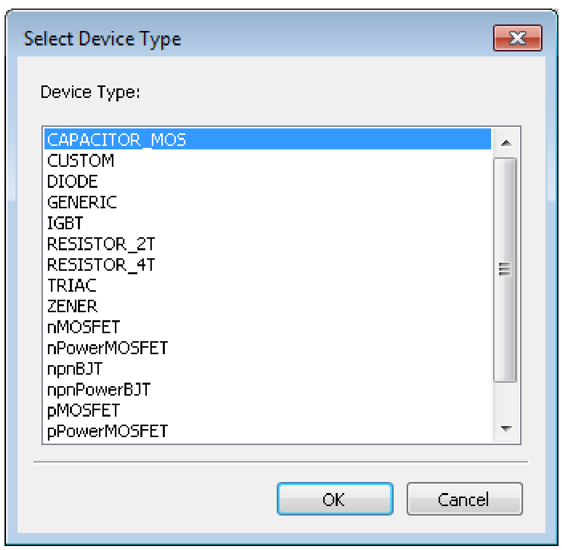
General-purpose PTM configuration

After inserting a general-purpose PTM into your project, you must configure it to meet your test requirements.

To configure a general-purpose PTM:

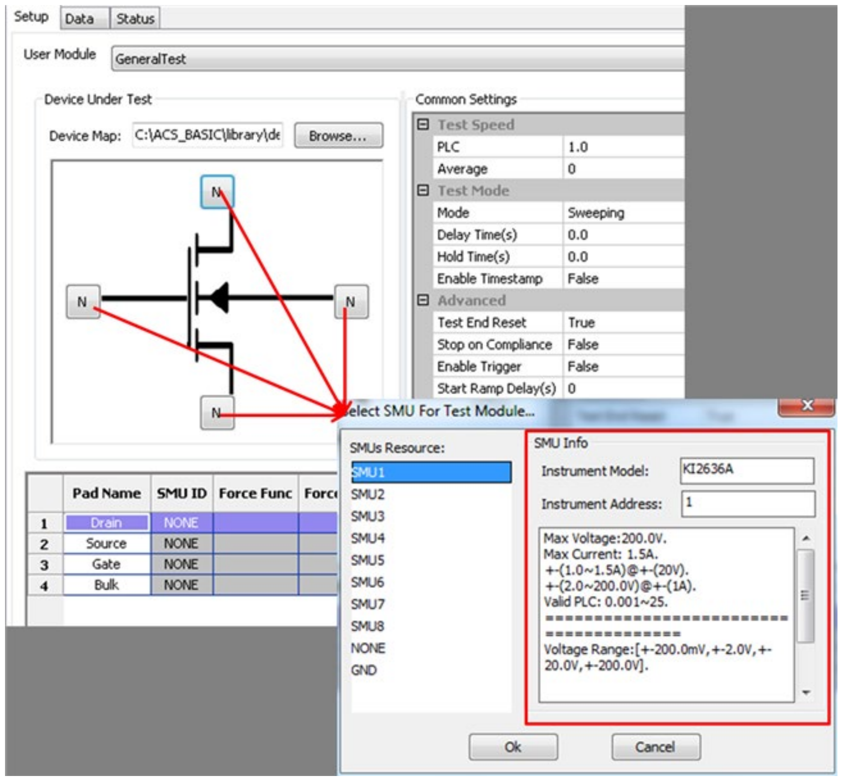
1. Select **Browse. T**he Select Device Type dialog displays types, as shown in the following figure.
2. Select a device type. The device type you select in the general-purpose PTM user interface is independent of the device in the configuration navigator. You must select a device in the module user interface to configure the general-purpose PTM. For consistency with device definitions, you should select the same device for the general-purpose module that the PTM exists under in the configuration navigator. By default, the imported PTM opens with the same device as the one in the configuration navigator where the PTM is created.

Figure 251: Select Device type dialog



3. Specify which SMU is connected to each device terminal. Select a device terminal and the Select SMU For Test Module dialog opens. Select the appropriate SMU. The device terminal reflects your selection, as shown in the following figure.

Figure 252: Map SMU and device terminal



NOTE

The 4200A SMU cannot use general-purpose PTMs if ACS is installed on a PC. This means that the 4200A SMU will not be displayed in the Select SMU For Test Module dialog. Only when ACS is installed on the 4200A will the 4200A-SMU be available for general-purpose PTMs.

If a connection to the device is deemed unsafe by the software, you will not be able to run the test and you will see an error message in the Help area of the user interface. Here are some items to consider when configuring your hardware:

- Non-A or B 26xx SMUs cannot be mixed with Model 265xA instruments.
- A Model 2657A SMU cannot be mixed with Models 2651A, 2601A, 2601B, 2602A, or 2602B.
- For two-terminal devices, a Model 2651A cannot be mixed with low-current SMUs.
- For three-terminal devices, if a Model 2651A SMU is connected to the collector, then the emitter must be connected to ground or another Model 2651A SMU. If a Model 2651A SMU is connected to the base, then the emitter and collector must be connected to ground or another Model 2651A SMU emitter and collector.
- For four-terminal devices, if a Model 2651A SMU is connected to the drain, a low-current SMU cannot be connected to the source.

You must set the SMU's Force Function, Force Range, Force Value, Measure, and Meas Range. For more information on how to set force and measure functions and variables, continue to the next topic in this section.

You must also set the Common Settings information in the PTM user interface. The setting includes the Test Speed, Test Mode, and Advanced information.

Configure the Force and Measure parameters

The Force and Measure settings are associated with an instrument object that is assigned to a device terminal. These settings configure the forcing function and measurements implemented by the instrument.

The following figure indicates all the parameters that need to be configured for forcing and measuring.

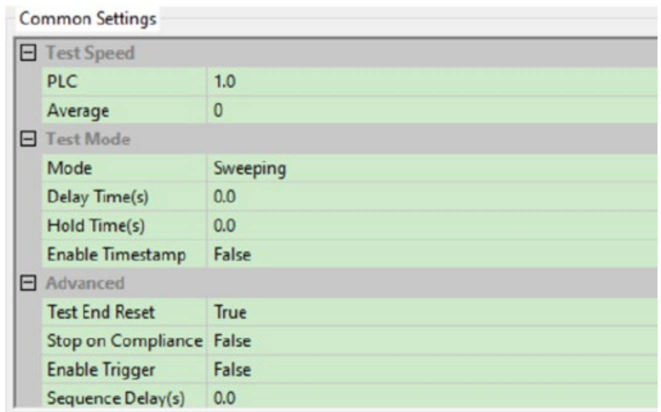
Figure 253: Force and Measure parameters

	Pad Name	SMU ID	Force Func	Force Range	Force Value	Meas Variable	Compliance	Meas Range	Sense Mode	Advanced
1	Drain	SMU1	Bias V	auto	0	None	0.01		Local	...
2	Source	SMU3	Bias V	auto	0	None	0.01		Local	...
3	Gate	SMU2	Bias V	auto	0	None	0.01		Local	...
4	Bulk	SMU4	Bias V	auto	0	None	0.01		Local	...

Common settings

The Common Settings are used primarily to configure the timing settings for the general-purpose PTM. There are three areas in the Common Settings: Test Speed, Test Mode, and Advanced, as shown in the following figure.

Figure 254: General purpose test module Common Settings



Test Speed: Controls the A/D (analog-to-digital) converter integration time used to measure a signal. Each measured reading by a SMU is the result of one or more A/D conversions. A short integration time for each A/D conversion results in a relatively fast measurement speed, however, there will be more noise. A long integration time results in a relatively low noise reading but takes more time.

- **PLC** (power line cycle): For optimum immunity to line cycle noise, use an integer number for the PLC (for example, 1, 2, 3, and so on). The integration time setting is based on the number of power line cycles (NPLCs):

  For 60 Hz line power, 1.0 PLO = 16.67 ms (1/60)

  For 50 Hz line power, 1.0 PLO = 20 ms (1/50)

- **Average**: The Average edit field is only used for series 23x and 2400 SMUs. For 2400 SMUs, the average number will range from 0 to 1000. For 23x SMUs, select these numbers = [0,1,2,4,8,16,32] (note that zero [0] disables the average number setup.

The next table shows the range of allowed values for a NPLC and the Average field setting.

Model	PLC	Average
KI24xx	[0.01 to 10]	[0 to 100]
KI23x	[0.01 to 1]	[0, 1, 2, 4, 8, 16, 32]
KI26xx	[0.001, 25]	ignored
KI4200 (on the 4200A)	[0.01 to 10]	ignored

Test Mode: The edit field lets you choose the sweeping or sampling test mode.

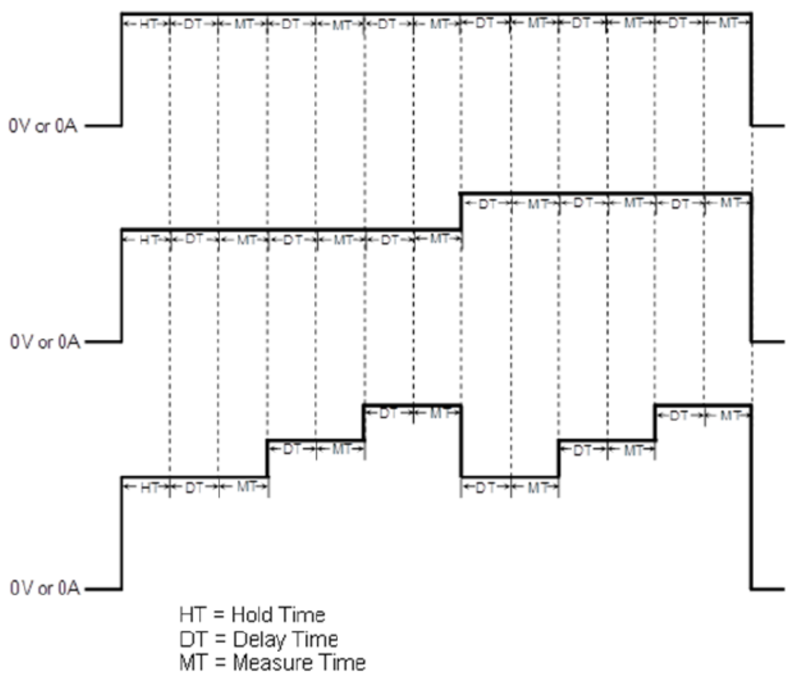
- **Sweeping**: Applies to any general test module in which one or more forced voltages or currents vary with time.

Figure 255: Sweep Test Mode settings

Test Mode	
Mode	Sweeping
Delay Time(s)	0.0
Hold Time(s)	0.0
Enable Timestamp	False

If any terminal of the device under test (DUT) is configured for a sweep forcing function (step or sweep), sweep mode is automatically enabled. In this case, you can configure two other test mode settings in the Common Settings: Delay Time and Hold Time.

Figure 256: Sweep mode timing diagram



  - **Hold Time**: Specifies the time delay before the first measurement is made. After the initial application of voltage or current by the SMUs, the source settling time can be substantial. To allow for settling, you can specify an extra Hold Time delay to be applied before making the first measurement. You can specify a Hold Time from 0 to 1000 seconds. Hold Time is a global setting and is therefore the same for all the SMUs in the test system.
  - **Delay Time**: The Delay Time (DT) allows the source to settle at every step of a sweep. All SMUs in the test system are synchronized, therefore, the delay time applied by the most delayed SMU is the time applied to all the SMUs. You can specify a delay time from 0 to 1000 seconds. The default delay time is 0 seconds.
- **Sampling Mode**: Applies to any general test module where all the forced voltages or currents are constant (bias I or bias V), and measurements are made at timed intervals. For example, sampling mode records one or more static measurements, or to time-profile the charging voltage of a capacitor while forcing a constant current.

ACS enables the sampling mode option only when all the terminals of the device under test are configured for static forcing functions: voltage bias, or current bias. If any terminal of the device under test is configured for a step or sweep forcing function, the sampling mode option is unavailable, as shown in the following figure.

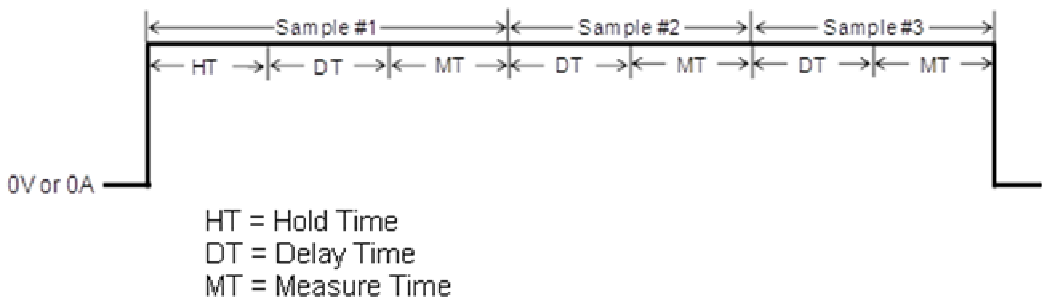
Figure 257: Sampling Test Mode settings

⊟ Test Mode	
Mode	Sampling
Delay Time(s)	0.0
Sample Number	1
Hold Time(s)	0.0
Enable Timestamp	False

The sampling mode measures voltage or current as a function of time while forcing constant voltage or current. Time is measured relative to when the SMU applies the forced voltage or current.

If a General-Purpose Test Module is configured for sampling mode, you can configure three sampling mode settings: Delay Time, Sample Number, and Hold Time, as shown in the following figure.

Figure 258: Sampling mode timing diagram

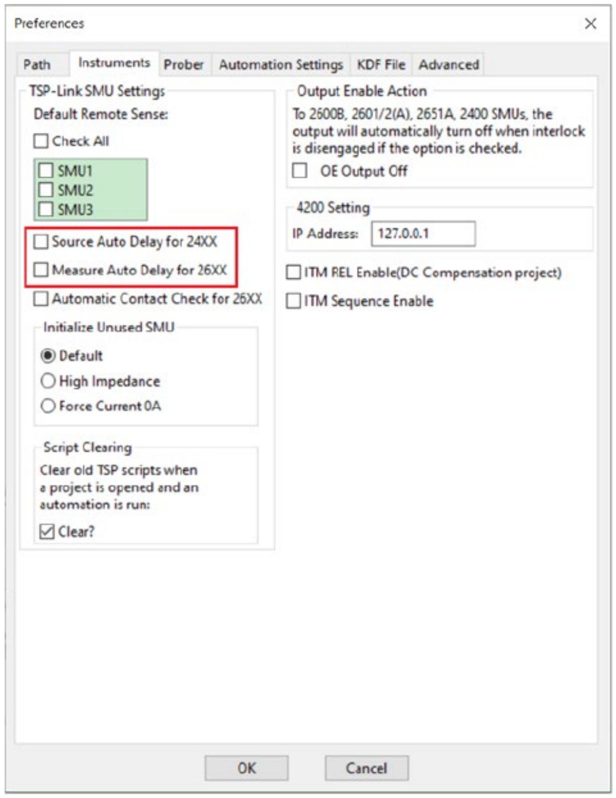


- **Hold Time**: Specifies the time delay before the first measurement is made. After the initial application of voltage or current by the SMUs, the source settling time can be substantial. To allow for settling, you can specify an extra Hold Time delay to be applied before making the first measurement. You can specify a Hold Time from 0 to 1000 s. Hold Time is a global setting and is therefore the same for all the SMUs in the test system.
- **Delay Time**: Specifies the time interval (INT) between measurements (data points). Delay Time can be set from 0 to 1000 s.
- **Sample Number**: Specifies the number of data points to be acquired. The Sample Number can be set from 1 to 4096.

A range-dependent delay is automatically applied to a SMU before each measurement to allow for source settling. All the SMUs in the test system are synchronized, therefore the range-dependent delay time applied by the most delayed SMU is the time applied to all the SMUs. If the **Source Auto Delay for 24xx** in the instruments tab of the Preferences dialog is selected (found in Tools > Preferences), the range-dependent delay time is applied to the Models 2450, 2460, and 2470. If **Measure Auto Delay for 26xx** is selected, the range-dependent delay time will be applied to the Models 2635, 2636, 2635A, and 2636A.

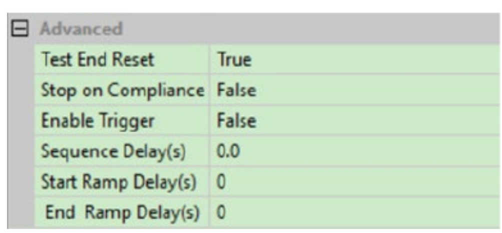
The SMU will delay the Measure Time (MT) before taking a measurement after forcing.

Figure 259: Auto Delay on Preferences Instruments tab



Advanced: In the Advanced area there are five items for you to configure for your testing requirements, as shown in the following figure.

Figure 260: Advanced area settings



Test End Reset: There are two choices:

- False: your test results will not be reset in the instruments when testing has completed.
- True: after testing is complete, all the test results will be reset in the instruments.

Stop on Compliance: There are two choices:

- False: the test will run to completion, even if a SMU enters compliance.
- True: the test will terminate immediately, if a SMU enters compliance.

Enable Trigger: Used for the Model 237 with the Model 2361 Trigger controller. There are two choices:

- False: do not capture the measurement trigger value.
- True: capture the measurement trigger value.

NOTE

Except for the Model 237, it is acceptable to always choose false for the Enable Trigger function for SMUs.

Sequence Delay: Determines the delay before turning on the SMU of the Pad.

Start Ramp Delay: If a soft ramp is enabled at the start of the test, this setting determines the delay time between the steps.

End Ramp Delay: If a soft ramp is enabled at the end of the test, this setting determines the delay time between the steps.

NOTE

For most SMUs, you can choose False for the Enable Trigger setting.

Configure the Force and Measure functions and variables

The Force and Measure settings are associated with an instrument object that is assigned to a device terminal. These settings configure the forcing function and measurements implemented by the instrument, as shown in the following figure.

Figure 261: Force and Measure functions variables

	Pad Name	SMU ID	Force Func	Force Range	Force Value	Meas Variable	Compliance	Meas Range	Sense Mode	Advanced
1	Drain	SMU79	Sweep I	auto	[0, 0, 1]	I+V	0.01	auto	Local	...
2	Source	GND								
3	Gate	SMU77	Bias V	auto	0	None	0.01		Local	...
4	Bulk	GND								

Force Func: There are seven functions that can be used in the Force Func parameter:

- Voltage bias (bias V)
- Current bias (bias I)
- Open
- Voltage sweep (sweep V)
- Current sweep (sweep I)
- Voltage step (step V)
- Current step (step I)

Sense Mode: There are two choices.

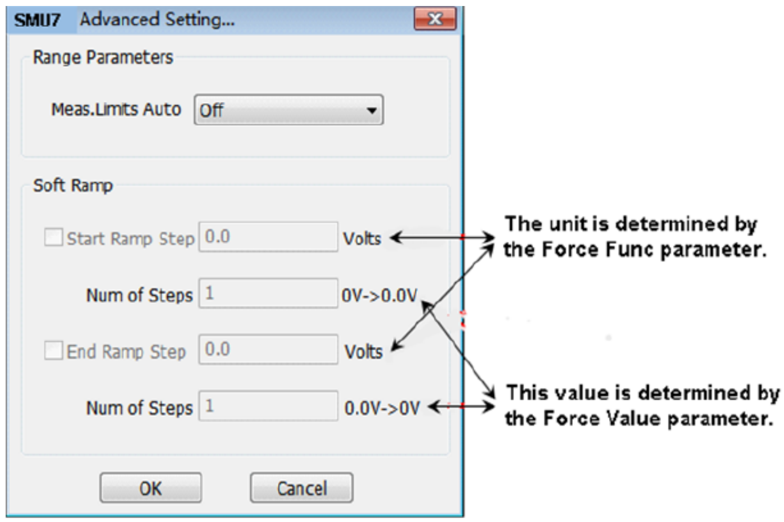
- Local: two-wire sensing
- Remote: four-wire sensing

NOTE

If the SMU is a Model 42xx-SMU, the Sense Mode parameter will be grayed out and not available to change.

Advanced: Double-click to open the Advanced Setting dialog, as shown in the following figure.

Figure 262: SMU Advanced Setting parameters



Meas. Limits Auto: When the Meas Variable parameter is set to current measurement (I) and Meas Range is set to auto, then Meas. Limits Auto is enabled. This setting specifies the minimum range that the SMU uses when automatically optimizing the measurements. This option saves testing time when the maximum resolution at minimum currents is not needed.

NOTE

When the Meas Range parameter is set to auto, the Meas. Limits Auto function is enabled (the default value is disabled). If the Meas Range parameter is not set to auto, the Meas. Limits Auto function is disabled.

The Meas. Limits Auto setting is available when a SMU is in any of the following modes:

Number	Force Func	Meas Variable	Meas Range
1	Sweep V	I+V	auto
2	Sweep V	I	auto
3	Sweep V	I+V (prog)	auto
4	Bias V	I+V	auto
5	Bias V	I	auto
6	Bias V	I+V (prog)	auto
7	Step V	I+V	auto
8	Step V	I	auto
9	Step V	I+V (prog)	auto

Soft Ramp: Allows the forced voltage or current to achieve the configured value after a certain number of steps.

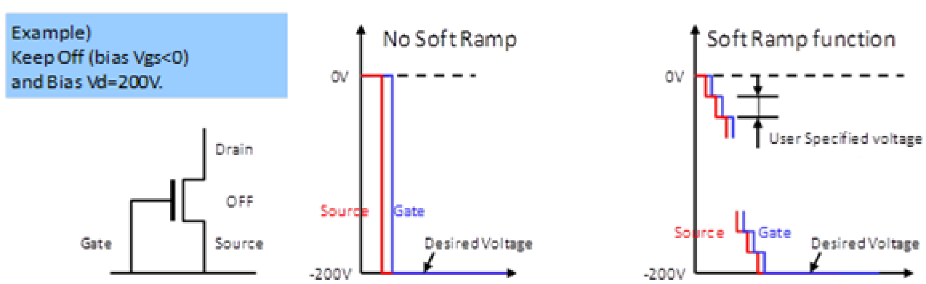
Start Ramp Step: The size of ramp steps at the beginning of the test.

Stop Ramp Step: The size of the ramp steps at the end of the test.

Num of Steps: The number of steps in the soft ramp.

The Soft Ramp is from 0 to the value (bias value or first and last sweep and list value). If the bias or first or last sweep and list point is 0, no soft ramp is necessary. In the "No Soft Ramp" example, the gate breakdown occurs because of the 200 V stress applied. However, with the "Soft Ramp function," you can keep the voltage differences at smaller levels until the selected voltage is applied, as shown in the following figure.

Figure 263: Comparison of Soft Ramp and without Soft Ramp



CVU connection compensation

It is recommended that you perform open and short compensations measurements before measuring the device capacitance. Once the compensation data is stored, it can be used repetitively across different devices if there is no change in instrument and cable connections.

You can correct offset and gain errors caused by the connections between the Model 4210-CVU and the device under test (DUT) by using CVU connection compensation. Correction is a two-part process:

1. Generate CVU connection compensation data for an open or short connection.
2. Enable open or short correction values before executing a test.

Perform CVU connection compensation any time the connection setup is changed. CVU connection compensation is not affected by changes in either temperature or humidity.

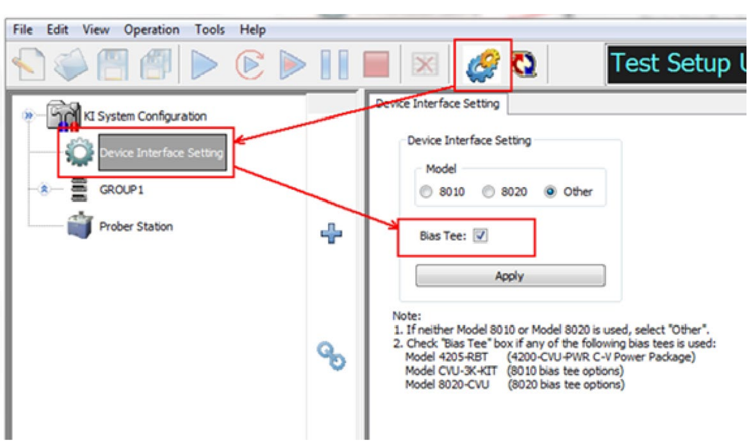
Generate connection compensation data

To open the CVU Compensation user interface

1. Select **Tools**.
2. Select **CVU Compensation**.

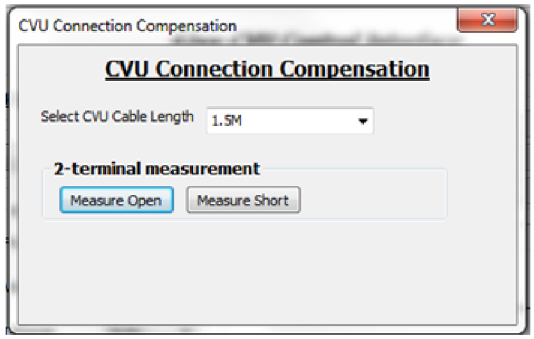
The device interface settings determine the CVU Compensation user interface that opens (for more information, refer to Device interface setting (on page 10-12)).

Figure 264: Device Interface Setting GUI



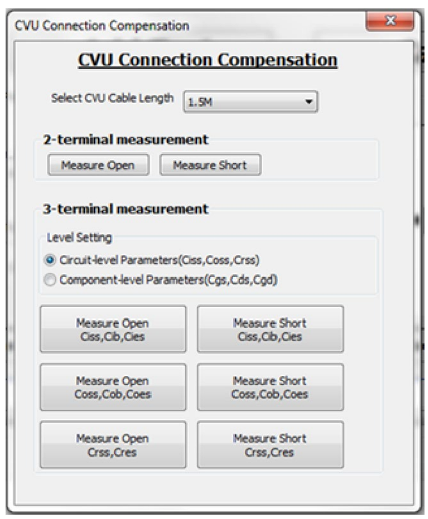
If the bias tee option is not selected, the following CVU compensation user interface appears:

Figure 265: CVU Compensation GUI without bias tee



However, if the bias tee option is selected, the CVU compensation user interface appears:

Figure 266: CVU Connection Compensation



Two-terminal measurement

Two-terminal measurement (without bias tee)

This measurement is applied to a CV test without a bias tee (for example, the CVITM; see Enable compensation).

- **Measure Open**: Select the button to generate open correction values.
- **Measure Short**: Select the button to generate short correction values.

See Measure open and measure short buttons (on page 5-193) for details on performing open and short corrections.

Two-terminal measurement (with bias tee)

This measurement is applied to a Generic HV CV PTM (see the following: Enable open or short correction for Generic HV CV PTM).

- **Measure Open**: Select the button to generate open correction values.
- **Measure Short**: Select the button to generate short correction values.

See Measure open and measure short buttons (on page 5-193) for details on performing open and short corrections.

Three-terminal measurement

This measurement is applied to both Generic HV CV PTM and device-specific PTM libraries.

Level Setting:

- **Circuit-level parameters**: This compensation data is applied to the following devices and libraries.

MOSFET	BJT	IGBT
Ciss	**Cib**	**Cies**
Coss	**Cob**	**Coes**
Crss		**Cres**

- **Component-level parameters**: This compensation data is applied to the following devices and libraries.

MOSFET	BJT	IGBT
Cgd	**Ccb**	**Cgc**
Cgs	**Ceb**	**Cge**
Cds	**Cce**	**Cce**

There are certain connection configurations that you can use with different devices and the compensation will only need to be performed once. The Measure Open and Measure Short buttons group the same capacitance terminologies for MOSFET, BJT and IGBT.

For example, the Ciss, Cib, and Cies input capacitance for MOSFET, BJT, and IGBT share the same connection configuration. The same compensation data can be used for all.

Using the CVU connection compensation

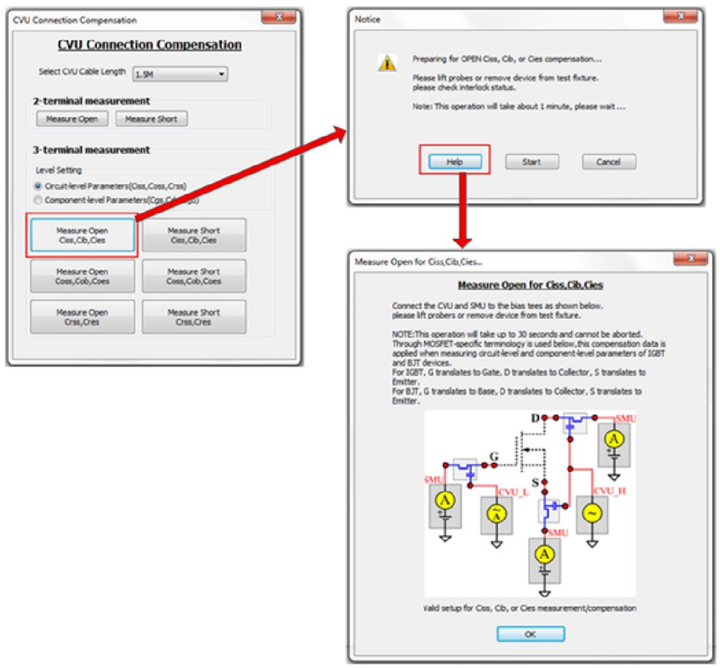
Use the menu to set the default SMA cable length.

- **0 M**: Use to make measurements at the terminals of the Model 4210-CVU (no cabling). This disables compensation.
- **1.5 M**: Use with the standard red SMA cables (1.5 m) supplied with the CVU (Keithley Instruments part number CA-447).
- **3.0 M**: Use the standard red SMA cables (3.0 m) (Keithley Instruments part number CA-446). This setting can also be used when using a switching matrix.

Measure open and measure short buttons

Based on the test that you want to perform, select Measure Open or Measure Short (see below for options). When you select Measure Open or Measure Short, a dialog opens. Select **Help** to see the connection information, as shown in the following figure.

Figure 267: Dialog for Measure Open Ciss, Cib, Ces



Make sure to set up the appropriate connection that appears once you select Help. For example, the previous figure shows the Measure Open Ciss, Cib, Ces CVU connection.

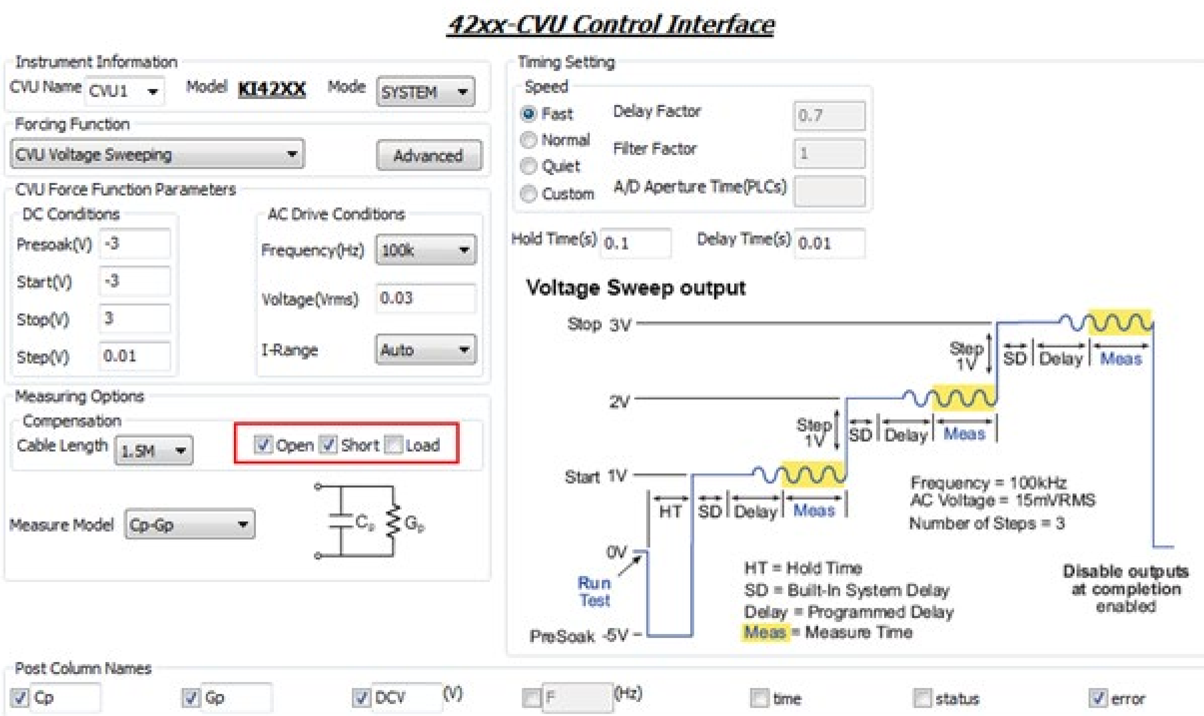
Once you have the test connections completed, select **Start** to begin generating open or short compensation data. The compensation data file that is created is saved in the computer directory `C:\ACS_BASIC\compensation`.

If you are testing two-terminal devices, the name of the generated file is `2TComp.csv`. If you are testing three-terminal devices, the name of the generated file depends on the test that you select. The format of the generated compensation date file name is similar to the following format: `Ciss_Cib_Cies_Comp.csv`.

Enable compensation

Compensation is done during test execution with open or short correction enabled. Open or short correction is enabled from the user interface Generic HV CV PTM or device-specific PTM, or the CVITM (ki42xxCVU). The compensation data can be seen in the Data tab for the Generic HV CV PTM or device-specific PTM:

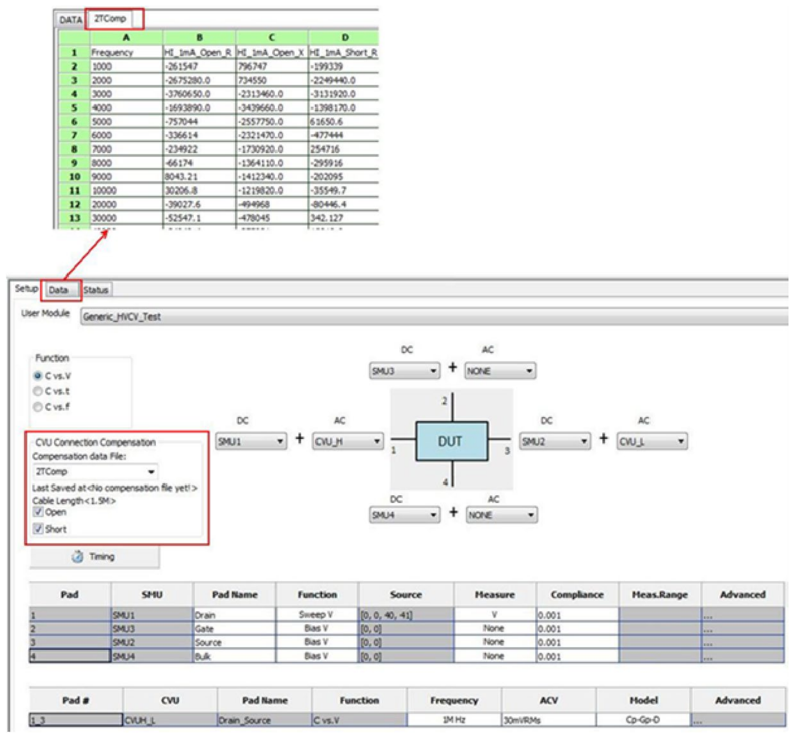
Figure 268: CVU Enable Open or Short Correction for CVITM



Enable open or short correction for Generic HV CV PTM

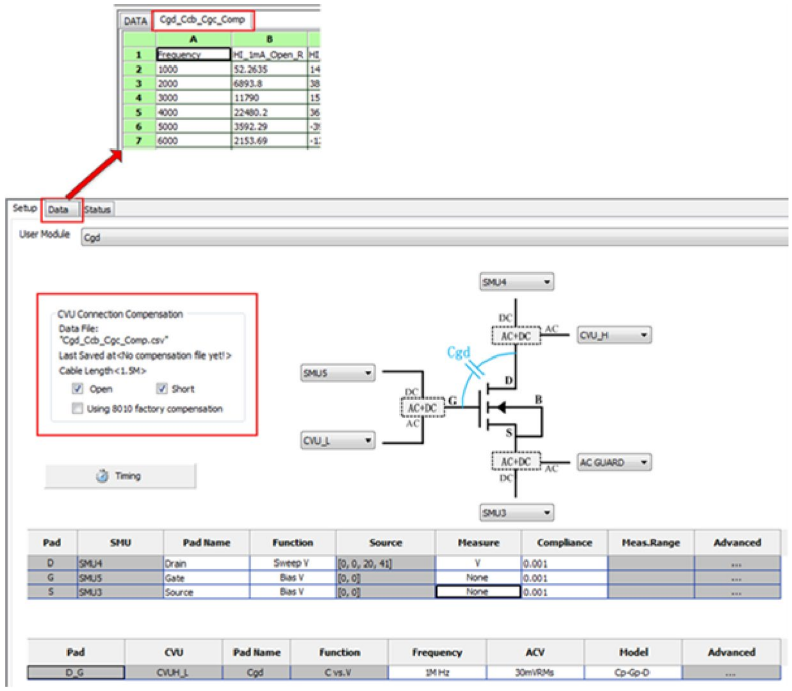
The following user interface shows where to enable compensation in the specific test module. Additionally, compensation data can be viewed in the Data tab of the test module. Compensation data appears as a second worksheet in that tab.

Figure 269: Enable open or short correction for Generic HV CV PTM



Enable open or short correction for device-specific PTM

Figure 270: Enable open or short correction for device-specific PTM



Within each test module, you have the option to select Using 8010 factory compensation. ACS stores default compensation data that was generated using cables and accessories supplied with remote bias tee kits (Model CVU-3K-KIT and PCT configurations).

You can load the Generic HV CV PTM using the default compensation data file from the Generic_HVCV_Test user interface. For a device-specific PTM you can use the default compensation data file by selecting the Using 8010 factory compensation option. Note the name of the default compensation data file will depend on the test selected (notice that the name begins with "factory_8010_"; it will be similar to this file name: "factory_8010_Ciss_Cib_Cies.csv").

Script Editor tool

Script Editor is a tool used to create and manage user libraries.

A user library is a collection of one or more user modules. User modules are Python programming language subroutines (or functions) or TSP script. User libraries are created to control instrumentation, analyze data, or perform any other system automation task programmatically. Once a user library has been successfully built using Script Editor, its user modules can be executed using the ACS software.

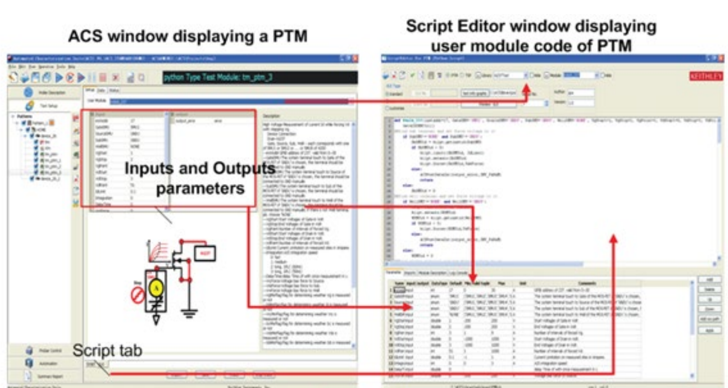
Script Editor provides a simple user interface that helps a novice programmer to effectively enter code, compile a user module, and link (build) a user library. Script Editor also provides user-library management features, including the Open file, Delete, Copy Module and Library, Check, edit, browse, Save, and export functions. Script Editor manages user libraries in a structured manner. You can create your own user libraries to extend the capabilities without requiring a software upgrade from Keithley Instruments.

To execute a Script Editor user module in ACS, import or create a test module as TSP or PTM and connect it to the user module. Once this user module is connected to the test module, the following occurs each time ACS executes the PTM or STM:

- ACS dynamically loads the user module and the appropriate user library.
- ACS passes the user-module parameters, stored in the STM or PTM, to the user module.
- Data generated by the user module is returned to the STM or PTM for interactive analysis.

The following figure illustrates the relationships between user libraries, user modules, STM or PTM, ACS, and Script Editor.

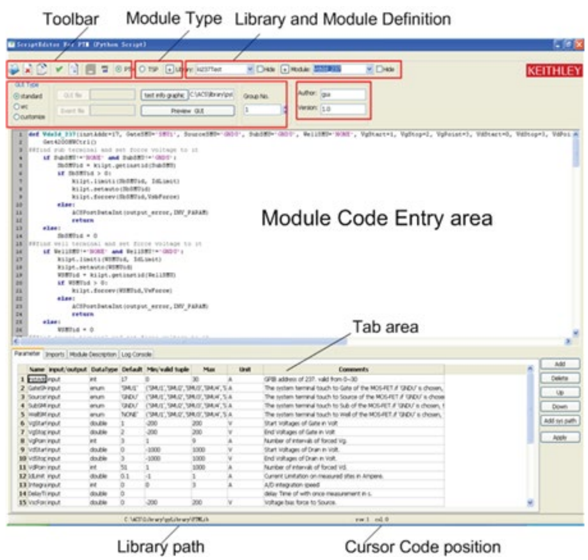
Figure 271: Relationship between ACS test module and Script Editor



Script Editor dialog

The Script Editor user interface is shown in the following figure. It provides all the functions, controls, and user-entry areas needed to create, edit, view, and build a user library and to create, edit, view, and compile a user module.

Figure 272: Script Editor



Understanding the module identification area

The module identification area is at the right of the toolbar and defines the presently open user library and user module. The components of this area are used as follows:

- **Module Type:** Displays the type of the presently open (active) user module. There are two types supported: TSP and PTM. For more information about understanding TSP and PTM, refer to Script Test Module (STM) configuration (on page 5-117) and Python Language Test Module (PTM) configuration (on page 5-152).
- **Library Name:** Displays the name of the presently open (active) user library. To specify a user library, use the Open file function. To delete a library, use the Delete function. To copy a library, use the Add function. To add a new library, use the add a new library function.
- **Module Name:** Displays the name of the presently open user module. This also allows you to create a copy of the presently open user module in the same user library.

NOTE

When naming a user module, remember to conform to case-sensitive Python or TSP programming language naming conventions. Do not duplicate names of existing user modules or user libraries.

- **Library Visible or Library Hidden:** Indicates if the presently open user library is available to ACS. To change the hidden or visible status, select or clear the Hide option in the Module and Library definition area.
- **GUI type:** Select the user interface type for the user module. In ACS, the PTM and STM can include a user interface. There are three types of user interfaces available: Standard, .xrc, and customize. For more information about using a STM with a user interface, refer to STM with the .xrc GUI File (on page 5-127).
- **Version info:** Shows the software author and the version information.

Understanding the module code entry area

The module code-entry area is below the module and library definition area. The module code-entry area is the ACS dialog location where you enter, edit, or view the user-module code. Scroll bars to the right and below the module-code entry area let you move through the code.

The parameters that the module depends on to operate are defined in the Parameters tab area. After you add the parameters (by selecting Add) and enter their usage specifications, select **Apply** to update the module definition area within the indicated changes, as shown in the following figure.

Figure 273: Function definition area

Do not enter the following Python code items in the module code-entry area. The Script Editor enters these at specific locations based on information within the Script Editor dialog:

- Import and global data
- The function prototype

To control internal or external instrumentation, use functions from the Linear Parametric Test Library (LPTLib). Refer to the *ACS Programmer's Manual*, "LPT library reference," for more information about the Series 2600B library.

Do not enter the following TSP code items in the module code-entry area. The Script Editor enters these at specific locations based on information within the Script Editor dialog:

- The function call
- The function prototype

To control internal or external instrumentation, use functions from the Linear Parametric Test Library (LPTLib). Refer to the *ACS Programmer's Manual*, "Series 2600B library and Python library," for more information.

Understanding the tab areas

The following tab areas are accessed by selecting one of the following tabs:

- Parameters
- Imports (PTM only)
- Module Description
- Log Console

Selections in these tab areas are facilitated using menus. The next sections describe the tab areas.

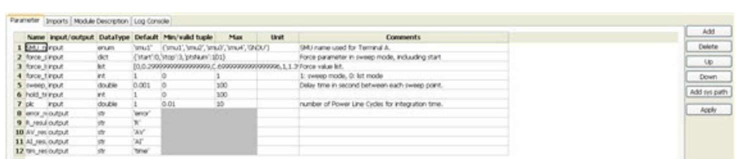
Parameters tab

The Parameters tab area defines and displays the following for each parameter that is included in the user module call:

- Parameter Name.
- Input or output (I/O) data direction.
- Parameter Data Type.
- Default, Min, and Max values, Unit, Comments for the parameter.

The Parameters tab area is near the bottom of the Script Editor main screen, as shown in the following figure.

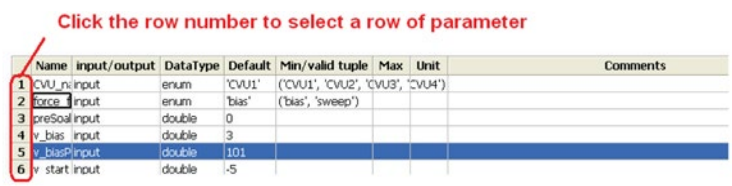
Figure 274: Parameters tab example entries



To add, delete, or apply a parameter, do one of the following:

- To add a parameter, select Add and enter the information as indicated in the field descriptions that follow.
- To delete a parameter, first select the parameter name or any of the adjacent fields; then select **Delete**.
- To move a row of parameter up or down, select the row of the parameters by selecting the row number, then this row will be highlighted. After that, select the Up or Down function.
- To apply changes made in the Parameters tab area, select **Apply**.
- To insert a new external path to Python system path list, select **Add sys path**.

Figure 275: Select a row of parameters to move



Name field

The Name field identifies the parameters that are passed to the user module. For example, these are the same parameters that are specified in the user-module function prototype. The Script Editor constructs from the Parameters tab entries when you select Apply and displays in the module-parameter display area.

Data Type field

The Data Type field specifies the parameter data type. Selecting the arrow at the right of the data type field opens a menu, which lists the following data types:

For TSP only:

enum	enumerable type data
str	string data
float	single precision, floating point data
double	double precision data
int	integer data
list	list type
dict	dictionary type
table	table type

Input and Output field

The input and output (I/O) field defines whether the parameter is an input or output type. Select an I/O field to display the Input and Output selections.

NOTE

Output can be selected only for string data type (str).

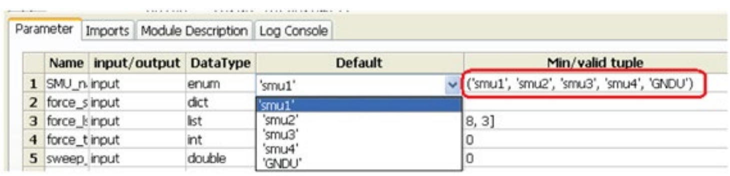
The Default, Min, and Max fields specify the following:

- **Default:** Specifies the default value for input parameter.
- **Min**: Specifies the minimum recommended value for input parameter (except list, dict, table, and str type). When the user module is used in ACS, configuration of the PTM or STM with a parameter value smaller than the minimum value causes ACS to display an out-of-range message. (For a brief explanation of STM and PTM, refer to Script test module (STM) configuration (on page 5-117) and Python Language Test Module (PTM) configuration (on page 5-152)).
- **Max:** Specifies the maximum recommended value for input parameter (except list, dict, table, and str type). When the user module is used in ACS, configuration of the PTM or STM with a parameter value larger than the maximum value causes ACS to display an out-of-range message.

NOTE

When the data type is enumerable (enum) data, the minimum value and the valid tuple field column in the parameter tab is a valid tuple. You must set the minimum value and valid tuple box first so that the default value can be selected from the menu, as shown in the following figure.

Figure 276: Select a default value for enumerable data



Import tab

The Import tab only displays information for the PTM and lists the import files used within the Python user module. This area can be used to add import statements to the presently open user module, as shown in the following figure. This tab can also be used to import the global statement.

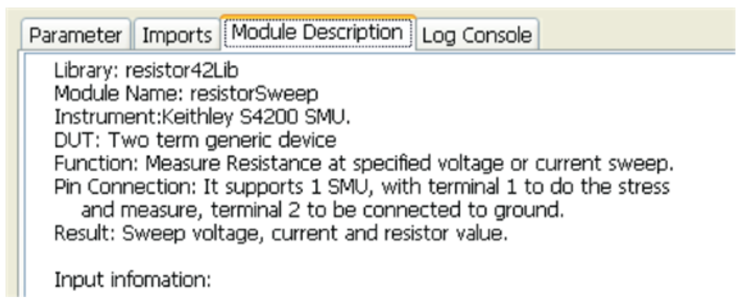
Figure 277: Imports tab

In most cases, Script Editor needs to enter data using the ACS_PostData import in the Import tab.

Module Description tab area

The Module Description tab area, shown in the following figure, allows you to enter descriptive information for the presently open user module. Information entered in this area documents the module to the ACS user and creates the ACS user library help, as shown in the following figure.

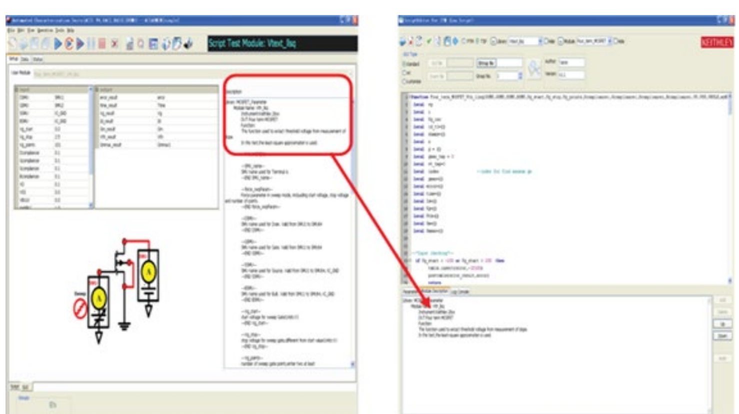
Figure 278: Module Description tab



NOTE

Do not use doc comment designators (---) in the Description tab area. When the user-module code is compiled, Script Editor also evaluates the text in this area. Code comment designators in the Description tab area can be misinterpreted; causing errors, as shown in the following figure.

Figure 279: Description tab in ACS



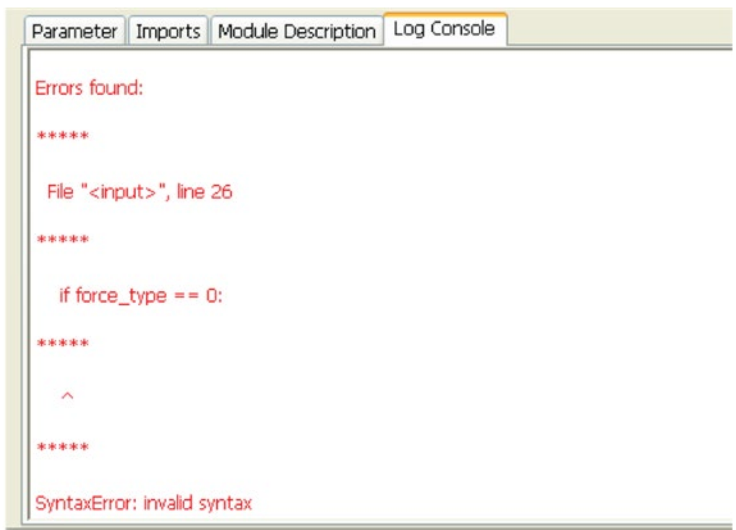
Log Console tab area

The Log Console tab displays any error syntax messages that are generated during a code check operation.

When you select the check function on the toolbar, a message is displayed in the log console tab. When an error displays in the Log Console tab, Script Editor shows the line of code where the error occurred or the next line, depending on how the compiler caught the error, as shown in the following figure. If no errors are found, the Build tab area displays the following details:

- After a check — Checking completed, No error found!

Figure 280: Log Console tab



Understanding the status bar

The Status bar at the bottom of the Script Editor dialog displays the path where the pylLibrary is located and indicates the row and column code entry of the cursor location. For example, the cursor is in row 42 and column 4 in the code entry, as shown in the following figure.

Figure 281: Status bar

Understanding the toolbar

NOTE

Most of the functions apply to the Script Editor user interface above the tab. Refer to Understanding the tab areas (on page 5-201) for more information.

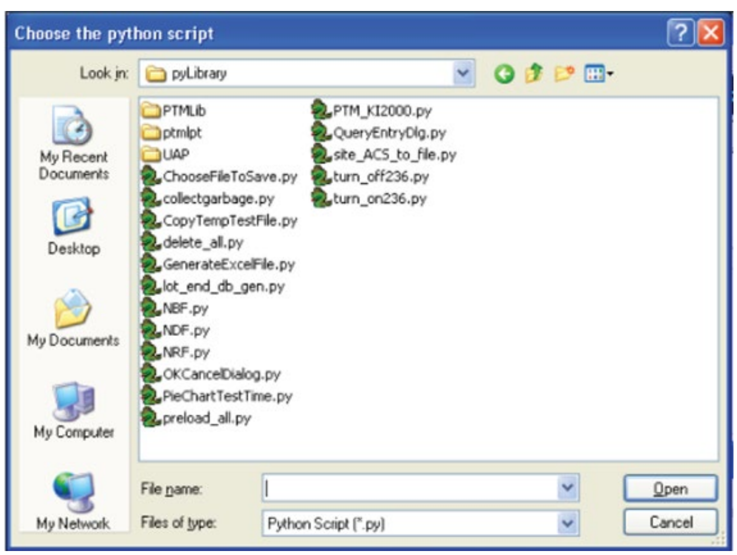
The toolbar is shown in the following figure.

Figure 282: Script Editor toolbar

The toolbar functions:

Import Library: Opens an existing user library and displays the first module that is listed for that library. Selecting Open Module displays the Choose a File dialog, as shown in the following figure, where you can select an existing user module. Select **Open** to see the selected module in place of the currently open module.

Figure 283: Select a .py file

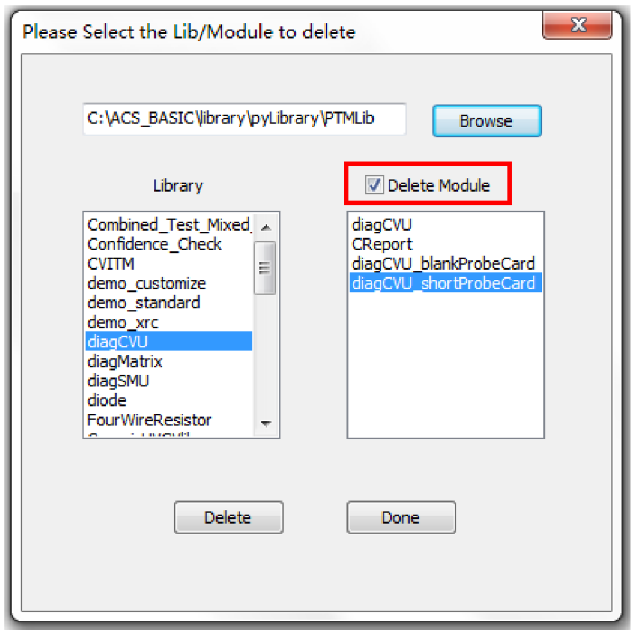


Delete: Deletes an existing user library or a user module from the indicated user library. Selecting **Delete** displays the "Please Select the Lib/Module to delete" dialog, as shown in the following figure. To delete a selected library and all its contents, select **Delete**. To delete a particular test module from the selected library, select **Delete Module**, then select **Delete**.

NOTE

Be careful when using the delete option. Deleting a library deletes all the files in it.

Figure 284: Choose a module to delete

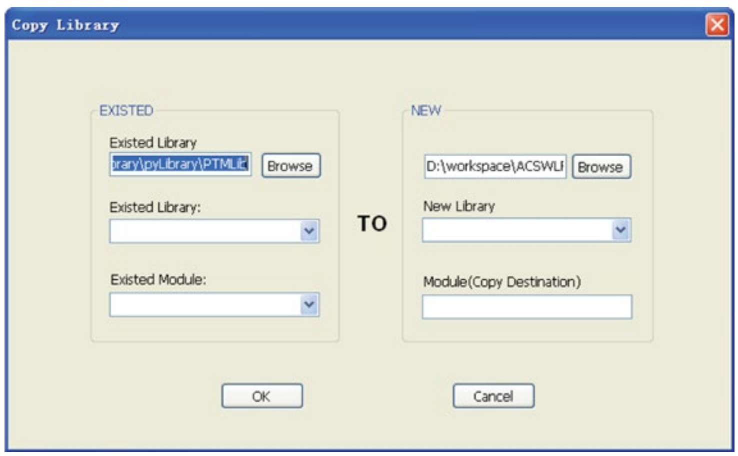


Copy Library and Module: Creates a copy of any existing library and module. Selecting Copy Module and Library displays the Copy Library dialog, as shown in the following figure. Select the user library and module to copy. Enter the new library and module name in the New section. You must enter a unique user-module name. Select **OK**.

NOTE

If you enter an existing library name to the New section, the module will be copied from one existing library into another existing library. If you enter a new library name, the module is copied into the new library. The module name in the existing area can be empty (all the modules in the source library are copied to the destination library).

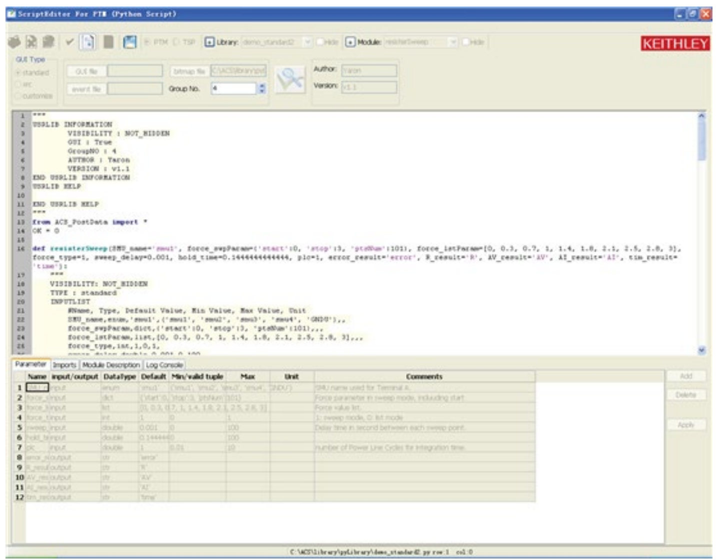
Figure 285: Copy library and module



Check: When selected, checks the source files for syntax errors (including the Description tab) in the module and, if successful, compiles them into object files.

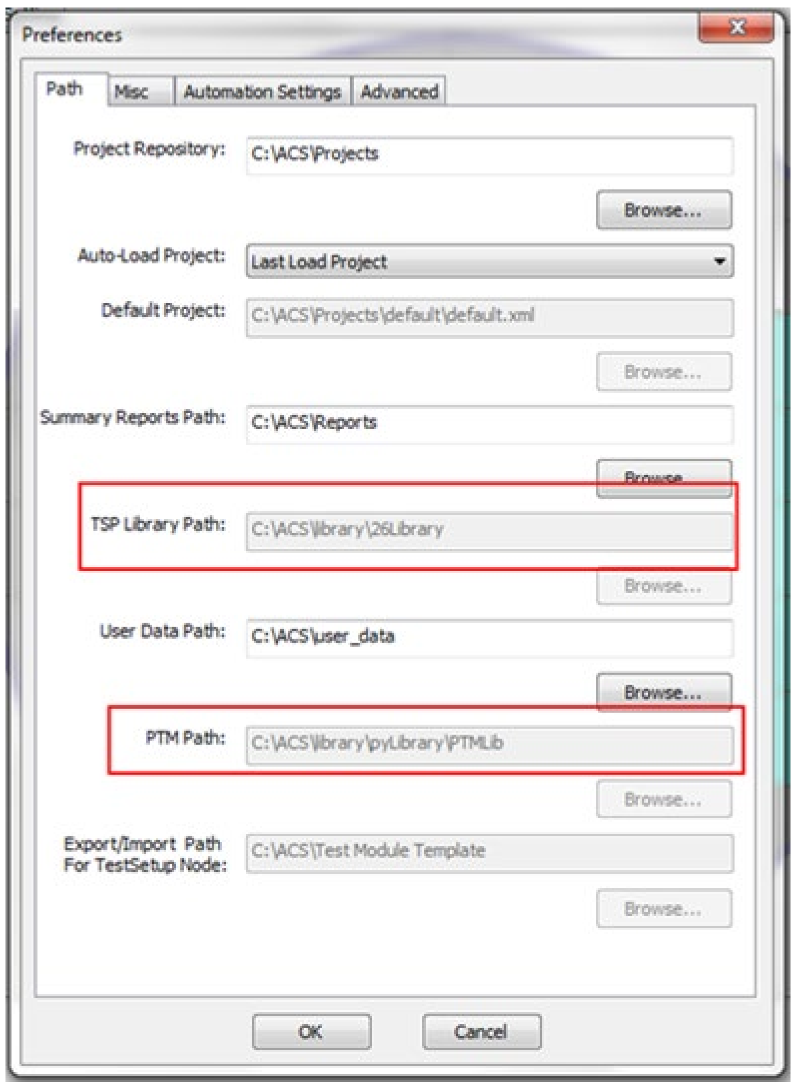
Browse or Edit the entire file: When selected, the Code Entry area changes to whole file states. You can Browse or Edit the entire file in this case, as shown in the following figure. When you import a TSP or Python source code that is not generated by the Script Editor, it may launch this Browse entire file, dialog directly if the file format does not match the expectations of the Script Editor.

Figure 286: Browse the entire file



Save Module: Saves the presently open user module source code. The save path is `\\ACS\library\26Library\TSPLib` for TSP and `\\ACS\library\pyLibrary\PTMLib` for a PTM. The path is assigned in Preferences settings dialog, as shown in the following figure. For more information you can go to Tools menu and find Preferences in the drop-down or go to the Tools menu (on page 3-18) topic.

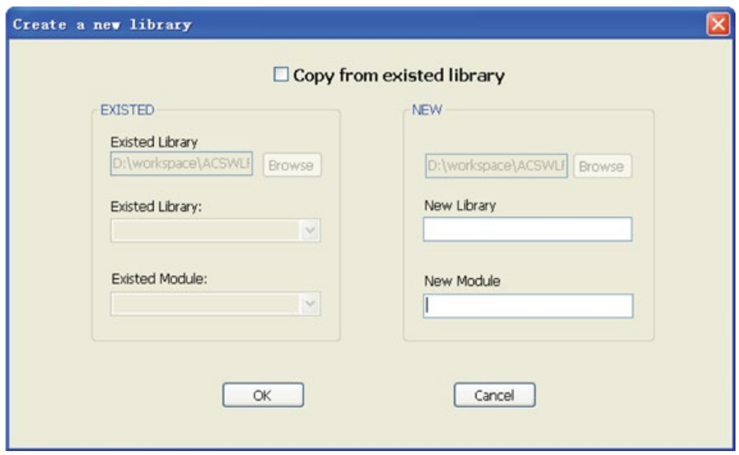
Figure 287: Set the TSP and PTM path



Show the .xrc GUI of test module: Select this function to display the .xrc GUI builder tool. For more information about how to use this GUI builder, refer to Create a TSP GUI file (on page 5-129).

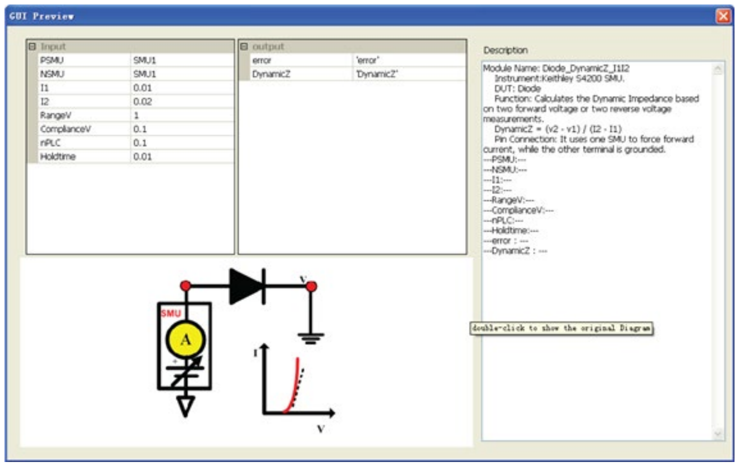
Add a library or module: Create a new library or user module. Select **Create a new library**. Enter a new library name and a new module name in the corresponding text boxes. The new name must not duplicate the name of any existing user module or user library in the user library collection. After entering the name, select **OK**. A new library or a new module will be added to the path. You can also select the browse button to choose an alternative path to save the new library, as shown in the following figure.

Figure 288: Add library and module



Preview GUI: After a `.bmp` file has been chosen and the library modules defined, select **GUI Preview**. The dialog opens, as shown in the following figure. so you can view the STM or PTM user interface. The errors in the user interface files can only be selected by using the Preview button.

Figure 289: GUI preview



Script Editor tutorials

This section includes two tutorials. Each tutorial provides step-by-step instructions to accomplish common tasks using Script Editor.

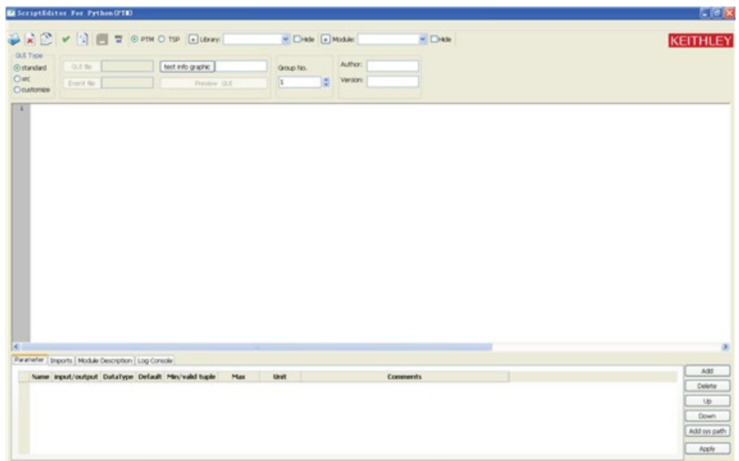
Tutorial 1: Create and run a new TSP script

Script Editor is a tool that facilitates the development of user libraries. Each user library is comprised of one or more user modules, and each user module is created using the TSP or Python programming language.

To create a new TSP script:

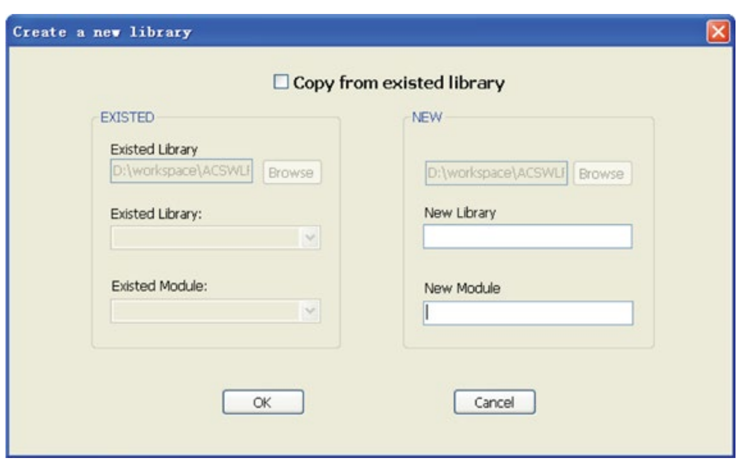
1. In ACS, select **Tools > Script Editor**. A blank Script Editor dialog opens named Script Editor: Module NoName Library NoName, as shown in the following figure.

Figure 290: Blank Script Editor



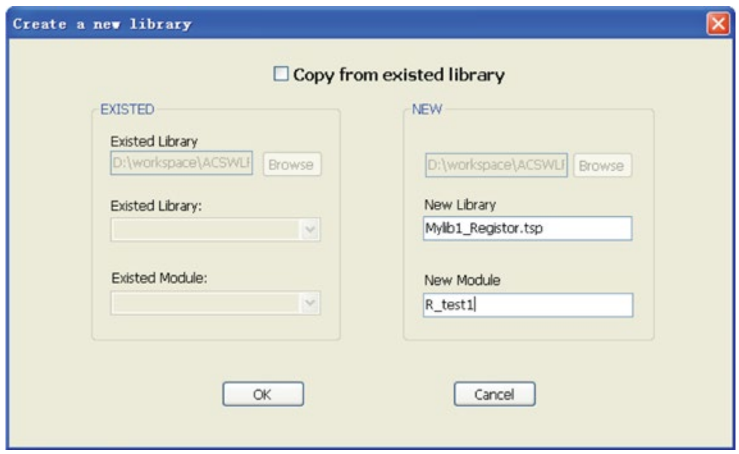
2. Select the **TSP** type in Module Type area.
3. Create a new user library: Select **Add a Library**. The Copy Library dialog opens, as shown in the following figure.

Figure 291: Create a new library



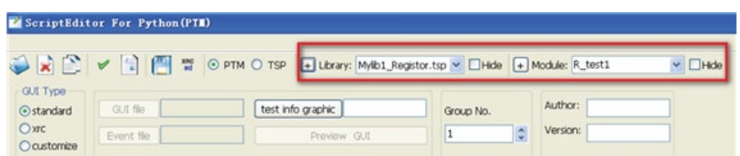
4. In the New Library text box, enter the new user library name. The library name must have an extension of `.py` or `.tsp`. For this tutorial, enter `Mylib1_Resistor.tsp` as the new user library name.
5. For this tutorial, enter `R_test1` as the new module name, as shown in the following figure.

Figure 292: Enter new library and module name



6. Select **OK**. The new library and module names appear in the Script Editor dialog, as shown in the following figure.

Figure 293: New Library name in Script Editor



To use the user interface for TSP:

1. Choose the standard option from the user interface Type items. When using the .xrc user interface, you can indicate the user interface file and the Event file for the library. For more information about the STM with .xrc user interface, refer to STM with the .xrc GUI File (on page 5-127).
2. Choose a test info graphic for the TSP.
3. Set the group number for the test.
4. Enter the Author and Version in the edit field.

Enter the required parameters for the code as follows:

1. Select Parameters (if the Parameters tab area is not displayed).
2. Select Add at the right side of the Parameters tab area.
3. Under Name, enter the first parameter name (or accept the default). For the `R_test1` user module, replace the default name with `RSMU1`.
4. Under the input and output, specify whether the first parameter is an input or output parameter.

NOTE

For an output parameter, only the string data type is acceptable.

5. Under Data Type, enter the data type for the first parameter. Select the enum type for the first parameter
6. Under Default, Minimum, Maximum, and Units, enter default, minimum, maximum and unit values for the parameter.
7. If the data type is enum, you should set the minimum and valid tuple value first:
   - Right-click the Min area. The Sequence Editor dialog opens.
   - Select Append.
   - Input the items in the **Add a new item** dialog.
   - Select **OK**. The enum data is set. These values can be selected in default.
8. If the data type is a list, set the default value as follows:
   - Right-click the default value (the Sequence Editor dialog opens).
   - Add the item as indicated.
   - Select **OK**. The value appears in default box with this format [a,b,c] (see the next two figures).

Figure 294: Sequence Editor for setting enumerable parameter

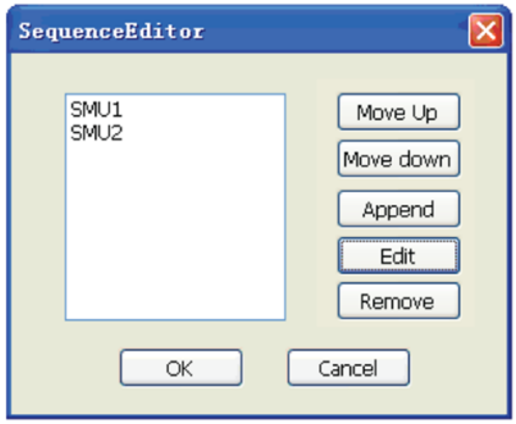
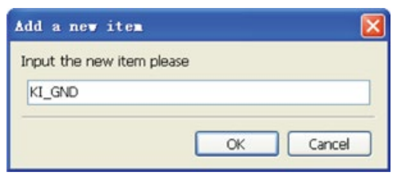
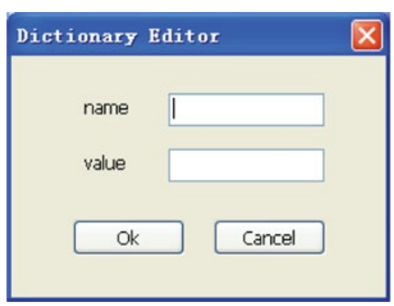


Figure 295: Add a new item for setting enumerable parameter



9. The dictionary type (dict type) is mostly used in PTM:
   - Right-click the default value to display the **Sequence Editor**.
   - Select **Append** to open the Dictionary Editor dialog.
   - Input the name and value in the **Dictionary Editor**, as shown in the following figure. The value is formatted as {a:1,b:2,c:3}.
10. For the table type (mostly used in TSP), add the item in the table.

Figure 296: Add name and value for Dictionary Editor

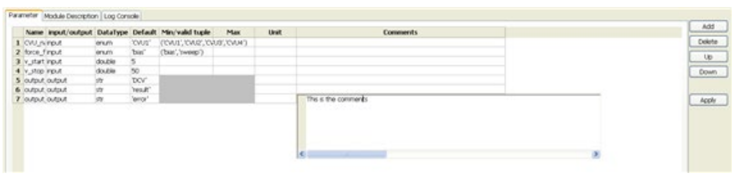


11. For the `mytsp1` user module, enter SMU1, SMU2, and KI_GND for RSMU1 enum data.
12. After the minimum and valid tuple value is set, select the default value. In the following figure, SMU1 for RSMU1 is selected.
13. Enter the comments for the parameter. Select the comment area to display in the edit field. Enter the parameter description.

NOTE

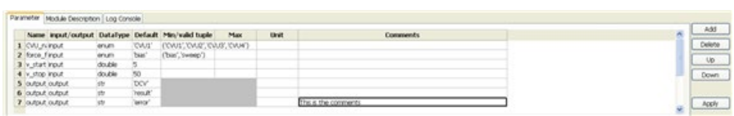
Do not use doc comment designators (`---`) in the Description tab area. When the user-module code is compiled, the Script Editor also evaluates the text in this area. Code comment designators in the Description tab area can be misinterpreted, causing errors.

Figure 297: Enter comments



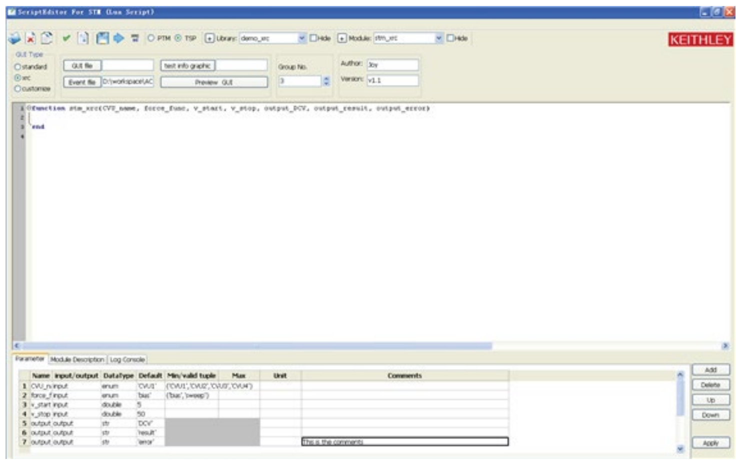
14. Repeat the steps to add a new item for all additional input and output parameters for the user module that you are creating. For the R_test1 module, add eight parameters, as shown in the following figure.

Figure 298: Parameter tab



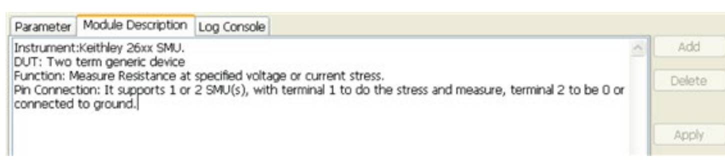
15. Select **Apply**. In the module-parameter display area, the function prototype now includes the declared parameters.
16. Module R_test1 (RSMU1, RSMU2, forcevalue, myLIMIT, myNPLC, testdelay, error, Rvalue).

Figure 299: Script Editor after entering and applying parameters



17. Enter the module description:
    - Select the Description tab at the bottom of the dialog. The Description tab area opens, as shown in the following figure.
    - Enter any text needed to adequately document the user module to the Script Editor user. The user does not see the comments that you include with the code.

Figure 300: Description tab



NOTE

Do not use doc comment designators (---) in the Description tab area. When the user-module code is compiled, Script Editor also evaluates the text in this area. Code comment designators in the Description tab area can be misinterpreted, causing errors.

18. Enter the TSP user module code for the R_test1 shown below. Refer to the *LPT library reference* in the *ACS Programmer's Manual (ACS-907-01)* for a complete list of supported input and output SMU commands.
19. For the `R_test1` user module, enter the sample code listed below.

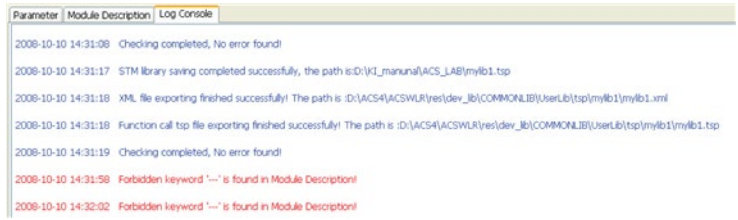
```
local v_value = {}
local i_value = {}
local error ={}
setmode(RSMU1, KI_INTGPLC, myNPLC)
limiti(RSMU1, myLIMIT)
forcev(RSMU1, forcevalue)

if RSMU2 ~= KI_GND then
   forcev(RSMU2, 0)
end --if

delay(testdelay)
intgv(RSMU1, v_value)
intgi(RSMU1, i_value)

Rvalue[1] = v_value[1]/i_value[1]
posttable("Rvalue", Rvalue)
table.insert(error, 0)
posttable("error",error)

devint()
end
```

For TSP, the Script Editor checks the user module description.

To compile the user module:

1. Select **Check** table. After you compile a user module, the Log Console tab area displays either a confirmation that the module compiled successfully or displays one or more compile-error messages. For TSP, the Script Editor checks the user module description part for the usage of the ”---”.
2. In the toolbar, select **Check**.
3. The user-module source-code file is compiled.
4. The Log console message tab indicates the compilation progress and, if problems are encountered, displays error messages. For example, when you check the `R_test1` user module, you see the Log console message tab, as shown in the following figure.

Figure 301: Log Console message tab

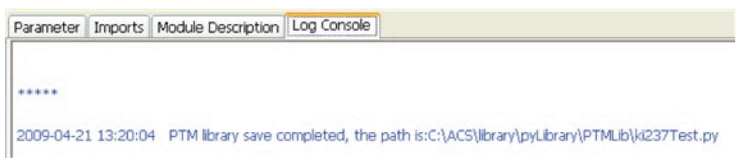


5. Check until no errors exist.

To save the user module:

1. Select **Save**. All information of source code saves into the path `C:\ACS\library\26Library\TSPLib`.
2. For the example in the following figure, after selecting the save function, the following files are saved.

Figure 302: Log Console path saved



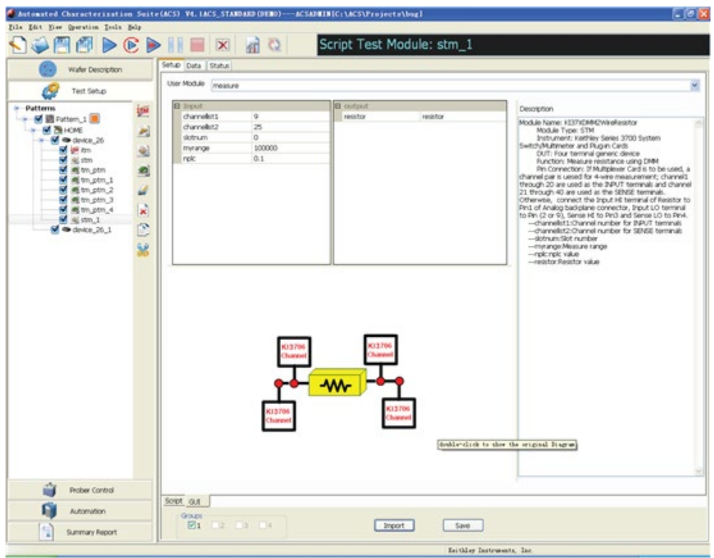
- **Source code file (.tsp).** The default save path is C:\ACS\library\26Library\TSPLib.
- **.Bmp file and .xrc file:** The default save path is C:\ACS\library\26Library\TSPLib\xrc.
- **Events file:** The default save path is C:\ACS\library\26Library\TSPLib\event.

The new standard TSP script is created and saved in the path. If you export this module from the Script Editor, you can open and run this module in ACS.

To run the TSP code:

1. Add a new STM to configuration navigator.
2. Select **Run** then choose a Python test module.
3. Choose a module from the TSPLib and it opens, as shown in the following figure. Accept the default parameters. You can experiment after you have established that the user module executes correctly.

Figure 303: UserLib module opening



4. Save the STM and the project by selecting the Save All icon at the top of ACS or by selecting Save All in the File menu.
5. Execute the STM by selecting the green triangular Run Test/Plan icon at the top of the screen or by selecting Run in the Run menu.
6. If the user module that you created generates data, check the execution results in the STM Data worksheet. To view the Data worksheet, select the Data tab at the top of the Definition document.

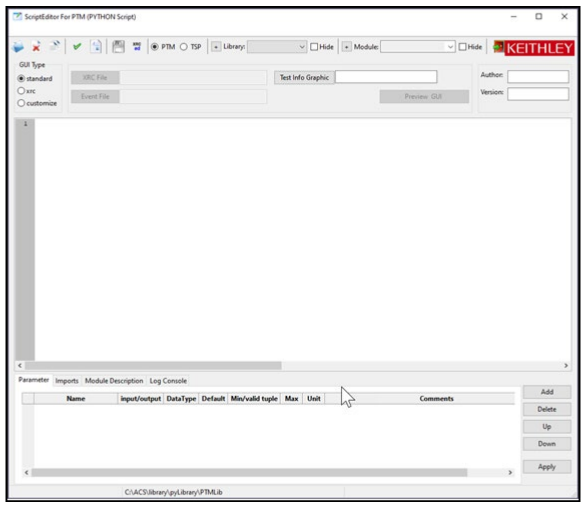
Tutorial 2: Create and run a new PTM

This section provides a tutorial to help you to create a new PTM. For more information about PTMs, refer to Python language Test Module (PTM) configuration (on page 5-152).

To create a new PTM:

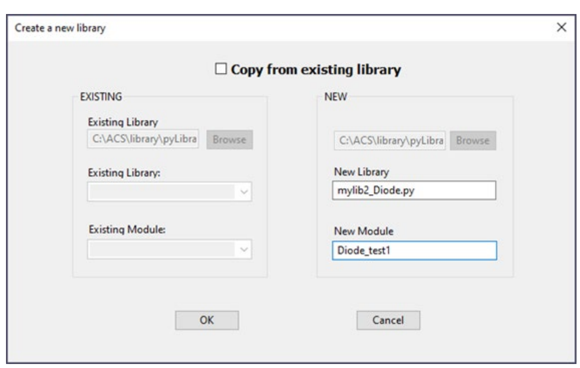
1. In ACS, select **Script Editor** in the **Tools** menu. A blank Script Editor opens, as shown in the following figure.

Figure 304: Blank Script Editor



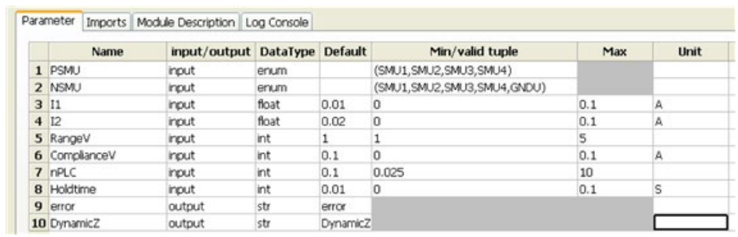
2. Select **+** to the left of the Library list. The Create a New Library dialog opens.
3. In New Library, enter `mylib2_Diode.py` as the new user library name.
4. In New Module, enter **Diode_test1** as the new module name, as shown in the following figure.

Figure 305: Name a new PTM library and module



5. Select **OK**. The new library and module names display in the Script Editor.
6. To the left of the Library name, select **PTM**.
7. For user interface Type, select **standard**.
8. Select **Test Info Graphic** to choose a bitmap.
9. Add the **Author** and **Version**.
10. Select the **Parameter** tab and enter the parameter information, as shown in the following figure.

Figure 306: Enter parameter for PTM



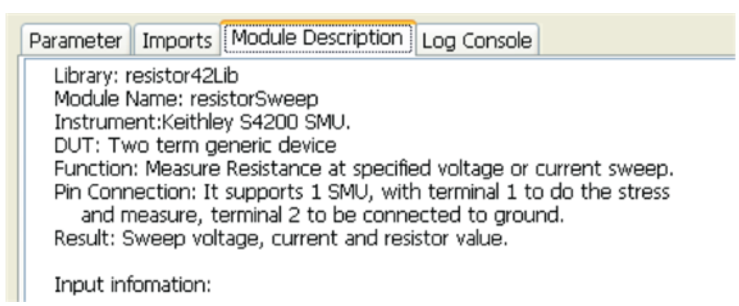
	Name	input/output	DataType	Default	Min/valid tuple	Max	Unit
1	PSMU	input	enum		(SMU1,SMU2,SMU3,SMU4)		
2	NSMU	input	enum		(SMU1,SMU2,SMU3,SMU4,GNDU)		
3	I1	input	float	0.01	0	0.1	A
4	I2	input	float	0.02	0	0.1	A
5	RangeV	input	int	1	1	5	
6	ComplianceV	input	int	0.1	0	0.1	A
7	nPLC	input	int	0.1	0.025	10	
8	Holdtime	input	int	0.01	0	0.1	S
9	error	output	str	error			
10	DynamicZ	output	str	DynamicZ			

11. Select the **Imports** tab. A PTM can import other libraries to use in the present library. Enter any additional imports files that are needed by the user module.

Figure 307: Imports tab

12. Select **Module Description**.
13. Enter any text needed to adequately document the user module.

Figure 308: Module Description tab



NOTE

Do not use doc comment designators (---) in the Module Description. When the user-module code is compiled, Script Editor also evaluates the text in this area. Code comment designators in the Module Description can be misinterpreted, causing errors.

14. Select **Apply**.
15. Enter the Python code for Diode_test1 as listed below.

```
#Map the SMUs with DUT terminals
PSMUId = lpt.getinstid(PSMU)
if not PSMUId:
   error = -10000
   post.ACSPostDataInt("error",error)
   return error, DynamicZ
NSMUId = lpt.getinstid(NSMU)
if not NSMUId:
   error = -10000
   post.ACSPostDataInt("error",error)
   return error, DynamicZ

#Set range and compliance
lpt.limitv(PSMUId, ComplianceV)
if NSMUId != 4096:
   lpt.limitv(NSMUId, ComplianceV)
if RangeV == 1:
   lpt.setauto(PSMUId)
elif RangeV == 2:
   lpt.lorangev(PSMUId, 0.2)
elif RangeV == 3:
   lpt.lorangev(PSMUId, 2)
elif RangeV == 4:
   lpt.lorangev(PSMUId, 20)
elif RangeV == 5:
   lpt.lorangev(PSMUId, 200)
else:
   lorangev(PSMUId, 2)

#Set instrument config
lpt.setmode(PSMUId, lpt.KI_INTGPLC,nPLC)
htime = int(Holdtime * 1000)

#Set stress
if NSMUId != 4096:
   lpt.forcev(NSMUId, 0)
lpt.forcei(PSMUId, I1)
lpt.delay(htime)
V1 = lpt.intgv(PSMUId)

lpt.forcei(PSMUId, I2)
lpt.delay(htime)
V2 = lpt.intgv(PSMUId)
```

```
#check compliance
cstatus = lpt.getstatus(PSMUId, KI_COMPLNC)
if cstatus == 2:
   error = KI_RANGE_COMPLIANCE
   post.ACSPostDataInt("error",error)
   return error, DynamicZ
if cstatus == 4:
   error = KI_COMPLIANCE
   post.ACSPostDataInt("error",error)
   return error, DynamicZ

#test complete
lpt.devclr()
Idelta = I1 - I2

if  Idelta == 0:
   Idelta = 1e-37

DynamicZ = (V1 -V2) / Idelta

post.ACSPostDataDouble("DynamicZ", DynamicZ)
post.ACSPostDataDouble("I1", I1)
post.ACSPostDataDouble("I2", I2)
post.ACSPostDataDouble("V1", V1)
post.ACSPostDataDouble("V2", V2)
post.ACSPostDataInt("error",error)
return error, DynamicZ
```

To compile the user module:

1. Select the Check table on the toolbar. After you compile a user module, the Log Console tab area displays either a confirmation that the module compiled successfully or displays one or more compile-error messages. For TSP, the Script Editor checks the user module description part for the usage of the ”---”.
2. In the toolbar, select **Check**. The following occurs:
   - The user-module source-code file is compiled.
   - The Log console message tab indicates the compilation progress and, if problems are encountered, displays error messages. For example, when you check the R_test1 user module, you see following figure the Log console message tab.
3. Check until no errors exist.
4. Select **Save**.

Figure 309: Save information in Log Console

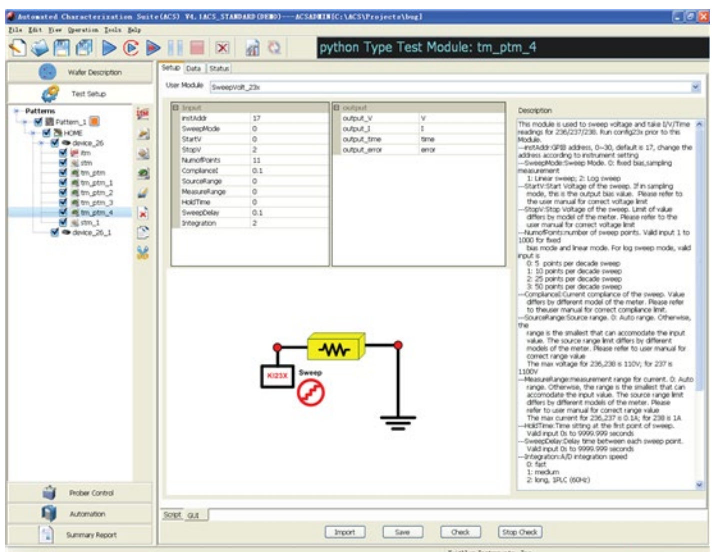
After selecting the save function the following files are saved:

- The source code file (`.py` file). The default saving path is `C:\ACS\library\pyLibrary\PTMLib`
- The `.bmp` file. The default saving path is `C:\ACS\library\pyLibrary\PTMLib\xrc`
- The event file. The default saving path is `C:\ACS\library\pyLibrary\PTMLib\event`

The new standard PTM is created and saved. After you export it, you can open and run this module in ACS.

The new standard PTM script is created and saved in the path. You can open and run this module in ACS, as shown in the following figure.

Figure 310: Import and open a PTM from PTMLib



To open and run the PTM:

1. Add a new PTM to the configuration navigator in the Test Setup area of ACS.
2. Select **Import**. The PTMLib dialog opens.
3. Choose a PTM. The PTM dialog opens.
4. Save the PTM.
5. Execute the PTM by selecting the Run Test/Plan icon at the top of the screen.
6. If the user module that you created generates data, check the execution results in the PTM Data worksheet. To view the Data worksheet, select the Data tab at the top of the Definition document.

Executing a project plan

You can modify the test patterns, subsites, devices, and test modules by using copy, paste, cut, and rename general editing commands. After the test plan has been built, you can execute an entire project plan or individual parts of a project plan. This section describes the following topics:

- Enabling tests (configuration navigator)
- Project pattern execution
- Individual subsite plans execution
- Individual device plans execution
- Individual execution of tests

Executing a project pattern

Executing an entire project pattern executes all its components (initialization pattern, subsite plans, device plans, ITMs, STMs, and CTMs) in the order in which they appear in the Test Setup panel. Before executing a plan, manually check whether the physical connections match the test plan connections correctly. If they do not match multiple times, manually check them until the specified SMU is matched correctly.

Each component of the project plan has a corresponding checkbox. Selecting the component enables the test plan, while clearing the checkbox disables it.

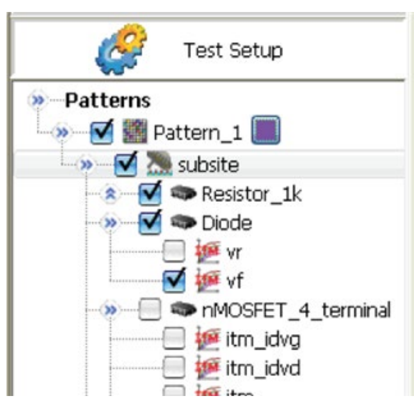
To execute an open project plan:

1. Select **Pattern** at the top of the Test Setup panel.
2. Place the probe at one site.
3. Enable the test by selecting the corresponding checkbox or disable the test by clearing it, as shown in the following figure.

NOTE

When running a project pattern, the prober controlling command is activated. Make sure the prober is correctly connected, or set to fake, to avoid errors.

Figure 311: Select a Pattern (indicated with checkmark in box)



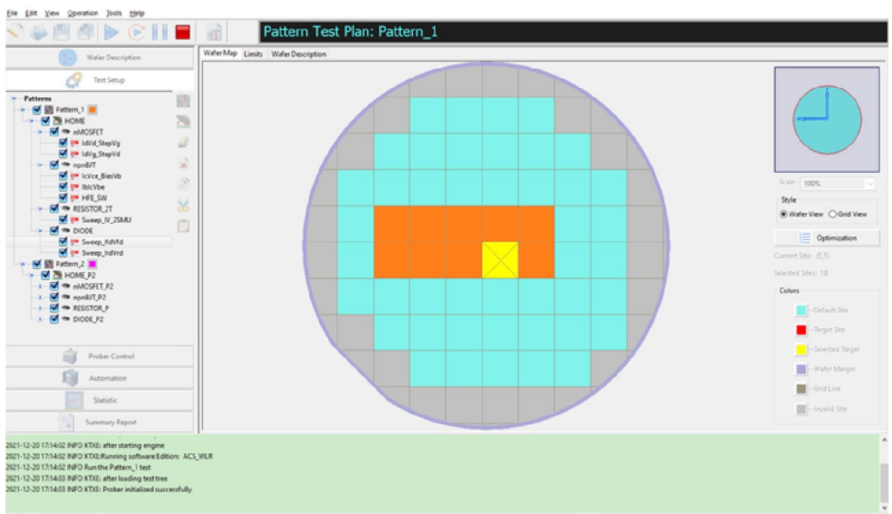
4. If the project plan has not been saved, save the project plan by selecting the Save function in the ACS toolbar.
5. Start execution by selecting the run test function in the toolbar (part A), as shown in the following figure.

Figure 312: Run Project Plan Test

6. The run test function becomes gray (part B), the project pattern executes, and the square abort test function is enabled for the duration of the execution, as shown above. You can select abort test at any time during execution to stop it.

During execution, the Definition tab of the ITM work area, and the Script tab of the CTM and STM work areas change color, and during execution, the wafer map tab of pattern change color and the configuration navigator, as shown in the following figure. These work areas cannot be edited while a test is running.

Figure 313: A running pattern



NOTE

Each time you execute a test or test sequence using the Run function, the data from each test is inserted into its own Data worksheet. Each new run updates the worksheet.

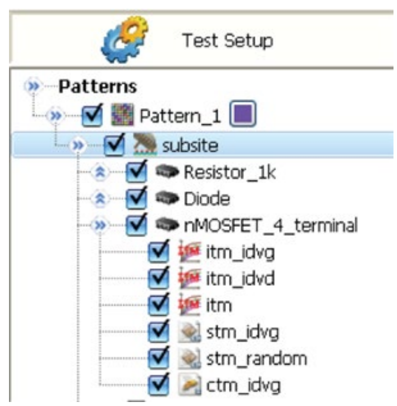
Executing individual subsite plans

Running a subsite plan executes only the components assigned to it (all its device plans, ITMs, STMs, and CTMs) in the order that they appear in the Test Setup panel.

To execute a subsite plan in an open project plan:

1. If the test system is connected to a probe, place the probe at the site that contains the Subsite to be evaluated.
2. In the Test Setup panel, select the name of the subsite plan to be executed, as shown in the following figure.

Figure 314: Select subsite plan to execute



3. Enable a test by selecting it or disable a test by clearing the corresponding checkmark.
4. If the project plan has not been saved, save the project plan by selecting save.
5. Start execution by selecting the triangular run test function in the toolbar (part A), as shown in the following figure. The run test function becomes gray (part B), the Subsite executes, and the square Abort Test icon illuminates for the duration of the test.

Figure 315: Run Project Plan Test

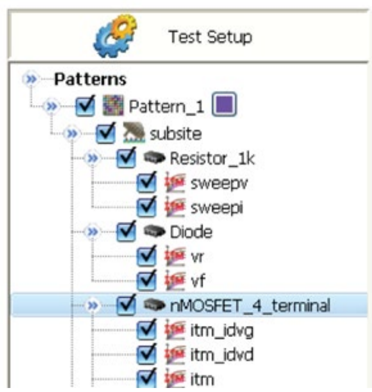
Executing individual device plans

You can execute an individual device plan on the components assigned to the specified device plan (all its ITMs and STMs), in the order in which they appear in the test setup panel.

To execute an individual device plan:

1. If the test system is connected to a probe, place the probe at the site that contains the device to be evaluated.
2. In the Test Setup panel, select the name of the device plan to be executed, as shown in the following figure.

Figure 316: Select the device to execute



3. Enable a test by selecting it or disable a test by clearing the corresponding check box.
4. If the project plan has not been saved, save the project plan by selecting save.
5. Start execution by selecting the triangular run test function in the toolbar (part A), as shown in the following figure. The run test toolbar becomes gray (part B), the Device executes, and the square Abort Test icon illuminates for the duration of the test.

Figure 317: Run Project Plan Test

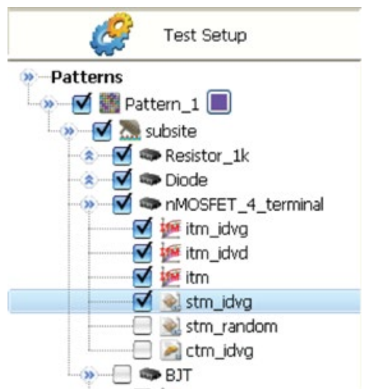
Executing individual tests

Executing an individual ITM or STM plan runs that specific test.

To execute an ITM or STM from an open project plan:

1. If the test system is connected to a probe, place the probe at the site that contains the ITM, STM, or CTM to be evaluated.
2. In the Test Setup panel, double-click the name of the ITM, CTM, or STM to be executed, as shown in the following figure.

Figure 318: Select the STM to be executed



3. If the project plan has not been saved, save the project plan by selecting save.
4. Start execution by selecting the triangular run test function (part A) in the toolbar, as shown below. The run test toolbar function becomes gray (part B), the ITM or STM executes, and the square Abort Test icon illuminates for the duration of the test, as shown in the following figure.

Figure 319: Run Project Plan Test

Data plot

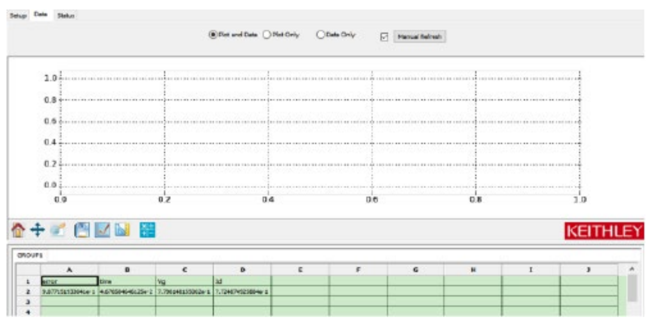
This section describes the variations available with plotting data.

Open a graph sheet

The Graph sheet is integrated in the same page with the Data sheet. There are two ways to open a Graph sheet:

- When setting up an ITM, STM, PTM, or CTM, you can switch to the Graph sheet of the current test by selecting the Data tab. Results of the current test are displayed, and graphs can be drawn according to the results. The following figure shows a graph sheet of a test with a data sheet at the bottom. For STM and PTM, you can refresh the graph by selecting the Manual Refresh button.

Figure 320: An unconfigured graph sheet



- Select **Tool**. Choose the Offline Data Plotting item and then select a blank Data Plotting sheet like the one shown above. Note that the data plotting sheet will have the same style except for the additional two functions that are over the top left corner. Data and graphs on this sheet are not limited to the current test or the current project. Select Import to import data, then select **Exit**.

Controlling graph properties

Items in the Graph Settings menu and the Graph toolbar control the properties of the graph.

Toolbar buttons

The toolbar is placed between the graph section and the data section.

The properties graph toolbar contains the following functions:

- **Zoom (in):** Enlarges a small, selected part of the graph.
- **Home:** Used to go back to the original size before zooming.
- **Pan:** Permits dragging along the graph.
- **Save:** Saves the graph sheet as a separate file; .png, .ps, .eps, and .svg are supported formats. Detailed instructions on these formats are discussed later in this section, Accessing the graph settings menu (on page 5-236).
- **Plot:** Draws a graph out of the data selected.
- **Plot Auto Scale:** Autoscales all the graphs on the graph sheet.
- **Formulator:** Performs simple or complex data calculations on test data and on the results of other Formulator calculations. Detailed instruction about using the Formulator data plot (on page 5-235) is discussed later in this section.

Save a data plot graph

When the Save function is selected, a save dialog displays. You can save the plot or test data to a `.png`, `.ps`, `.eps`, `.svg`, or `.csv` file format.

Formulator data plot

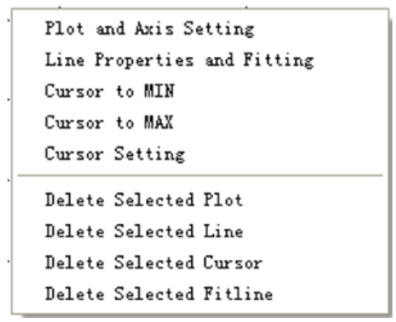
The Formulator, accessible from the Data tab, allows you to perform simple and complex data calculations on test data and on the results of other Formulator calculations. The Formulator provides a variety of computational functions, common mathematical operators, and common constants (see Formulator functions (on page 5-255) for more information).

A formula created by the Formulator is an equation composed of a series of functions, operators, constants, and arguments.

Accessing the graph settings menu

To view the Graph Settings menu, right-click any portion of the graph, as shown in the following figure.

Figure 321: Graph Settings Menu



The Graph Settings Menu functions are summarized below:

- **Reset original view:** Returns to the original size after zooming or panning.
- **Forward navigation view:** Moves forward one step at-a-time when zooming or panning.
- **Back navigation view:** Goes backward one step at-a-time after zooming or panning.
- **Zoom to rectangle:** Zooms in on the section of the graph that is within the drawn rectangle.
- **Plot pan:** Moves the graph by dragging.
- **Plot and Axis Setting:** Sets general properties of the selected graph and parameters of its axis. Many sub-functions are provided. For more information, refer to Define graph properties (on page 5-241).
- **Cursor Setting:** Adds a cursor to a graph and sets the cursor properties. For more information, refer to Define a cursor (on page 5-245).
- **Line Properties and Fitting:** Sets the properties of a selected line and plots data if the cursors are set. For more information, refer to Edit line properties (on page 5-246).
- **Cursor to MIN:** Makes the selected cursor find the minimum point of attached curves.
- **Cursor to MAX:** Makes the selected cursor find the maximum point of attached curves.
- **Delete Selected Plot:** The plot on which the menu is shown is deleted from the graph sheet and the other plots will be resized.

NOTE

When performing the Delete Selected Line, Delete Selected Cursor, and Delete Selected Fitline actions, do not move the mouse from the selected item during manipulation.

- **Delete Selected Line:** Removes a selected line from the plot. To delete a specific line, select the line and right-click, then choose Delete Selected Line from the menu.
- **Delete Selected Cursor:** Deletes the specified cursor. To delete a cursor, first select the cursor to be deleted, right-click, then choose the Delete Selected Cursor menu item.
- **Delete Selected Fitline:** Deletes the specified Fitline cursor. To delete a Fitline, first select the Fitline (if any) to be deleted, right-click, then choose the Delete Selected Fitline menu item.

Define a graph and plotting

There are two steps you need to do when defining a graph and plotting:

1. Define the data to be graphed.
2. Plot according to the data.

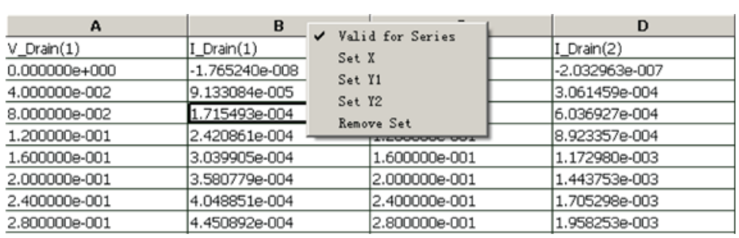
Defining the graph data

There are two steps in defining data to be graphed:

1. Setting the axis to data.
2. Selecting the data to plot.

Right-click the title of any column in the Data sheet to display the Graph Definition Menu, as shown in the following figure.

Figure 322: Graph definition menu



- **Valid for Series:** The Valid for Series command is used for a multiple-step test and there are a series of results, as shown above. If the command is selected, the following settings to any column in the series are valid for all other columns in the same series. For example, if column B (I_Drain(1)) is set as X, column D (I_Drain(2)) will be set automatically as X.
- **Set X:** The selected column is set as the X Axis on the graph.
- **Set Y1:** The selected column is set as the Y Axis on left side of the graph.
- **Set Y2:** The selected column is set as the Y Axis on right side of the graph.

To draw a graph, at least one X and one Y1 or Y2 should be set.

NOTE

ACS supports multiple plotting using one data sheet. This allows different columns to be plotted to different graphs.

After an axis is set, choose data to be plotted on one graph. To choose data, select the title of a column.

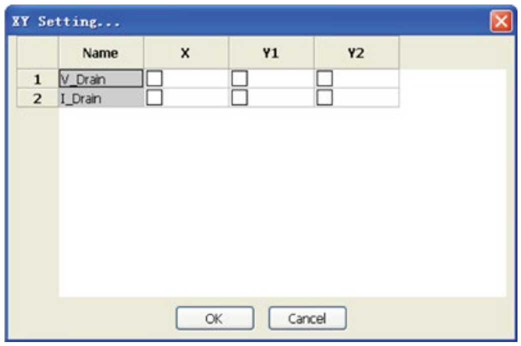
To choose more than one column, drag the mouse over the column or select the column. At least one X and one Y1 or Y2 must be selected. The following figure shows the results of choosing data.

Figure 323: Result of data selection

	A(X)	B(Y1)	C(X)	D(Y1)
1	V_Collector(1)	I_Collector(1)	V_Collector(2)	I_Collector(2)
2	0.000000e+000	1.836736e-007	0.000000e+000	2.466173e-007
3	-2.000000e-002	-5.159855e-006	-2.000000e-002	-7.082264e-006
4	-4.000000e-002	-1.381116e-005	-4.000000e-002	-1.888655e-005
5	-6.000000e-002	-2.545521e-005	-6.000000e-002	-3.462865e-005
6	-8.000000e-002	-3.779190e-005	-8.000000e-002	-5.113309e-005
7	-1.000000e-001	-4.814865e-005	-1.000000e-001	-6.482606e-005
8	-1.200000e-001	-5.537796e-005	-1.200000e-001	-7.429314e-005
9	-1.400000e-001	-5.985659e-005	-1.400000e-001	-8.010770e-005
10	-1.600000e-001	-6.244343e-005	-1.600000e-001	-8.345355e-005

If the data has not been selected, select a plot and the XY setting dialog displays, as shown in the following figure. Select the X-Y settings.

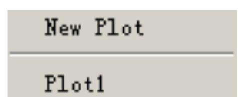
Figure 324: XY setting



Plot according to data

Select **Plot** from the toolbar to display the Plotting menu, as shown in the following figure.

Figure 325: Plotting Menu



After selecting the data to be plotted, choose one of the following commands:

- **Plot1:** Draws a graph on plot1, which already exists. If plot1 is not blank, the original lines remain instead of being deleted.
- **New Plot:** Draws a graph on a new plot separate from the existing one.

NOTE

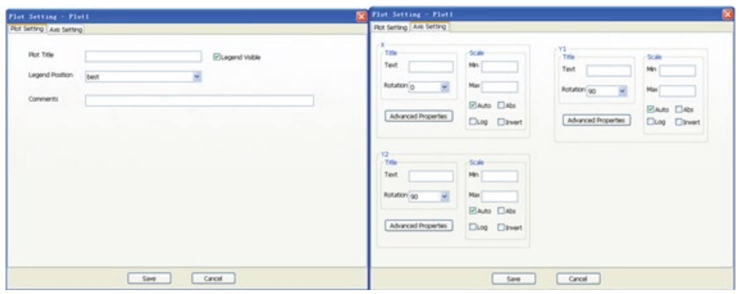
If the test is run on a multigroup, the results of all groups under test are shown on one plot, as shown in the following figure.

Figure 326: Plot shows two ids_vd plots

Define graph properties

Right-click any portion of the graph to be edited, then choose the Plot and Axis Setting command from the Graph Settings menu. The Plotting Setting dialog is displayed. It contains two tabs: On the left is the Plot Setting tab; and on the right is the Axis Setting tab. The following figure illustrates the dialog displayed for each of the tabs.

Figure 327: Plot Setting and Axis Setting



The Plot Setting dialog contains the following settings:

- **Plot Title:** Sets the title of the selected plot.
- **Legend Visible:** If selected, the legend appears on the plot.
- **Legend Position:** Select the position for the legend on the plot. The recommended position choice is Best.
- **Comments:** Used to write comments for the selected plot. In a plot, you can select Ctrl and drag the comment with the mouse to move its position, as shown in the following figure.

After choosing all the settings, select **Save** to apply the settings, or select **Cancel**. The following figure is an example of a plot having a Title on top, Legend at the right side, and Comments at the upper left corner of the plot.

Figure 328: Results after setting the Plot Settings

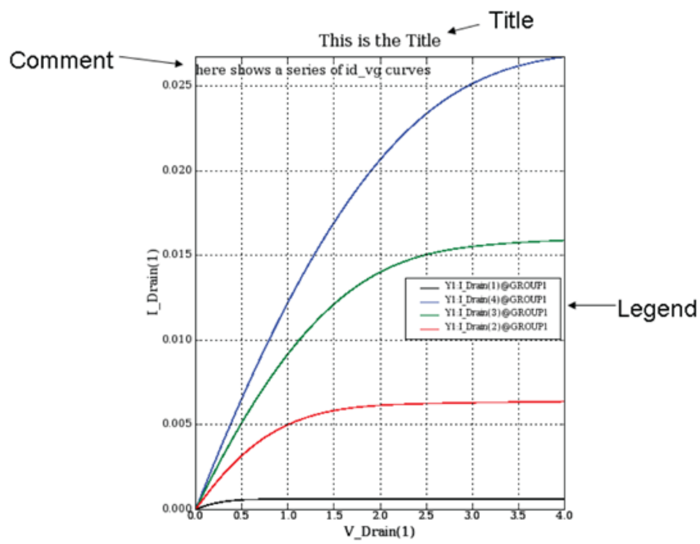
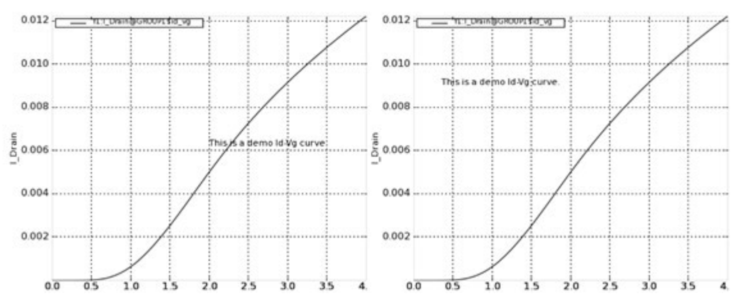
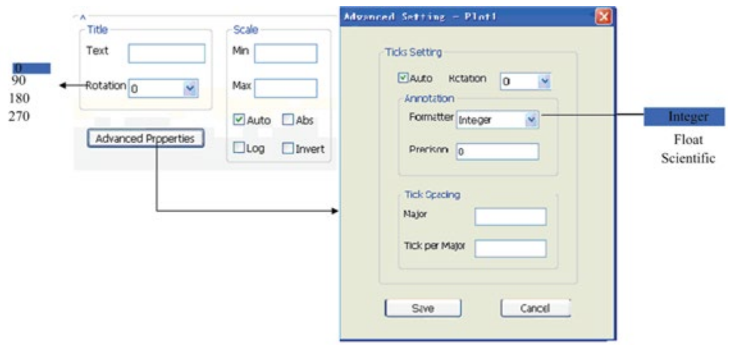


Figure 329: Results of dragging comment



The axis setting tab contains three parts that corresponds to the settings of the X, Y1, and Y2 axis. The items in these three parts are the same for each axis. Only one axis is used in this example. The following figure shows all the items and subitems for one axis on the Axis Setting tab.

Figure 330: Items on the Axis Settings tab



Title section

Sets the properties for the axis title. The options are:

- **Text:** The name of the axis.
- **Rotation:** Rotates the title if necessary. The X-axis default is zero. The Y1 or Y2-axis default is a 90º rotation.
- **Scale section:** Sets the scale of the chosen axis.
- **Min:** Sets the minimum value of the axis. Valid when Auto is cleared.
- **Max:** Sets the maximum value of the axis. Valid when Auto is cleared.
- **Auto:** Autoscales the axis to fit the data. Auto is the default setting.

NOTE

To specify exact numbers for the minimum and maximum values of the scale, clear Auto and enter the values in the Min and Max text boxes.

- **Abs:** Plots the absolute value of the data along the axis.
- **Log:** Arranges the axis in a logarithmic format if selected. If not selected, the axis is arranged in a linear format. The default setting is cleared (linear format).

NOTE

Log format can only be applied to positive data. If there is negative data, select **Abs** to allow Log format.

- **Invert:** Changes the minimum value of the axis to the maximum value and the maximum value to the minimum value.

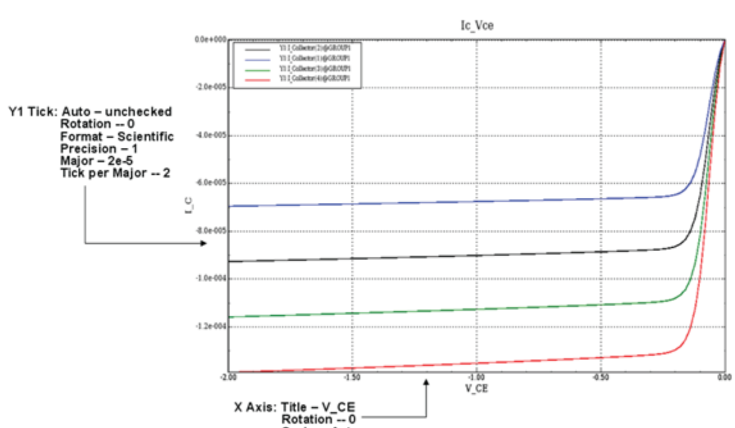
Advanced Properties

Select the **Advanced Properties** button (see next figure) to open the dialog. Items in the Advanced Setting dialog are as follows:

- **Auto:** Sets the ticks (scale) automatically; the other settings are ignored. This is selected by default.
- **Rotation:** Rotates the ticks if necessary. Zero is set by default.
- **Formatter:** Sets the format for the annotation of the ticks. Three formats are provided: Integer, Float, and Scientific.
- **Precision:** Sets the precision of the annotation. Valid when Float or Scientific format is selected.
- **Major:** Sets the value between major ticks along the axis.
- **Tick per Major:** Sets the number of subticks inside a major tick.

After all the properties are set, select **Save** to apply the settings, or **Cancel** to cancel the settings. The following figure shows a series of Ic_Vce curves with marked features.

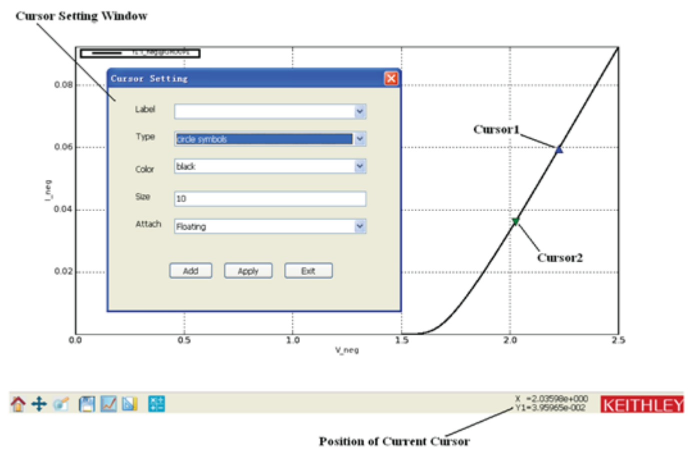
Figure 331: Ic_Vce curves bearing features of Axis and Ticks



Define a cursor

Right-click any portion of the plot and choose Cursor Setting from the Graph Settings menu. The Cursor Setting dialog displays. The following figure shows the Cursor Setting dialog with the cursors on the I_V curve of a diode.

Figure 332: Cursor Setting and the Cursors on Curve



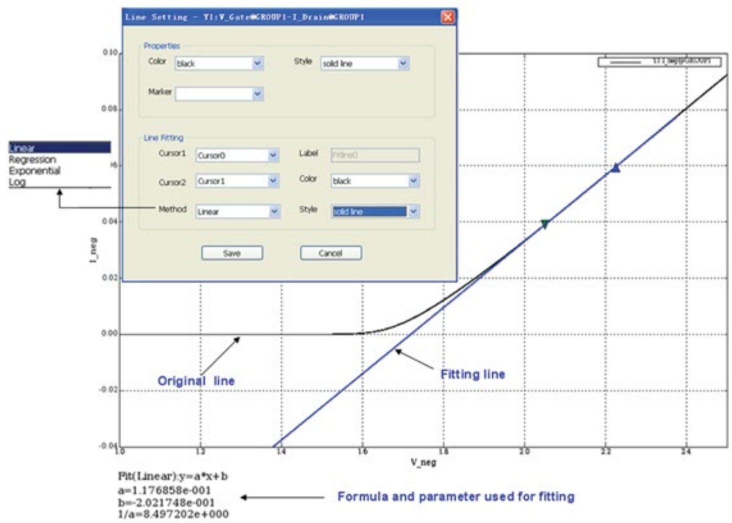
- **Label:** Chooses the cursor to be edited. You can choose from the existing cursors, but cannot change their names.
- **Type:** Chooses the shape of the selected cursor.
- **Color:** Chooses the color of the selected cursor.
- **Size:** Edits the size of the selected cursor.
- **Attach:** Chooses between Floating and certain curves. If Floating is set, the cursor can go anywhere on the plot, otherwise it can only run along the selected curve.
- **Add:** Adds a new cursor. Once Add is selected, a new cursor appears in the Label.
- **Apply:** Applies the settings.
- **Exit:** Exits the active dialog. If Apply is not selected before Exit, the settings are ignored.

To apply the settings for the cursor, select **Apply**. After selecting Apply, the cursor displays on the plot. If you select the cursor, the coordinates (x, y) of the current position display at the bottom of the plot.

Edit line properties

Select the line to be edited, then right-click to choose the Line Properties and Fitting command from the Graph Settings menu. The Line Settings dialog displays, as shown in the following figure.

Figure 333: I-V curve of a diode with linear fitting line and formula used



The previous figure illustrates the original line (black) together with the fitting line (blue) without features. The formula used displays at the bottom left corner of the sheet.

Properties of the line:

- **Color:** Indicates the color of the line selected.
- **Style:** Indicates the style of the line: Solid, Dashed, Dotted, and Dash-Dot.
- **Marker:** Marks the data points. Indicates the type for the marker.

Line Fitting:

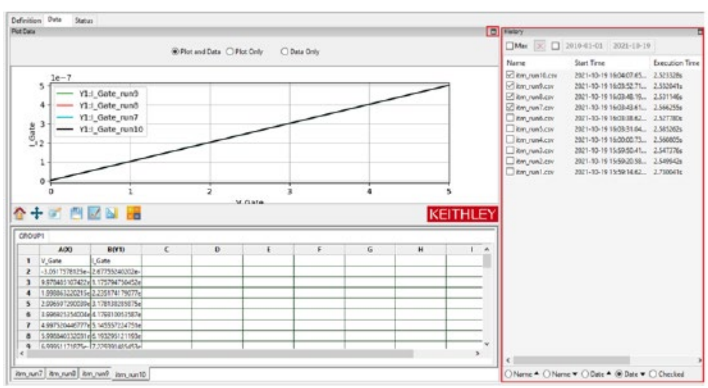
Only one original line can have a fitting line. If the number of plots on the graph exceeds two, the formula will not display. After you complete the settings, select the Save function in the Line Settings dialog to apply the settings, or Cancel.

- **Cursor1:** The first cursor defining the segment of the line to be fitted. Select from the list.
- **Cursor2:** The second cursor defining the segment of the line to be fitted. Select from the list.
- **Method:** Select the function used to fit the segment of the line. Linear, Regression, Exponential, or Log.
- **Label:** Shows the label of the fitting line. You cannot edit this label.
- **Color**: Indicates the color of the fitting line.
- **Style:** Indicates the style of the fitting line: Solid, Dashed, Dotted, or Dash-Dot.

History data

The history data shows data generated from running tests. You can view the data by selecting the enlarge button of the Plot and Data section.

Figure 334: History Data panel

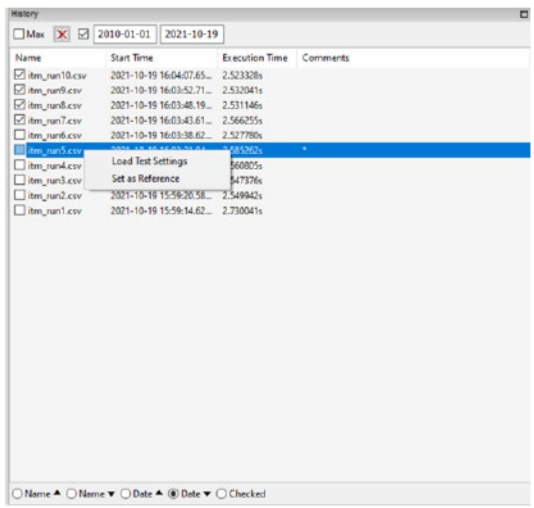


When you select the run button, a new page of test data is appended in the Data panel. The new data is named `testname_runX` (*X* is the count of runs). Each test adds a new plot of test data that is overlayed in the Plot panel. The history data is added in the History section, which is named `testname_runX.csv` (*X* is the count of runs).

The history data item includes the name and information about start time, execution time, and comments. You can edit the comments of each history data item in the interface.

When Max is selected, the first five pages from the run history are displayed. When you select or clear the history data item, the corresponding data and plot can be viewed or hidden in the data and plot area. For example, in the following figure, `itm_run1.csv` is cleared and is not displayed.

Figure 335: History data



There are two parameters that contain test history data. These files are in the directory `C:\ACS\KATS\ACS_settings.ini`.

max_pages:	The number of pages that can be selected and displayed in the plot data panel.
max_histories:	The amount of history data that can be saved.

Figure 336: Parameters settings history data

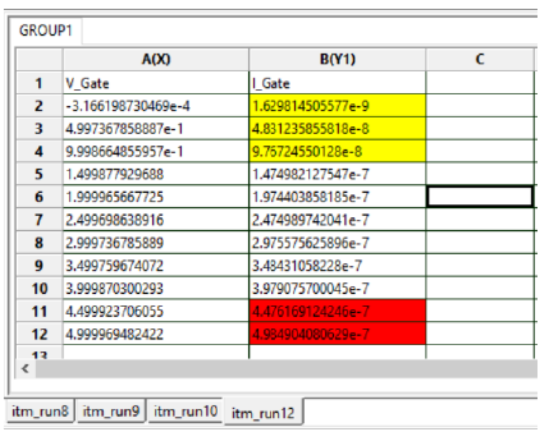
```
[run_history]
max_pages = 5
max_histories=50
```

- **Test settings:** To load test settings from your previous testing right-click a history data item and choose **Load Test Settings**. This allows you to use the test settings on the current test module (refer to the previous figure).
- **Reference:** To create a reference, you can right-click a history data item and choose **Select as Reference**. This reference is used for all the selected history data on the Data and Plot area (refer to the previous figure).
- **Date:** To see the date, select the box next to the red X delete button. You can filter the dates by selecting a range of dates (refer to the previous figure).
- **Sort data:** You can sort your test data by name, date, selected or cleared in ascending or descending order.
- **Delete:** To delete a history data item, select the item you want to delete, then select the red X delete button (refer to the previous figure).

Fail data

If a test result is out of specifications for Spec Low or the Spec High range, as defined in the Pattern-Limits page, the background color of the data cell changes to the Manual Run Fail Color Setting defined in the **Tools > Preference > Advanced** panel. For example, in the following figure, the yellow cells indicate the specifications are lower than expected and the red cells indicate they are higher.

Figure 337: Background color fail data


GROUP1

	A(X)	B(Y1)	C
1	V_Gate	I_Gate	
2	-3.166198730469e-4	1.629814505577e-9	
3	4.997367858887e-1	4.831235855818e-8	
4	9.998664855957e-1	9.75724550128e-8	
5	1.499877929688	1.474982127547e-7	
6	1.999965667725	1.974403858185e-7	
7	2.499698638916	2.474989742041e-7	
8	2.999736785889	2.975575625896e-7	
9	3.499759674072	3.48431058228e-7	
10	3.999870300293	3.979075700045e-7	
11	4.499923706055	4.476169124246e-7	
12	4.999969482422	4.984904080629e-7	
13			

itm_run8 itm_run9 itm_run10 itm_run12

Formulator arguments and constants

A formula created using the Formulator performs calculations on any combination of the following:

- ITM, STM, CTM, or PTM test data.
- Secondary data created by other Formulator formulas.
- Standard constants in the list of constants.

Some of the functions operate on Data tab columns of values (vectors) only. Others operate on single values (scalar) only. Still others operate on both single values and columns of values (vectors).

Likewise, the results of real-time functions, operators, and formulas may be a column of values (vector) in the Data tab or a column containing only a single value.

The Formulator provides a variety of functions and operators. Some of these may be used for real-time, in-test calculations for ITM data. Others may be used only for post-test data computations.

A formula containing exclusively real-time operators and functions is a real-time formula. If a real-time formula is specified as part of an ITM definition, it executes for each data point generated by the ITM, just after it is generated. The results of a real-time formula may be viewed in the Data sheet or plotted during the test in the same way as test data.

NOTE

Real-time calculations do not apply to CTM data. A CTM Data worksheet does not update until the test is finished.

The following operators and functions are real-time operators and functions:

- Operators: +, -, *, /, ^ (exponent).
- Functions: ABS, DELTA, DIFF, EXP, LN, LOG10, SQRT

The formula below is a real-time formula:

- RESULT1 = ABS(DELTA(I_ drain))

Real-time formulas execute as follows:

- If a real-time formula is created before the ITM has been run, the formula executes automatically during each run.
- If a real-time formula is created after an ITM has been run, the formula executes initially upon adding it to the ITM and automatically during each subsequent run.

Post-test-only functions and formulas

A formula containing one or more of the remaining Formulator functions is a post-test-only formula. It executes only at the end of each run of the ITM, STM, or CTM in which the formula is defined.

The results of a post-test-only formula may be viewed in the Data worksheet or plotted only at the end of a test.

The following functions are post-test-only functions:

AT	AVG	EXPFIT	EXPFITA	EXPFITB
LINFIT	LINFITSLP	LINFITXINT	LINFITYINT	MAX
MAXPOS	MIN	MINPOS	REGFIT	REGFITSLP
REGFITXIN	REGFITYINT	TANFIT	TANFITSLP	TANFITXINT
TANFITYINT	VTLINGM	VTSATGM		

The formula below is a post-test-only formula and MAX is a post-test-only function:

```
RESULT2 = MAX(ABS(DELTA(I_ drain)))
```

Post-test-only formulas execute as follows:

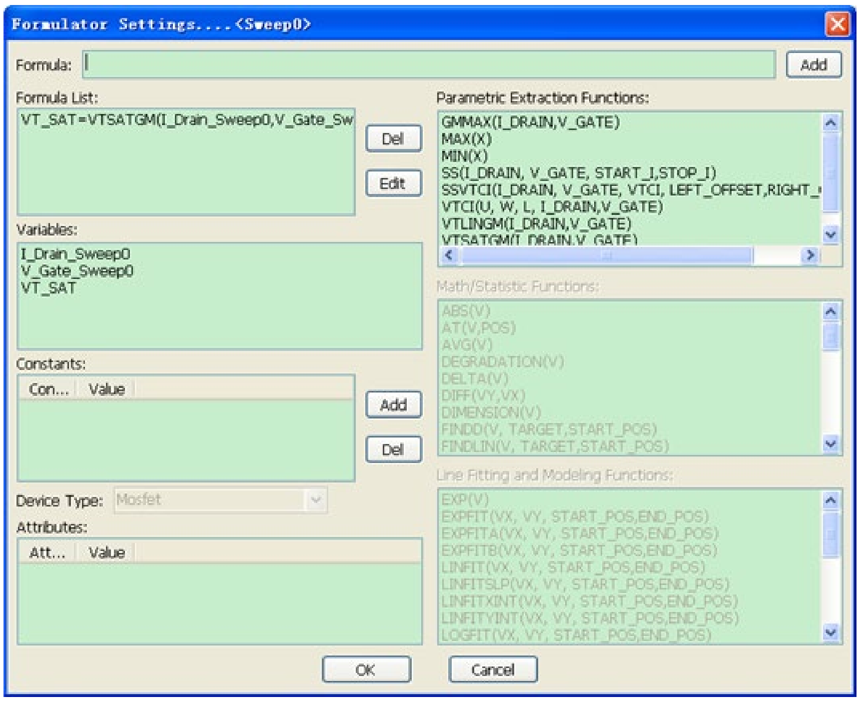
- If a post-test-only formula is created before the ITM, STM or CTM has been run, the formula executes automatically at the conclusion of each run.
- If a post-test-only formula is created after an ITM, STM, or CTM has been run, the formula executes initially upon adding it to the STM, CTM, or ITM and automatically at the conclusion of each subsequent run.

Formulator settings

To start the Formulator:

1. Open the Data tab for the test data that you want to analyze.
2. In the Data tab, select Formulator. The Formulator dialog opens, as shown in the following figure.

Figure 338: Formulator Settings dialog



Formulator dialog

This section summarizes each Formulator dialog feature:

- Formula area
- Formula boxes

The boxes in the Formula area of the dialog used as follows:

- The upper Formula box is for creating new formulas or editing existing formulas.
- The lower Formula box displays the formulas that have been created for the ITM, STM, CTM, or PTM.

Add formulas function

Selecting the Add function produces the following results:

- Starts calculation execution for the formula in the upper Formula box.
- Moves a formula from the upper Formula box into the collection of formulas displayed in the lower Formula box.

Delete formulas function

Selecting the Delete function deletes the formula selected in the lower box.

Copy to All formulas function

Selecting the Copy to All function copies the formula settings of the subdevice to other subdevices.

Functions area

The functions area displays a matrix of functions and a generic F= function that can be used in place of a variable name to complete the equation. When you select a function, the function is added to the equation and displayed in the upper dialog of the Formula box at the cursor position.

Variables area

The Variables area lists the names of all columns in the Data sheet.

NOTE

Columns created by some functions may contain only a single value.

Device Attributes area

The Device Attributes area describes the Attribute, Value, and Unit of the device. The L, T, and W below the Attribute represents Length, Temperature, and Width.

Constants and Values area

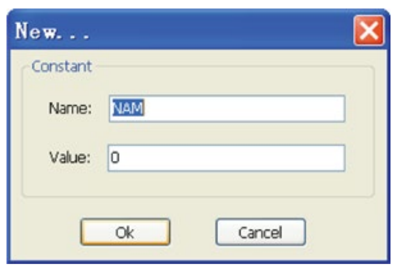
The Constants and Values area provides constants that may be conveniently inserted by name.

When you select the constant in the constants list, the constant is added to the equation in the upper dialog at the cursor position.

Add new constants

To add a new constant, select New. A dialog opens that allows you to input the name and value of your new constant. After you input the name and value, select **OK**. The new constant is added to the constants list, as shown in the following figure.

Figure 339: New constant



Delete constants

To delete a constant, select **Del**.

Copy existing project to a new system

When you copy an existing project to a new system, if the project is in the Formulator, you should also copy the `KATS_const.ini` file. If you do not copy this file, the Formulator functions are not available in the new system.

If ACS is installed on a computer where environment variables contain `KIDAT`, the `KATS_Const.ini` file is in the path specified in the value of `KIDAT`. For example, `KIDAT=C:\S4200\sys\dat` indicates that the variable is defined in the file `C:\S4200\sys\dat\KATS_Const.ini`.

If `KIDAT` is not defined in environment variables, the `KATS_Const.ini` file is in the path where ACS is installed. For example, if ACS is in the path `\\ACS\KATS`, the variable-defined files are in `\\ACS\KATS\KATS_Const.ini`.

Formulator functions

The Formulator functions, operators, and constants can be used individually or in combination to create simple or complex analysis equations.

Multiple functions can be nested. For example, in one equation you can perform the following:

- Calculate a series of moving averages for a column of data (vector) in the data worksheet using the MAVG function.
- Find the maximum value of the MAVG averages using the MAX function.
- Multiply the MAX value by a constant.

The equation below illustrates this use of nested Formulator functions.

```
MAXDIFF = 10*MAX(MAVG(ColumnA))
```

The number of levels of nesting is unlimited. A brief description of each Formulator function is provided in the following table. The purpose, format, and arguments for the Formulator functions and all other functions available in the Formulator are described in the following sections.

General

Function	Brief description
ABS	Calculates the absolute value of each value in the designated column (vector) or the absolute value of any operand.
DELTA	Returns the differences between the adjacent values in a column (vector).
EXP	Returns the exponential, e^{value}, for each value in a column (vector) or for any operand.
LN	Returns the base-e (natural, Napierian) log of each value in a designated column (vector) or the Napierian log of any operand.
LOG	Returns the log value of each value in a designated column (vector) or the log of any operand.
SQRT	Returns the square root of each value in a designated column (vector) or the square root of any operand.

Statistics

Function	Brief description
AVG	Returns the average of all values in the column (vector).
GNMAX	Returns the maximum value from two columns (vectors).
MAVG	Provides moving averages for the vector.
MAX	Searches all values in a column (vector) and returns the maximum value.
MEDIAN	Calculates the average value between the start and stop index.
MIN	Searches all values in a column (vector) and returns the minimum value.

Array

Function	Brief description
AT	Extracts and returns a single value from a column (vector).
DIFF	For all the values in two selected columns (vectors), returns a third column (vector) that contains the difference between the coefficients.
DIMENSION	Returns a subarray that contains the number of occurrences of the last elements of the array.
FINDD	Searches down the column until it finds a value that matches the user-specified value *X*. FINDD searches a column *V*, beginning at *START*. Then it returns the row number of that value.
FINDLIN	Searches down the column until it finds a value that is closest, but does not exceed, the user-specified value *X*.
FINDU	Searches up the column until it finds a value that matches the user-specified value *X*.
FIRSTPOS	Returns the row number of the first value in a column.
JOIN	Combines multiple arrays and returns a new array.
LASTPOS	Returns the row number of the last value in a column.
MAXPOS	Searches all values in a column, finds the maximum value, and returns the row number of the maximum value.
MINPOS	Searches all values in a column, finds the minimum value, and returns the row number of the minimum value.
SUBARRAY1	Extracts a subarray based on a full array.
SUBARRAY2	Extracts a subarray based on a full array.

Line fits

Function	Brief description
EXPFIT	Performs an exponential fit and returns a new column of Y values.
EXPFITA	Returns the value of the constant EXPFITA as part of the formula to perform an exponential fit.
EXPFITB	Returns the value of the constant EXPFITB as part of the formula to perform an exponential fit.
LINFIT	Finds a linear equation and returns a new column.
LINFITSLP	Finds a linear equation and returns the slope.
LINFITXINT	Finds a linear equation and returns the X intercept.
LINFITYINT	Finds a linear equation and returns the Y intercept.
LOGFIT	Returns the results of a log equation.
REGFIT	Performs a linear regression fit.
REGFITSLP	Performs a linear regression fit.
REGFITXINT	Performs a linear regression fit.
REGFITYINT	Performs a linear regression fit.
REGFIT_LGX_LGY	Performs a linear regression operation using the linear equation LGY = a + bLGX.
REGFIT_LGX_Y	Performs a linear regression operation using the linear equation Y = a + bLGX.
REGFIT_X_LGY	Performs a linear regression operation using the linear equation LGY = a + bX.
TANFIT	Finds a linear equation of the form `Y = a + bX` from two columns, *VX* and *VY*.
TANFITSLP	Finds a linear equation of the form `Y = a + bX` from two columns, *VX* and *VY*.
TANFITXINT	Finds a linear equation of the form `Y = a + bX` from two columns, *VX* and *VY*.
TANFITYINT	Finds a linear equation of the form `Y = a + bX` from two columns, *VX* and *VY*.

FFT

Function	Brief description
GET_FREQ	Gets the fast Fourier transform (FFT) frequency for an input time list.
POWERFFT	Performs the forward fast Fourier transform (FFT) for the data source input and return the power spectrum.
SMOOTH	Smooth the vector using fast Fourier transform (FFT).

Misc

Function	Brief description
CURRENT_NOISE	Calculates noise current from Ig.
DEGRADATION	Calculates the degradation based on the first position.
GET_TBD	Gets breakdown (BD) time of the time-dependent dielectric breakdown (TDDB) test.
GET_TBD2	Gets breakdown (BD) time of the time-dependent dielectric breakdown (TDDB) test.
GET_VBD	Gets breakdown (BD) voltage for the VRamp test.
LOG10	Returns the base-10 log of each value in a designated column (vector) or the base-10 log of any operand.
MOBILITY	Mobility at Vg_tar by Id-Vg curve.
NOISE_UTIM	Generates correct time array for noise current.
POW	Returns the power value of each value in a designated column (vector) or any operand.
RES	Calculates the resistance vector value by giving V and I.
RES_4WIRE	Calculate 4-wire resistance vector value by giving V and I.
RES_4WIRE_AVG	Calculates 4-wire resistance average value using V and I.
RES_AVG	Calculates resistance average value using V and I.
RSQUAR	Calculates correlation coefficients from an input matrix *VX* and *VY*.
SS	Extracts the subthreshold swing between a certain duration of current.
SSVTCI	Extracts the sub-threshold swing for a specific duration between left offset and right offset of the constant current threshold voltage.
TAR_YatX	Finds y_tar in y corresponding to x_tar in the x index.
TTF_DID_LGT	Finds the time to failure (TTF) by analyzing two arrays in percentage format (time array and degradation).
TTF_DID_T	Finds the time to failure (TTF) by analyzing two arrays in percentage format (time array and degradation).
TTF_LGDID_LGT	Finds the time to failure (TTF) by analyzing two arrays in percentage format (time array and degradation).
TTF_LGDID_T	Finds the time to failure (TTF) by analyzing two arrays in percentage format (time array and degradation).
VTCI	Finds the threshold voltage at target drain current using current density, device width, and device length.
VTEXLIN	Extrapolates the threshold voltage from the measurement of the maximum slope for the drain current and gate voltage curve using 0.5 times bias drain voltage.
VTLINGM	Extrapolates threshold voltage from the measurement of the maximum slope for the drain current and gate voltage curve in the linear region of a MOSFET.
VTSATGM	Extrapolates threshold voltage from the measurement of the drain current and gate voltage curve in the saturation region of a MOSFET.
VT_SAT_TRI	Extrapolates threshold voltage by triple dividing the measured drain current root curve.

Formulator function reference

Each of the Formulator functions is described in the following.

ABS Formulator function

Calculates the absolute value of each value in the designated column (vector) or the absolute value of any operand.

Usage

`ABS(Value)`

Value	The name of any column (vector) in the Data Series list or any operand

Details

You can use this function to do calculations in real time (while a test is executing).

Example

`F2 = ABS(GateI)`	Returns the absolute value of the gate current.

Also see

None

DELTA Formulator function

This command returns the differences between the adjacent values in a column (vector). That is, for column V, `DELTA` returns (V2 − V1), (V3 − V2), and so on.

Usage

`DELTA(Value)`

Value	The name of any column (vector) in the Data Series list

Details

You can use this function to do calculations in real time (while a test is executing).

Example

`GM = DELTA(DRAINI)/DELTA(GATEV)`	

Also see

None

EXP Formulator function

Returns the exponential, e^{value}, for each value in a column (vector) or for any operand.

Usage

`EXP(Value)`

Value	The name of any column (vector) in the Data Series list or any operand

Details

You can use this function to do calculations in real time (while a test is executing).

Example

```
NEWCURRENT = CURRENT*EXP(ANODEV)
```

Also see

LN Formulator function (on page 5-260)

LN Formulator function

This command returns the base-e (natural, Napierian) log of each value in a designated column (vector) or the Napierian log of any operand.

Usage

`LN(Value)`

Value	The name of any column (vector) in the Data Series list or any operand

Details

You can use this function to do calculations in real time (while a test is executing).

Example

```
DIODEV = LN(ANODEI)*0.026
```

Also see

EXP Formulator function (on page 5-260)

LOG Formulator function

Returns the log value of each value in a designated column (vector) or the log of any operand.

Usage

`LOG(Value, VBASE)`

Value	The name of any column (vector) listed under Columns or a number
VBASE	The base value of logarithms

Details

This function cannot accept two arrays in its parameters.

You can use this function to do calculations in real time (while a test is executing).

Example

```
F1 = LOG(I_DRAINI, 2)
```

Also see

None

SQRT Formulator function

Returns the square root of each value in a designated column (vector) or the square root of any operand.

Usage

`SQRT(Value)`

Value	The name of any column (vector) in the Data Series list or any operand

Details

A negative value of `X` returns `#REF` in the worksheet.

You can use this function to do calculations in real time (while a test is executing).

Example

```
TWO = SQRT(4)
```

Also see

None

Array

The following Formulator functions work with arrays.

AT Formulator function

Extracts and returns a single value from a column (vector).

Usage

`AT(Value, POS)`

Value	The name of any column (vector) in the Data Series list or any operand
POS	The row number of column `Value` where the single value is located

Example

```
IDSAT = AT(DRAINI, 36)
```

Also see

None

DIFF Formulator function

For all the values in two selected columns (vectors), returns a third column (vector) that contains the difference between the coefficients.

Usage

`DIFF(V1, V2)`

V1	The name of any column (vector) listed under **Columns**
V2	The name of any column (vector) listed under **Columns**

Details

Each coefficient is calculated as follows:

V1/*V2*

Where:

- *V1* = The difference between a pair of adjacent values in the first column.
- *V2* = The difference between the corresponding values in the second column.

That is, for columns *V1* and *V2*, `DIFF` returns the following:

$(V1_2 - V1_1)/(V2_2 - V2_1)$, $(V1_3 - V1_2)/(V2_3 - V2_2)$, and so on.

You can use this function to do calculations in real time (while a test is executing).

Example

```
GM = DIFF(DRAINI, GATEV)
```

Also see

None

DIMENSION Formulator function

Returns a subarray that contains the number of occurrences of the last elements of the array.

Usage

`DIMENSION(V)`

V	The name of any column (vector) listed under **Columns**

Example

```
V2 = DIMENSION(V1)
```

Also see

None

FINDD Formulator function

The find down function searches down the column until it finds a value that matches the user-specified value *X*. `FINDD` searches a column *V*, beginning at *START*. Then it returns the row number of that value.

Usage

`FINDD(V, X, START)`

V	The name of any column (vector) listed under Columns
X	Any value, which may be the result of another calculations
START	The row number of the starting value for the search

Details

If `FINDD` does not find an exact match for *X*, it returns the row number of the *V* value that is closest to *X*.

Example

```
IF = AT(ANODEI, FINDD(ANODEV, 0.7, FIRSTPOS(ANODEV)))
```

Also see

FINDLIN (on page 5-264)
FINDU (on page 5-265)

FINDLIN Formulator function

Find using linear interpolation searches down the column until it finds a value that is closest, but does not exceed, the user-specified value *X*.

Usage

`FINDLIN(V, X, START)`

V	The name of any column (vector) listed under Columns
X	Any value, which may be the result of another calculation
START	The row number of the starting value for the search

Details

`FINDLIN` searches a column *V*, beginning at *START*. Linear interpolation is then used to determine its decimal location between the found value and the next value in the column. The returned index number (in decimal format) indicates the position of the specified value.

Assume you want to use `FINDLIN` to locate value 6 in the following array:

```
(Index 1)0
(Index 2)1
(Index 3)4
(Index 4)8
```

The search finds the index marker that is closest to (but does not exceed) 6. In this case, Index 3 is the closest. Linear interpolation is then used to determine the decimal position of the specified value (6) that is between `Index 3` (value 4) and `Index 4` (value 8). Value 6 is halfway between Index 3 and Index 4. Therefore, `FINDLIN` returns Index 3.5.

Example

```
IF = AT(ANODEI, FINDLIN(ANODEV, 0.7, FIRSTPOS(ANODEV)))
```

Also see

FINDD (on page 5-263)
FINDU (on page 5-265)

FINDU Formulator function

Find up searches up the column until it finds a value that matches the user-specified value *X*. It then returns the row number of that value. `FINDU` searches a column *V*, beginning at *START*. Then it returns the row number of that value.

Usage

`FINDU(V, X, STARTPOS)`

V	The name of any column (vector) listed under Columns
X	Any value, which may be the result of another calculation
STARTPOS	The row number of the starting value for the search

Details

If `FINDU` does not find an exact match for *X*, it returns the row number of the *V* value that is closest to *X*.

Example

```
IF = AT(ANODEI, FINDU(ANODEV, 0.7, LASTPOS(ANODEV)))
```

Also see

FINDD (on page 5-263)
FINDLIN (on page 5-264)

FIRSTPOS Formulator function

Returns the row number of the first value in a column.

Usage

`FIRSTPOS(V)`

V	The name of any column (vector) listed under **Columns**

Example

```
STARTOFARRAY = FIRSTPOS(DRAINI)
```

Also see

LASTPOS (on page 5-266)

JOIN Formulator function

Combines multiple arrays and returns a new array.

Usage

`JOIN(array1, array2))`

array1	First array
array2	Second array

Example

```
newarray = JOIN(array1, array2)
```

Also see

None

LASTPOS Formulator function

Returns the row number of the last value in a column.

Usage

`LASTPOS(Value)`

Value	The name of any column (vector) listed under **Columns**

Example

```
NUMSWEEPPTS = LASTPOS(COLLECTORI)
```

Also see

FIRSTPOS (on page 5-265)

MAXPOS Formulator function

Searches all values in a column, finds the maximum value, and returns the row number of the maximum value.

Usage

`MAXPOS(V)`

V	The name of any column (vector) listed under Columns

Example

```
PEAKSTRESS = AT(GATEV, MAXPOS(SUBSTRATEI))
```

Also see

None

MINPOS Formulator function

Searches all values in a column, finds the minimum value, and returns the row number of the minimum value.

Usage

`MINPOS(V)`

V	The name of any column (vector) listed under Columns

Example

```
LOCATION = MINPOS(DRAINI)
```

Also see

None

SUBARRAY1 Formulator function

Extracts a subarray based on a full array.

Usage

`SUBARRAY1(V, START, PERIOD)`

V	Vector array
START	Start index
PERIOD	Extraction time

Example 1

```
VGate2 = SUBARRAY1(VGate, 1, 5)
```

Also see

None

SUBARRAY2 Formulator function

Extracts a subarray based on a full array.

Usage

`SUBARRAY2(V, START, STOP)`

V	Vector array
START	Start index
STOP	Stop index

Example 1

```
VGate2 = SUBARRAY2(VGate, 1, 50)
```

Also see

None

Line fits

The Line Fit Formulator functions allow you to set up distinct types of line fits.

EXPFIT Formulator function

Performs an exponential fit and returns a new column of Y values.

Usage

`EXPFIT(VX, VY, STARTPOS, ENDPOS)`

VX	The name of any column (vector) listed under Columns
VY	The name of any column (vector) listed under Columns
STARTPOS	For the range of X and Y values to be exponentially fitted, the row number of the starting values
ENDPOS	For the range of X and Y values to be exponentially fitted, the row number of the ending values

Details

Performs an exponential fit. Fits the following exponential relationship to a specified range of values in two columns (vectors): one column, VX, containing X values and the other column, VY, containing Y values:

$$Y = EXPFITA * e^{(EXPFITB * X)}$$

Where `EXPFITA` and `EXPFITB` are fit constants. Using this exponential relationship, returns a new column (vector) containing Y values calculated from all X values in column VX.

If a *VX* or *VY* value at either *STARTPOS* or *ENDPOS* is an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
DIODEI = EXPFIT(ANODEV, ANODEI, 2, LASTPOS(ANODEV))
```

Also see

EXPFITA (on page 5-270)
EXPFITB (on page 5-271)

EXPFITA Formulator function

Returns the value of the constant `EXPFITA` as part of the formula to perform an exponential fit.

Usage

`EXPFITA(VX, VY, STARTPOS, ENDPOS)`

VX	The name of any column (vector) listed under Columns
VY	The name of any column (vector) listed under Columns
STARTPOS	For the range of X and Y values to be exponentially fitted, the row number of the starting values
ENDPOS	For the range of X and Y values to be exponentially fitted, the row number of the ending values

Details

Performs an exponential fit. Fits the following exponential relationship to a specified range of values in two columns (vectors): one column, VX, containing X values and the other column, VY, containing Y values:

$$Y = EXPFITA * e^{(EXPFITB * X)}$$

Where `EXPFITA` and `EXPFITB` are fit constants.

If a *VX* or *VY* value at either *STARTPOS* or *ENDPOS* is an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
OFFSETI = EXPFITA(ANODEV, ANODEI, 2, LASTPOS(ANODEV))
```

Also see

EXPFIT (on page 5-269)
EXPFITB (on page 5-271)

EXPFITB Formulator function

Returns the value of the constant `EXPFITB` as part of the formula to perform an exponential fit.

Usage

`EXPFIT(VX, VY, STARTPOS, ENDPOS)`

VX	The name of any column (vector) listed under Columns
VY	The name of any column (vector) listed under Columns
STARTPOS	For the range of X and Y values to be exponentially fitted, the row number of the starting values
ENDPOS	For the range of X and Y values to be exponentially fitted, the row number of the ending values

Details

Performs an exponential fit. Fits the following exponential relationship to a specified range of values in two columns (vectors): one column, VX, containing X values and the other column, VY, containing Y values:

$$Y = EXPFITA * e^{(EXPFITB * X)}$$

Where `EXPFITA` and `EXPFITB` are fit constants.

If a *VX* or *VY* value at either *STARTPOS* or *ENDPOS* is an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
DIODEIDEALITY = 1/(EXPFITB(ANODEV, ANODEI, 2, LASTPOS(ANODEV))*0.0257)
```

Also see

EXPFITA (on page 5-270)
EXPFITB (on page 5-271)

LINFIT Formulator function

Finds a linear equation and returns a new column.

Usage

`LINFIT(VX, VY, STARTPOS, ENDPOS)`

VX	The name of any column (vector) listed under Columns
VY	The name of any column (vector) listed under Columns
STARTPOS	The row number of the first set of `X` and `Y` values
ENDPOS	The row number of the second set of `X` and `Y` values

Details

Finds a linear equation of the form `Y = a + bX` from two sets of `X` and `Y` values selected from two columns (vectors), *VX* and *VY*. This equation corresponds to a line drawn through two points on a curve that is created by plotting the values in *VY* against the values in *VX*. The two points are specified by the arguments *STARTPOS* and *ENDPOS*.

Using the linear equation, returns a new column (vector) containing `Y` values calculated from all `X` values in column `VX`.

If a *VX* or *VY* value at either *STARTPOS* or *ENDPOS* is an invalid number (that is, the value is `#REF`), the function will not return a valid result.

To return a linear regression fit for two columns (vectors), use the `REGFIT` function.

Example

```
RESISTORFIT = LINFIT (RESV, RESI, FIRSTPOS(RESV), LASTPOS(RESV))
```

Also see

REGFIT (on page 5-277)

LINFITSLP Formulator function

Finds a linear equation and returns the slope.

Usage

`LINFITSLP(VX, VY, STARTPOS, ENDPOS)`

VX	The name of any column (vector) listed under Columns
VY	The name of any column (vector) listed under Columns
STARTPOS	The row number of the first set of `X` and `Y` values
ENDPOS	The row number of the second set of `X` and `Y` values

Details

Finds a linear equation and returns the slope as follows:

- Finds a linear equation of the form `Y = a + bX` from two sets of `X` and `Y` values selected from two columns, *VX* and *VY*. This equation corresponds to a line drawn through two points on a curve that is created by plotting the values in *VY* against the values in *VX*. The two points are specified by the arguments *STARTPOS* and *ENDPOS*.
- Returns the slope of the linear equation (value of `b` in `Y = a + bX`).

If a *VX* or *VY* value at either *STARTPOS* or *ENDPOS* is an invalid number (that is, the value is `#REF`), the function will not return a valid result.

To return the slope of a linear regression fit for two columns (vectors), use the `REGFITSLP` function.

Example

```
RESISTANCE = 1/LINFITSLP(RESV, RESI, FIRSTPOS(RESV), LASTPOS(RESV))
```

Also see

REGFITSLP (on page 5-278)

LINFITXINT Formulator function

Finds a linear equation and returns the X intercept.

Usage

`LINFITXINT(VX, VY, STARTPOS, ENDPOS)`

VX	The name of any column (vector) listed under Columns
VY	The name of any column (vector) listed under Columns
STARTPOS	The row number of the first set of X and Y values
ENDPOS	The row number of the second set of X and Y values

Details

Finds a linear equation and returns the X intercept as follows:

- Finds a linear equation of the form `Y = a + bX` from two sets of `X` and `Y` values selected from two columns (vectors), *VX* and *VY*. This equation corresponds to a line drawn through two points on a curve that is created by plotting the values in *VY* against the values in *VX*. The two points are specified by the arguments *STARTPOS* and *ENDPOS*.
- Returns the X intercept of the linear equation (value of `-a/b` in `Y = a + bX`).

If a *VX* or *VY* value at either *STARTPOS* or *ENDPOS* is an invalid number (that is, the value is `#REF`), the function will not return a valid result.

To return the `X` intercept of a linear regression fit for two columns (vectors), use the `REGFITXINT` function.

Example

```
EARLYV = LINFITXINT(COLLECTORV, COLLECTORI, 56, 75)
```

Also see

REGFITXINT (on page 5-279)

LINFITYINT Formulator function

Finds a linear equation and returns the Y intercept.

Usage

`LINFITYINT(VX, VY, STARTPOS, ENDPOS)`

VX	The name of any column (vector) listed under Columns
VY	The name of any column (vector) listed under Columns
STARTPOS	The row number of the first set of X and Y values
ENDPOS	The row number of the second set of X and Y values

Details

Finds a linear equation and returns the Y intercept as follows:

- Finds a linear equation of the form `Y = a + bX` from two sets of `X` and `Y` values selected from two columns, *VX* and *VY*. This equation corresponds to a line drawn through two points on a curve that is created by plotting the values in *VY* against the values in *VX*. The two points are specified by the arguments *STARTPOS* and *ENDPOS*.
- Returns the Y intercept of the linear equation (value of `a` in `Y = a + bX`).

If a *VX* or *VY* value at either *STARTPOS* or *ENDPOS* is an invalid number (that is, the value is `#REF`), the function will not return a valid result.

To return the `Y` intercept of a linear regression fit for two columns, use the `REGFITYINT` function.

Example

```
OFFSET = LINFITYINT(GATEV, GATEI, FIRSTPOS(GATEV), LASTPOS(GATEV))
```

Also see

REGFITYINT (on page 5-280)

LOGFIT Formulator function

Returns the results of a log equation.

Usage

`LOGFIT(VX, VY, STARTPOS, ENDPOS, POINTS = 0)`

VX	The name of any column (vector) listed under Columns
VY	The name of any column (vector) listed under Columns
STARTPOS	The row number (index) of the first set of X and Y values
ENDPOS	The row number (index) of the second set of X and Y values

Details

Performs a log fit as follows:

- Uses one column, *VX*, containing X values and the other column, *VY*, containing Y values to find a log equation of the form:
  `Y = a + b * log(X)`
- Using the above logarithmic relationship, returns a new column containing Y values calculated from all X values in column *VX*.

If a *VX* or *VY* value at either *STARTPOS* or *ENDPOS* is an invalid number (that is, the value is #REF), the function will not return a valid result.

Example

```
F3 = LOGFIT(I_FM_pre1, I_Neg_pre1, 23, 56)
```

Also see

None

REGFIT Formulator function

Performs a linear regression fit.

Usage

`REGFIT(VX, VY, STARTPOS, ENDPOS)`

VX	The name of any column (vector) listed under Columns
VY	The name of any column (vector) listed under Columns
STARTPOS	For the range of X and Y values to be fitted, the row number of the starting values
ENDPOS	For the range of X and Y values to be fitted, the row number of the ending values

Details

Performs a linear regression fit as follows:

- Fits the following relationship, of the form `Y = a + bX`, to a specified range of values in two columns (column *VX* containing X values and column *VY* containing Y values):
  `Y = REGFITYINT + REGFITSLP * X`

  where `REGFITSLP` and `REGFITYINT` are slope and Y-intercept constants.
- Using the above linear relationship, returns a new column that contains Y values calculated from all X values in column *VX*.

If a *VX* or *VY* value at either *STARTPOS* or *ENDPOS* is an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
COLLECTORFIT = REGFIT(COLLECTORV, COLLECTORI, 25, LASTPOS(COLLECTORV))
```

Also see

REGFITSLP (on page 5-278)
REGFITYINT (on page 5-280)

REGFITSLP Formulator function

Performs a linear regression fit.

Usage

`REGFITSLP(VX, VY, STARTPOS, ENDPOS)`

VX	The name of any column (vector) listed under Columns
VY	The name of any column (vector) listed under Columns
STARTPOS	For the range of `X` and `Y` values to be fitted, the row number of the starting values
ENDPOS	For the range of `X` and `Y` values to be fitted, the row number of the ending values

Details

Performs a linear regression fit as follows:

- Fits the following relationship, of the form `Y = a + bX`, to a specified range of values in two columns (column *VX* containing `X` values and column *VY* containing `Y` values):
  `Y = REGFITYINT + REGFITSLP * X`

  where `REGFITSLP` and `REGFITYINT` are slope and Y-intercept constants.
- Returns the value of the slope constant `REGFITSLP` in the relationship above.

If a *VX* or *VY* value at either *STARTPOS* or *ENDPOS* is an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
COLLECTORRES = 1/REGFITSLP(COLLECTORV, COLLECTORI, 25, LASTPOS(COLLECTORV))
```

Also see

REGFIT (on page 5-277)
REGFITYINT (on page 5-280)

REGFITXINT Formulator function

Performs a linear regression fit.

Usage

`REGFITXINT(VX, VY, STARTPOS, ENDPOS)`

VX	The name of any column (vector) listed under Columns
VY	The name of any column (vector) listed under Columns
STARTPOS	For the range of `X` and `Y` values to be fitted, the row number of the starting values
ENDPOS	For the range of `X` and `Y` values to be fitted, the row number of the ending values

Details

Performs a linear regression fit as follows:

- Fits the following relationship, of the form `Y = a + bX`, to a specified range of values in two columns (column *VX* containing `X` values and column *VY* containing `Y` values):
  `Y = REGFITYINT + REGFITSLP * X`

  where `REGFITSLP` and `REGFITYINT` are slope and Y-intercept constants.
- Returns the value of the `X` intercept for relationship above. (`–REGFITYINT/REGFITSLP`).

If a *VX* or *VY* value at either *STARTPOS* or *ENDPOS* is an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
EARLYV = REGFITXINT(COLLECTORV, COLLECTORI, 25, LASTPOS(COLLECTORV))
```

Also see

REGFIT (on page 5-277)
REGFITYINT (on page 5-280)

REGFITYINT Formulator function

Performs a linear regression fit.

Usage

`REGFITYINT(`*`VX`*`, `*`VY`*`, `*`STARTPOS`*`, `*`ENDPOS`*`)`

VX	The name of any column (vector) listed under Columns
VY	The name of any column (vector) listed under Columns
STARTPOS	For the range of `X` and `Y` values to be fitted, the row number of the starting values
ENDPOS	For the range of `X` and `Y` values to be fitted, the row number of the ending values

Details

Performs a linear regression fit as follows:

- Fits the following relationship, of the form `Y = a + bX`, to a specified range of values in two columns (column *`VX`* containing `X` values and column *`VY`* containing `Y` values):
  `Y = REGFITYINT + REGFITSLP * X`

  where `REGFITSLP` and `REGFITYINT` are slope and Y-intercept constants.
- Returns the value of the `Y` intercept for relationship above (`REGFITYINT`).

If a *`VX`* or *`VY`* value at either *`STARTPOS`* or *`ENDPOS`* is an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
OFFSET = REGFITYINT(COLLECTORV, COLLECTORI, 25, LASTPOS(COLLECTORV))
```

Also see

REGFIT (on page 5-277)
REGFITXINT (on page 5-279)

REGFIT_LGX_LGY Formulator function

Performs a linear regression operation using the linear equation LGY = a + bLGX.

Usage

`REGFIT_LGX_LGY(`*`VX`*`, `*`VY`*`, `*`STARTPOS`*`, `*`ENDPOS`*`)`

VX	The name of any column (vector) listed under Columns
VY	The name of any column (vector) listed under Columns
STARTPOS	For the range of `X` and `Y` values to be fitted, the row number of the starting values
ENDPOS	For the range of `X` and `Y` values to be fitted, the row number of the ending values

Details

Performs a linear regression fit as follows:

- Fits the following relationship to a specified range of values in two columns (column *VX* containing `LGX` values and column *VY* containing `LGY` values):

  `LGY = REGFITYINT + REGFITSLP * LGX`

  where `REGFITSLP` and `REGFITYINT` are slope and Y-intercept constants.
- Returns a new column (vector) containing `LGY` values calculated from all `LGX` values. For example, input value *VX*, *VY* is normal, however, the operation is built on `LGY` and `LGX`.

If a *VX* or *VY* value at either *STARTPOS* or *ENDPOS* is an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
COLLECTORFIT1 = REGFIT_LGX_LGY(COLLECTORV, COLLECTORI, 25, LASTPOS(COLLECTORV))
```

Also see

REGFIT (on page 5-277)
REGFITXINT (on page 5-279)

REGFIT_LGX_Y Formulator function

Performs a linear regression operation using the linear equation Y = a + bLGX.

Usage

`REGFIT_LGX_Y(VX, VY, STARTPOS, ENDPOS)`

VX	The name of any column (vector) listed under Columns
VY	The name of any column (vector) listed under Columns
STARTPOS	For the range of `X` and `Y` values to be fitted, the row number of the starting values
ENDPOS	For the range of `X` and `Y` values to be fitted, the row number of the ending values

Details

Performs a linear regression fit as follows:

- Fits the following relationship to a specified range of values in two columns (column `VX` containing `LGX` values and column *`VY`* containing `Y` values):
  `Y = REGFITYINT + REGFITSLP * LGX`

  where `REGFITSLP` and `REGFITYINT` are slope and Y-intercept constants.
- Returns a new column (vector) containing `Y` values calculated from all `LGX` values in column `VX`. For example, input value *`VX`*, *`VY`* is normal, however, the operation is built on `Y` and `LGX`.

If a *`VX`* or *`VY`* value at either *`STARTPOS`* or *`ENDPOS`* is an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
COLLECTORFIT2 = REGFIT_LGX_Y(COLLECTORV, COLLECTORI, 25, LASTPOS(COLLECTORV))
```

Also see

REGFIT (on page 5-277)
REGFITXINT (on page 5-279)

REGFIT_X_LGY Formulator function

Performs a linear regression operation using the linear equation LGY = a + bX.

Usage

`REGFIT_X_LGY(VX, VY, STARTPOS, ENDPOS)`

VX	The name of any column (vector) listed under Columns
VY	The name of any column (vector) listed under Columns
STARTPOS	For the range of `X` and `Y` values to be fitted, the row number of the starting values
ENDPOS	For the range of `X` and `Y` values to be fitted, the row number of the ending values

Details

Performs a linear regression fit as follows:

- Fits the following relationship to a specified range of values in two columns (column `VX` containing `X` values and column *VY* containing `LGY` values):
  `LGY = REGFITYINT + REGFITSLP * X`

  where `REGFITSLP` and `REGFITYINT` are slope and Y-intercept constants.
- Returns a new column (vector) containing `LGY` values calculated from all `X` values in column `VX`. For example, input value *VX*, *VY* is normal, however, the operation is built on `LGY` and `X`.

If a *VX* or *VY* value at either *STARTPOS* or *ENDPOS* is an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
COLLECTORFIT3 = REGFIT_X_LGY(COLLECTORV, COLLECTORI, 25, LASTPOS(COLLECTORV))
```

Also see

REGFIT (on page 5-277)
REGFITXINT (on page 5-279)

TANFIT Formulator function

Finds a linear equation of the form `Y = a + bX` from two columns, *VX* and *VY*.

Usage

`TANFIT(VX, VY, POS)`

VX	The name of any column (vector) listed under Columns
VY	The name of any column (vector) listed under Columns
POS	The row number where the tangent is to be found

Details

Finds a linear equation of the form `Y = a + bX` from two columns, *VX* and *VY*. This equation corresponds to a tangent of the curve that is created by plotting the values in *VY* against the values in *VX*. The value at which the tangent is found is specified by the argument *POS*.

Using the linear equation, returns a new column containing `Y` values calculated from all `X` values in column *VX*.

If a *VX* or *VY* value at *POS* is an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
VTFIT = TANFIT(GATEV, DRAINI, MAXPOS(GM))
```

Also see

TANFITSLP (on page 5-285)
TANFITXINT (on page 5-286)
TANFITYINT (on page 5-287)

TANFITSLP Formulator function

Finds a linear equation of the form `Y = a + bX` from two columns, *VX* and *VY*.

Usage

`TANFITSLP(VX, VY, POS)`

VX	The name of any column (vector) listed under Columns
VY	The name of any column (vector) listed under Columns
POS	The row number where the tangent is to be found

Details

Finds a linear equation of the form `Y = a + bX` from two columns, *VX* and *VY*. This equation corresponds to a tangent of the curve that is created by plotting the values in *VY* against the values in *VX*. The value at which the tangent is found is specified by the argument *POS*.

Returns the slope of the linear equation (value of `b` in `Y = a + bX`).

If a *VX* or *VY* value at *POS* is an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
VTSLOPE = TANFITSLP(GATEV, DRAINI, MAXPOS(GM))
```

Also see

TANFIT (on page 5-284)
TANFITXINT (on page 5-286)
TANFITYINT (on page 5-287)

TANFITXINT Formulator function

Finds a linear equation of the form `Y = a + bX` from two columns, *`VX`* and *`VY`*.

Usage

`TANFITXINT(VX, VY, POS)`

VX	The name of any column (vector) listed under Columns
VY	The name of any column (vector) listed under Columns
POS	The row number where the tangent is to be found

Details

Finds a linear equation of the form `Y = a + bX` from two columns, *`VX`* and *`VY`*. This equation corresponds to a tangent of the curve that is created by plotting the values in *`VY`* against the values in *`VX`*. The value at which the tangent is found is specified by the argument *`POS`*.

Returns the `X` intercept of the linear equation (value of `-a/b` in `Y = a + bX`).

If a *`VX`* or *`VY`* value at *`POS`* is an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
VT = TANFITXINT(GATEV, DRAINI, MAXPOS(GM))
```

Also see

TANFIT (on page 5-284)
TANFITSLP (on page 5-285)
TANFITYINT (on page 5-287)

TANFITYINT Formulator function

Finds a linear equation of the form `Y = a + bX` from two columns, *VX* and *VY*.

Usage

`TANFITYINT(VX, VY, POS)`

VX	The name of any column (vector) listed under Columns
VY	The name of any column (vector) listed under Columns
POS	The row number where the tangent is to be found

Details

Finds a linear equation of the form `Y = a + bX` from two columns, *VX* and *VY*. This equation corresponds to a tangent of the curve that is created by plotting the values in *VY* against the values in *VX*. The value at which the tangent is found is specified by the argument *POS*.

Returns the Y intercept of the linear equation (value of `a` in `Y = a + bX`).

If a *VX* or *VY* value at *POS* is an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
OFFSET = TANFITYINT(GATEV, DRAINI, GMMAX)
```

Also see

TANFIT (on page 5-284)
TANFITSLP (on page 5-285)
TANFITXINT (on page 5-286)

FFT

The fast Fourier transform (FFT) formulas convert a signal or sequential group of measurements from the time domain to the frequency domain. They can also be used to convert from the frequency domain to the time domain.

For additional detail on using the FFT formulas, refer to the application note “Using the Built-In FFT Functions to Calculate the Spectral Density, 1/F Noise, AC Impedance and Other Parameters with the 4200A-SCS Parameter Analyzer” (1KW-73767).

GET_FREQ Formulator function

Gets the fast Fourier transform (FFT) frequency for an input time list.

Usage

`GET_FREQ(time_source)`

time_source	The name of time column (vector) listed under Columns

Details

This Formulator function is only used in the 1/f noise test project.

Example

```
Fre1 = GET_FREQ(time1)
```

Also see

None

POWERFFT Formulator function

Performs the forward fast Fourier transform (FFT) for the data source input and return the power spectrum.

Usage

`POWERFFT(data_source, window)`

data_source	Value of the sample data (a list)
window	Name of the window function (integer); see **Details**

Details

This function is only used in the 1/f noise test project.

Value mappings of the parameter window:

Value	*window*	Function execution
1	rectangular	1
2	hanning	`w(n) = 0.5*(1 - cos(2 * pi * n/(N-1)))`
3	hamming	`w(n) = 0.54 - 0.46 * cos(2 * pi * n/(N-1))`
4	blackman	`w(n) = 0.42 - 0.5 * cos(2 * pi * n/(N-1)) + 0.08 * cos(4 * pi * n/(N-1))`
5	welch	`w(n) = 1 - ((n - 0.5 * (N - 1))/0.5 * (N+1))^2`

Example

```
pow1 = POWERFFT(IDrain, 2)
```

Also see

None

SMOOTH Formulator function

Smooth the vector using fast Fourier transform (FFT).

Usage

`SMOOTH (V, RANK)`

V	The name of any column (vector) listed under Columns
RANK	Rank number of smooth level

Details

If a *V* value contains an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
I2 = SMOOTH (I1, 2)
```

Also see

None

Misc

The Misc Formulator function allows you to compare user-defined expressions.

CURRENT_NOISE Formulator function

Calculates noise current from Ig.

Usage

`CURRENT_NOISE(IMEAS,POINT_NUM)`

IMEAS	Measured current
POINT_NUM	Number of points calculated

Example

```
Ig_noise = CURRENT_NOISE(Ig, 50)
```

Also see

None

DEGRADATION Formulator function

Calculates the degradation based on the first position.

Usage

`DEGRADATION(V)`

V	Column name

Example

`=DEGRADATION(I_GATE)`	

Also see

None

GET_TBD Formulator function

Gets breakdown (BD) time of the time-dependent dielectric breakdown (TDDB) test.

Usage

```
GET_TBD(IS, TIME, CUR_J, SLOPE_J, NOISE_J, FIX_CUR_J, SLOPE_P_NUM, NOISE_P_NUM,
    AND_OR, METHODS)
```

IS	Gate current
TIME	Test time
CUR_J	Criteria current times for BD; `|Ig[n]/Ig[0]|` ≥ *CUR_J*
SLOPE_J	Criteria slope times for BD; `|slope[n]/slope[n-1]|` ≥ *SLOPE_J*
NOISE_J	Criteria noise times for BD; `|Inoi[n]/Inoi[n-1]|` ≥ *NOISE_J*
FIX_CUR_J	Fixed cursor for criteria; `|Ig[n]|` ≥ *FIX_CUR_J*
SLOPE_P_NUM	Number of slope points for each step calculated; for example, if equal to 5, the slope of Ig is calculated every five points
NOISE_P_NUM	Number of noise points for each step calculated; for example, if equal to 5, the noise current of Ig is calculated every five points
AND_OR	`1`: `AND`, which indicates that all selected methods were achieved for BD `0`: `OR`, which indicates that any selected method was achieved for BD
METHODS	Indicates the four methods for BD criteria; `1234` means all four methods are chosen; `14` means only first and fourth are chosen

Details

You choose the attributes of the data you need.

It includes the following parameters:

- Two variables: *IS* and *TIME*
- Four methods for BD criteria: *CUR_J*, *SLOPE_J*, *NOISE_J*, *FIX_CUR_J*
- Two parameters for calculation: *SLOPE_P_NUM*, *NOISE_P_NUM*
- One logic relationship: *AND_OR*

Example

```
TBD = GET_TBD(Ig, time, 100, 50, 10000, 1E-4, 5, 5, 0, 14)
```

Using this example above and calculating TBD using Ig and time array along with the 1st and 4th methods (*CUR_J* and *FIX_CUR_J*), the following indicates that BD has been achieved. The criteria current time is 100, criteria slope time is 50, the criteria noise time is 10000, and the fixed cursor data for criteria is 1E-4. The slope of Ig is calculated every five points and noise current of Ig is calculated every five points.

Also see

None

GET_TBD2 Formulator function

Gets breakdown (BD) time of the time-dependent dielectric breakdown (TDDB) test.

Usage

```
GET_TBD2(IS, IS_TIME, ISILC, SILC_TIME, IS_CUR_J, ISILC_CUR_J, SLOPE_J, NOISE_J,
   FIX_CUR_J, SLOPE_P_NUM, NOISE_P_NUM, METHODS)
```

IS	Gate stressed current
IS_TIME	Stressed time
ISILC	Gate leakage current
SILC_TIME	Measure ISILC time
IS_CUR_J	Criteria current times for breakdown (BD); `|Ig[n]/Ig[0]|` ≥ *CUR_J*.
SLOPE_J	Criteria slope times for BD; `|slope[n]/slope[n-1]|` ≥ *SLOPE_J*
NOISE_J	Criteria noise times for BD; `|Inoi[n]/Inoi[n-1]|` ≥ *NOISE_J*
FIX_CUR_J	Fixed cursor for criteria; `|Ig[n]|` ≥ *FIX_CUR_J*
SLOPE_P_NUM	Number of slope points for each step calculated; for example, if equal to 5, the slope of Ig is calculated every five points
NOISE_P_NUM	Number of noise points for each step calculated; for example, if equal to 5, the noise current of Ig is calculated every five points
METHODS	Indicates the four methods for BD criteria; `1234` means all four methods are chosen; `14` means only first and fourth are chosen

Details

You choose the attributes of the data you need.

It includes the following parameters:

- Four variables: *IS*, *IS_TIME*, *ISILC*, *SILC_TIME*
- Four methods for BD criteria: *IS_CUR_J*, *SLOPE_J*, *NOISE_J*, *FIX_CUR_J*
- Two parameters for calculation: *SLOPE_P_NUM*, *NOISE_P_NUM*

Example

```
TBD = GET_TBD2(Ig, time_str, I_silc, time_silc, 100, 50, 10000, 1E-4, 5, 5, 0, 14)
```

Using the example above and calculating TBD time using two groups of arrays (`Ig` and `time_str`, `I_silc` and `time_silc`) along with the 1st and 4th methods (*IS_CUR_J* and *FIX_CUR_J*), the following indicates that BD has been achieved. The criteria current time is 100, criteria slope time is 50, the criteria noise time is 10000, and the fixed cursor data for criteria is 1E-4. The slope of Ig is calculated every five points and noise current of Ig is calculated every five points.

Also see

None

GET_VBD Formulator function

Gets breakdown (BD) voltage for the VRamp test.

Usage

`GET_VBD(IS, TIME, CUR_J, SLOPE_J, SLOPE_P_NUM, AND_OR, METHODS)`

IS	Variable; gate current
TIME	Variable; test time
CUR_J	Method for BD criteria; current times for breakdown; `|Ig[n]/Ig[0]| >= CUR_J`
SLOPE_J	Method for BD criteria; slope times for breakdown; `|slope[n]/slope[n-1]| >= SLOPE_J`
SLOPE_P_NUM	Parameter for calculation; number of slope points for each step calculated; for example, if equal to 5, slope of Ig is calculated every five points
AND_OR	`1`: AND, all selected methods were achieved for BD `0`: OR, any selected method was achieved for BD
METHODS	`12`: Selects both *CUR_J* and *SLOPE_J* methods for BD criteria `1`: Selects the first method for BD criteria

Details

You choose the attributes of the data you need.

Example

`VBD = GET_VBD(Ig, time, 100, 50,5, 5, 0,1)`
Using the example above and calculating VBD using two groups of arrays (Ig and time) along with the 1st method (*CUR_J*), the indicates that BD has been achieved. The criteria current time is 100 and criteria slope time is 50. The slope of Ig is calculated every five points.

Also see

None

LOG10 Formulator function

Returns the base-10 log of each value in a designated column (vector) or the base-10 log of any operand.

Usage

`LOG10(V)`

V	The name of any column (vector) listed under Columns or any operand

Details

You can use this function to do calculations in real time (while a test is executing).

Example

```
F2 = LOG10(I_DRAIN)
```

Also see

None

MOBILITY Formulator function

Mobility at Vg_tar by Id-Vg curve.

Usage

MOBILITY(*w*, *l*, *ci*, *Vg_tar*, *Vth*, *Id*, *Vg*)

w	Width of device gate
l	Length of device gate
ci	Capacitance of gate
Vg_tar	Target VG
Vth	Threshold voltage
Id	Measured Drain current
Vg	Measured Gate voltage

Details

Vg_tar should be in the saturation region of the MOSFET.

Example 1

```
U1 = MOBILITY(10, 0.03, 1, 0.05, 0.078, Id, Vg)
```

Example 2

```
MOBILITY = dIds/dVg *1/Vd * L/W * 1/Cox
```

Example 3

When MOSFET goes into saturation region, Vd = Vg – Vth, use the following formula.

```
MOBILITY = 2 * Id_tar@Vg_tar/(Vg_tar-Vth)2 * L/W * 1/Cox
```

Also see

None

NOISE_UTIM Formulator function

Generates correct time array for noise current.

Usage

`NOISE_UTIM(`*`MEASTIMES`*`, `*`POINT_NUM`*`)`

MEASTIMES	Measurement time vector
POINT_NUM	Number of points calculated

Example

```
NoiseTime = NOISE_UTIM(MeasTime, 80)
```

Also see

None

POW Formulator function

Returns the power value of each value in a designated column (vector) or any operand.

Usage

`POW(`*`VBASE`*`, `*`VPOW`*`)`

VBASE	The name of any column (vector) listed under Columns, or a number as base value
VPOW	The name of any column (vector) listed under Columns, or a number as power value

Details

This function cannot accept two arrays in its parameters.

Example

```
I2 = POW(I_DRAIN, 2)
I2pow = POW(2, I_DRAIN)
```

Also see

None

RES Formulator function

Calculates the resistance vector value by giving V and I.

Usage

`RES(V, I))`

V	Voltage vector listed in variables box
I	Current vector listed in variables box

Details

If a V or I value contains an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
RG_1 = RES(V_Gate_1, I_Gate_1)
```

Also see

None

RES_4WIRE Formulator function

Calculate 4-wire resistance vector value by giving V and I.

Usage

`RES_4WIRE(VPOS, VNEG, I)`

VPOS	Positive voltage vector listed in variables box
VNEG	Negative voltage vector listed in variables box
I	Current vector listed in variables box

Details

If *VPOS*, *VNEG*, or *I* value contains an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
RGD = RES_4WIRE(V_Gate_1, V_Drain_1, I_Gate_1)
```

Also see

None

RES_4WIRE_AVG Formulator function

Calculates 4-wire resistance average value using V and I.

Usage

`RES_4WIRE_AVG(VPOS, VNEG, I)`

VPOS	Positive voltage vector listed in variables box
VNEG	Negative voltage vector listed in variables box
I	Current vector listed in variables box

Details

If *VPOS*, *VNEG*, or *I* value contains an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
RGD_AVG = RES_4WIRE_AVG(V_Gate_1, V_Drain_1, I_Gate_1)
```

Also see

None

RES_AVG Formulator function

Calculates resistance average value using V and I.

Usage

`RES_AVG(V, I)`

V	Voltage vector listed in variables box
I	Current vector listed in variables box

Details

If a *V* or *I* value contains an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
RG_1 = RES_ AVG (V_Gate_1, I_Gate_1)
```

Also see

None

RSQUAR Formulator function

Calculates correlation coefficients from an input matrix *VX* and *VY*.

Usage

`RSQUAR(VX, VY, START_POS, END_POS)`

VX	Column vector
VY	Column vector
START_POS	Start position to do correlation
END_POS	End position to do correlation

Example

```
R1 = RSQUAR(V_Pos, I_Pos, 5, 20)
```

Also see

None

SS Formulator function

Extracts the subthreshold swing between a certain duration of current.

Usage

`SS(I_DRAIN, V_GATE, START_I, STOP_I)`

I_DRAIN	Drain current measured
V_GATE	Gate voltage forced
START_I	Start current to define the calculation region
STOP_I	Stop current to define the calculation region

Details

The subthreshold swing is defined as the difference between Vg and log Id. For example, `SS = dVg/dlogId`.

If an *I_DRAIN* or *V_GATE* value contains an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
S0 = SS (I_Drain_1, V_Gate_1, 1.3e-5, 1.5e-5)
```

Also see

None

SSVTCI Formulator function

Extracts the sub-threshold swing for a specific duration between left offset and right offset of the constant current threshold voltage.

Usage

`SSVTCI(I_DRAIN, V_GATE, VTCI, LEFT_OFFSET, RIGHT_OFFSET)`

I_DRAIN	Drain current measured in test
V_GATE	Gate voltage
VTCI	Constant current threshold voltage
LEFT_OFFSET	Left offset of the constant current threshold voltage
RIGHT_OFFSET	Right offset of the constant current threshold voltage

Details

If a *I_DRAIN* or *V_GATE* value contains an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
SLOPE = SSVTCI(I_Drain_1, V_Gate_1, VTH, -0.1, -0.1)
```

Also see

None

TAR_YatX Formulator function

Finds `y_tar` in `y` corresponding to `x_tar` in the `x` index.

Usage

`TAR_YatX(x, y, x_tar)`

x	X array
y	Y array
x_tar	Target x

Example

```
y0 = TAR_YatX(V, I, 0.1)
```

Also see

None

TTF_DID_LGT Formulator function

Finds the time to failure (TTF) by analyzing two arrays in percentage format (time array and degradation).

Usage

`TTF_DID_LGT(DID, T, START_POS, END_POS, GOAL)`

DID	Drain current degradation (in percentage) listed under Columns
T	Time array column name
STARTPOS	Range of *T* and *DID* values to be plotted, the row number or position (index) of the starting values
ENDPOS	Range of *T* and *DID* values to be plotted, the row number or position (index) of the ending values
GOAL	Degradation value goal (in percentage)

Details

In HCI test, you may want to analyze two arrays, time array and degradation (in percentage), and find the specific time corresponding to the aimed degradation. This is accomplished by choosing different fitting models.

In this Formulator function, the fitting relationship is built on time array and log value of degradation of drain current.

For instance, you may want to analyze the HCI test results and find the specific time corresponding to the degradation target. This can be accomplished by using discrete data points to get a curve formula that shows a trend and helps predict future data points. The data points are built on the time array and log value of the drain current measurement.

If a *T* or *DID* value at *STARTPOS* or *ENDPOS* contains an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
F1 = TTF_DID_LGT (I_DRAIN_DEG, TIME, 5, 11, 10)
```

Also see

None

TTF_DID_T Formulator function

Finds the time to failure (TTF) by analyzing two arrays in percentage format (time array and degradation).

Usage

`TTF_DID_T(DID, T, START_POS, END_POS, GOAL)`

DID	Drain current degradation (in percentage) listed under Columns
T	Time array column name
START_POS	Range of *T* and *DID* values to be plotted, the row number or position (index) of the starting values
END_POS	Range of *T* and *DID* values to be plotted, the row number or position (index) of the ending values
GOAL	Degradation value goal (in percentage)

Details

In HCI test, you may want to analyze two arrays, time array and degradation (in percentage), and find the specific time corresponding to the aimed degradation. This is accomplished by choosing different fitting models.

In this Formulator function, the fitting relationship is built on time array and log value of degradation of drain current.

For instance, you may want to analyze the HCI test results and find the specific time corresponding to the degradation target. This can be accomplished by using discrete data points to get a curve formula that shows a trend and helps predict future data points. The data points are built on the time array and log value of the drain current measurement.

If a *T* or *DID* value at *START_POS* or *END_POS* contain an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
F3= TTF_DID_T (I_DRAIN_DEG, TIME, 5, 11, 10)
```

Also see

None

TTF_LGDID_LGT Formulator function

Finds the time to failure (TTF) by analyzing two arrays in percentage format (time array and degradation).

Usage

`TTF_LGDID_LGT(DID, T, START_POS, END_POS, GOAL)`

DID	Drain current degradation (in percentage)
T	Time array column name
STARTPOS	Range of *T* and *DID* values to be plotted, the row number or position (index) of the starting values
ENDPOS	Range of *T* and *DID* values to be plotted; the row number or position (index) of the ending values
GOAL	Degradation value goal (in percentage)

Details

In HCI test, you may want to analyze two arrays, time array and degradation (in percentage), and find the specific time corresponding to the aimed degradation. This is accomplished by choosing different fitting models.

In this Formulator function, the fitting relationship is built on time array and log value of degradation of drain current.

For instance, you may want to analyze the HCI test results and find the specific time corresponding to the degradation target. This can be accomplished by using discrete data points to get a curve formula that shows a trend and helps predict future data points. The data points are built on the time array and log value of the drain current measurement.

If a *T* or *DID* value at *STARTPOS* or *ENDPOS* contains an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
F4 = TTF_LGDID_LGT(I_DRAIN_DEG, TIME, 5, 11, 10)
```

Also see

None

TTF_LGDID_T Formulator function

Finds the time to failure (TTF) by analyzing two arrays in percentage format (time array and degradation).

Usage

`TTF_LGDID_T(DID, T, START_POS, END_POS, GOAL)`

DID	Drain current degradation (in percentage)
T	Time array column name
STARTPOS	Range of *T* and *DID* values to be plotted, the row number or position (index) of the starting values
ENDPOS	Range of *T* and *DID* values to be plotted; the row number or position (index) of the ending values
GOAL	Degradation value goal (in percentage)

Details

In HCI test, you may want to analyze two arrays, time array and degradation (in percentage), and find the specific time corresponding to the aimed degradation. This is accomplished by choosing different fitting models.

In this Formulator function, the fitting relationship is built on time array and log value of degradation of drain current.

For instance, you may want to analyze the HCI test results and find the specific time corresponding to the degradation target. This can be accomplished by using discrete data points to get a curve formula that shows a trend and helps predict future data points. The data points are built on the time array and log value of the drain current measurement.

If a *T* or *DID* value at *STARTPOS* or *ENDPOS* contain an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
F2 = TTF_LGDID_T (I_DRAIN_DEG, TIME, 5, 11, 10)
```

Also see

None

VTCI Formulator function

Finds the threshold voltage at target drain current using current density, device width, and device length.

Usage

`VTCI(u, w, l, I_DRAIN, V_GATE)`

u	Current density
w	Device width
l	Device length
I_DRAIN	Drain current measured
V_GATE	Gate voltage measured

Details

If the drain current measured and gate voltage measured values at the data point position contain an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
VTH1 = VTCI(0.001, 1, 0.03, I_DRAIN, V_GATE)
```

Also see

None

VTEXLIN Formulator function

Extrapolates the threshold voltage from the measurement of the maximum slope for the drain current and gate voltage curve using 0.5 times bias drain voltage. For example, `VTEXLIN = VTLINGM - 0.5 x VdLin`.

Usage

`VTEXLIN(Id, Vg, Vd_Lin)`

Id	Measured drain current
Vg	Measured gate voltage
Vd_Lin	

Example

```
VTH1 = VTEXLIN(Id, Vg, 2.4)
```

Also see

None

VTLINGM Formulator function

Extrapolates threshold voltage from the measurement of the maximum slope for the drain current and gate voltage curve in the linear region of a MOSFET.

Usage

`VTLINGM(I_DRAIN, V_GATE)`

I_DRAIN	Drain current measured in the test
V_GATE	Gate voltage measured

Details

DC bias voltages for Vth measurements are:

- VDS = VDS_lin
- VBS = VBB

Typically, for PMOS, `VDS_lin=-0.1V(@VDD=5V)`; for NMOS, `VDS_lin=0.1V(@VDD=5V)`.

For example, `Vth = VGS(@GMMAX) - ID(@GMMAX)/GMMAX`

Where:

- `VGS(@GMMAX)` is the gate voltage at the point of the maximum slope for the ID-VGS curve
- `ID(@GMMAX)` is the drain current at the point of the maximum slope for the ID-VGS curve
- `GMMAX` is the maximum slope for the ID-VGS curve

If the drain current measured and gate voltage measured values contain an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
VTH0 = VTLINGM(I_DRAIN, V_GATE)
```

Also see

None

VTSATGM Formulator function

Extrapolates threshold voltage from the measurement of the drain current and gate voltage curve in the saturation region of a MOSFET.

Usage

`VTSATGM(I_DRAIN, V_GATE)`

I_DRAIN	Measured drain current
V_GATE	Measured gate voltage

Detail

For example, `Vth = VGS(@GMMAX) - SQRT(ID)(@GMMAX))/ GMMAX`.

Where:

- `VGS(@GMMAX)` is the gate voltage at the point of the maximum slope for the `SQRT(ID)-VGS` curve
- `SQRT(ID)(@GMMAX)` is the square root of the drain current at the point of the maximum slope for the `SQRT(ID)-VGS` curve
- `GMMAX` is the maximum slope of the `SQRT(ID)-VGS` curve
- `SQRT(ID)` is the square root of ID
- DC bias voltages for Vth saturation measurements are `VDS = VGS = VDD`, `VBS = VBB`

If drain current measured and gate voltage measured values at the data point position contain an invalid number (that is, the value is `#REF`), the function will not return a valid result.

Example

```
VTH1 = VTSATGM(I_DRAIN, V_GATE)
```

Also see

None

VT_SAT_TRI Formulator function

Extrapolates threshold voltage by triple dividing the measured drain current root curve.

Usage

`VT_SAT_TRI(Vg_tar, Id, Vg)`

Vg_tar	Gate voltage target
Id	Measured drain current
Vg	Measured gate voltage

Example

```
VT0 = VT_SAT_TRI(0.08, Id, Vg)
```

Also see

None

Statistics

The following Formulator functions provide statistics operations.

AVG Formulator function

Returns the average of all values in the column (vector).

Usage

`AVG(Value)`

Value	The name of any column (vector) in the Data Series list

Example

```
LEAKAGE = AVG(GATEI)
```

Also see

MAVG (on page 5-310)

GMMAX Formulator function

Returns the maximum value from two columns (vectors).

Usage

```
GMMAX(Id, Vg)
GM = DIFF(Id, Vg)
```

Id	The current drain vector
Vg	The voltage gate vector

Example

```
Neg = GMMAX(DIFF(I_drain, V_gate))
```

Also see

None

MAVG Formulator function

Provides moving averages for the vector.

Usage

```
MAVG(V, N)
```

V	Column name
N	Window size

Details

If *N* = 3 and *V* contains the 12 values `X1`, `X2`, `X3`, `X4`, `X5`, `. . .` , `X10`, `X11`, `X12`, then `MAVG` returns a column (vector) that contains the following values:

```
#REF, (X1 + X2 + X3)/3, (X2 + X3+ X4)/3, (X3 + X4+ X5)/3, ... (X10 + X11+ X12)/3,
    #REF
```

The new column's values may contain instances of `#REF` (as shown above) because `MAVG` uses cells from both sides of the target cell for its calculation.

Example

```
FILTER = MAVG(GATEI, 3)
```

Also see

None

MAX Formulator function

Searches all values in a column (vector) and returns the maximum value.

Usage

`MAX(Value)`

Value	The name of any column (vector) in the Data Series list

Example

`MAXGM = MAX(DIFF(DRAINI, GATEV))`	

Also see

None

MEDIAN - Formulator function

Calculates the average value between the start and stop index.

Usage

`MEDIAN(Value, start_pos, end_pos)`

Value	Column name
start_pos	The starting value
end_pos	The ending value

Example

`avg1= MEDIAN(V, 1, 20)`	Calculates the average value between 1 and 20.

Also see

None

MIN Formulator function

Searches all values in a column (vector) and returns the minimum value.

Usage

`MIN(Value)`

Value	The name of any column (vector) in the Data Series list

Example

`SMALLESTI = MIN(DRAINI)`	

Also see

None

User Access Points (UAP)

The User Access Point (UAP) is a user-friendly management interface used to configure all test features. UAPs are supported to let users extend the features and functions of the Keithley-provided test execution engine at the device, subsite, site, wafer, and cassette levels. This is accomplished by executing either PTM (Python Test Module) or CTM (C language Test Module) at the entry or exit of each level of automation. PTM and CTM code has access to the results (data collected) and instrumentation (see next two figures).

Figure 340: UAP execution example

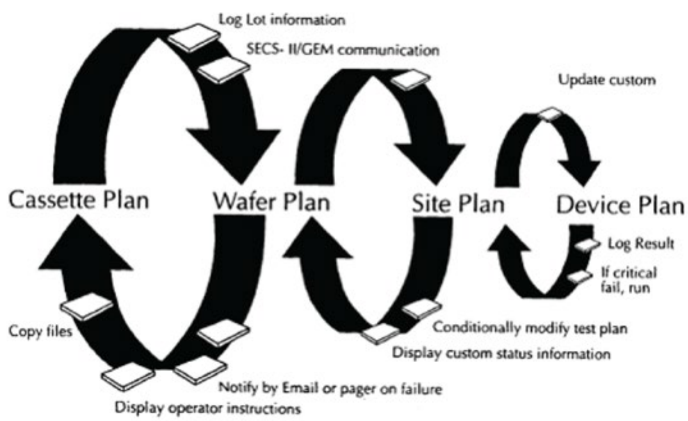


Figure 341: Test execution engine

Cassette Level	***Wafer Level***	***KTM Level***
Initialize engine Get command line args Load cassette plan ***UAP_PROG_ARGS*** Get lot id **UAP_CASS_LOAD** Collect lot information ***UAP_LOT_INFO*** Initialize prober for cassette ***UAP_PROBER_INIT*** ***UAP_WRITE_LOT_INFO*** Wafer loop ***Wafer Level*** End of wafer loop Log end of lot ***UAP_LOT_END*** Close up execution engine ***UAP_ENGINE_EXIT*** Exit KTXE	***UAP_WAFER_PREPARE*** Load next wafer plan Load probe card file Load limits file Load wafer desc. file Load all ktms Prepare wafer work plan Load wafer on prober Execute wafer plan ***KTM Level*** ***UAP_WAFER_END*** End of wafer-plan execution	Move to next site ***UAP_SITE_CHANGE*** Move to next subsite ***UAP_SUBSITE_CHANGE*** Execute ktms ***UAP_TEST_BEGIN*** Execute ktm ***UAP_TEST_END*** Log data ***UAP_TEST_DATA_LOG*** Check for abort ***UAP_HANDLE_ABORT*** Call endsite

With UAPs you can perform several robust tasks:

- Display user instructions
- Log lot information
- Notified by text message or e-mail to pager or cell phone of a failure or exit condition
- Display custom status information
- Log results
- Format and write custom results files

To customize the test execution environment:

1. Create a C-code or Python module.
2. Pair the modules with user access points (UAP). User access points begin at the end of…Programs, Lots, Wafers, Sites, and Subsites.

Handling the UAP tree

Several UAPs are provided, such as:

lot_begin	lot_end	wafer_begin	wafer_end
site_begin	site_end	subsite_begin	subsite_end
device_begin	device_end	test_begin	test_end

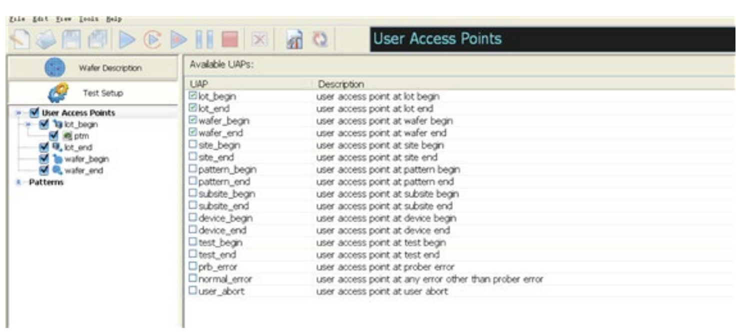
To add a UAP:

1. To access the UAP tree, right-click the configuration navigator in the Test Setup panel. A display menu appears.
2. Select the Show UAP Tree menu, as shown in the following figure.

Figure 342: Show UAP Tree

3. To add a UAP, select the corresponding UAP, as shown in the following figure.

Figure 343: UAP Test tree setup



Editing a UAP

To edit a UAP:

1. Select a UAP in the UAP tree; an editing panel appears.
2. Select **Enable this Function**. The Shell Command is enabled during the automation test. You can also choose the execution timing of the Shell Command (before or after the nested modules). The following directory is called before all nested modules, as shown in the following figure.

Figure 344: UAP Shell Command editing panel

3. Two modules are available: Select New CTM adds a C module (the same as CTM); select New PTM adds a Python module. Select the modules in the UAP tree. The edit panel of that module appears. Once all the modules and Shell Commands are set, go to the Automation function, and select **RUN** (see the next two figures).

Figure 345: UAP CTM editing panel

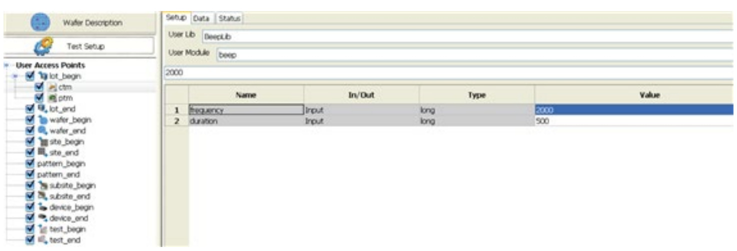
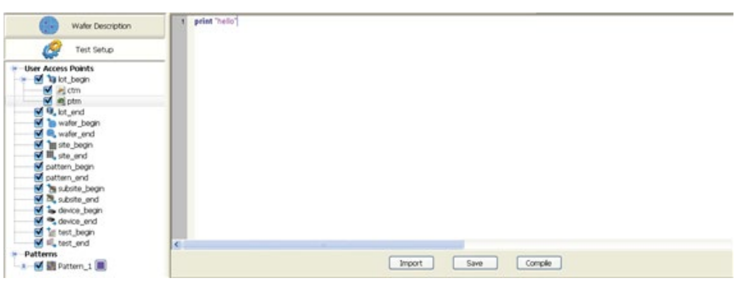


Figure 346: UAP PTM editing panel; edit in Python language



The following table provides the UAP global variable name, type, and the associated comment. These can be accessed in a UAP PTM.

UAP global variables

Variable Name	Type	Comment
ACS_frame	string	the frame of ACS
ACS_uap_level	string	current UAP level
ACS_proj_path	string	current path
ACS_proj_name	string	current project name
ACS_lot_id	string	current lot ID
ACS_slot_no	int	current slot number
ACS_wafer_id	string	current wafer ID
ACS_pattern_id	string	current pattern ID
ACS_site_id	string	current die ID
ACS_site_coord	tuple	current die coordinates
ACS_ssite_id	string	current subsite ID
ACS_ssite_loop	int	current subsite index of subsite loop
ACS_ssite_loopNum	int	total loop num of the current subsite
ACS_ssite_coord	tuple	current die coordinates
ACS_dut_id	string	current DUT ID
ACS_test_id	string	current test ID
ACS_test_data	dictionary	current test data
ACS_kdf_head	dictionary	kdf file header information
ACS_kdf_data	list	current wafer data
ACS_wdf_head	dictionary	wdf header information
ACS_wdf_nonexist	dictionary	wdf non-exist dices dictionary
ACS_t_start	string	test start time
ACS_t_end	string	test end time
ACS_first_wafer	bool	whether it is the first wafer in one cassette
ACS_last_wafer	bool	whether it is the last wafer in one cassette
ACS_first_site	bool	whether it is the first site in one wafer
ACS_last_site	bool	whether it is the last site in one wafer
ACS_output_list	list	all tests output list
ACS_temp_klf	string	temporary klf file location
ACS_prb_errcode	int	prober error code
ACS_operator	string	user name
ACS_die_type	string	die type
ACS_remark	string	remark in automation panel
ACS_session_name	string	session name in automation panel
ACS_equipment_id	string	equipment in automation panel
ACS_fixture_id	string	fixture id in automation panel
ACS_testplan_ver	string	test plan version in automation panel
ACS_process_level	string	process level in automation panel
ACS_userdata_path	string	full path of user data
ACS_ptm_path	string	PTM directory setting in preference
ACS_site_loop	int	current sub site index of subsiteloop
ACS_site_loopNum	int	total loop num of the current site
ACS_ktxe_exit	int	exit the execution of ktxe
ACS_ktxe_loop_wafer	int	current wafer loop flag
ACS_run_mode	string	indicate automation or manual run

ACS_Stress_Time	float	The accumulated time for the stress measure loop. Changed to different value according to subsite loop number.
ACS_Stress_Duration	float	Stress duration time for each subsite loop.
ACS_Disable_Subsite_List	list	If subsite name is in this list, the subsite will be skipped.
ACS_ Disable_DUT_List	list	If device name is in this list, the device will be skipped.
ACS_Disable_Test_List	list	If test name is in this list, the test will be skipped.

Add an attribute using UAP

In ACS, you can set attributes for the test setup by using a UAP. An attribute adds further analysis to the ACS test results. An attribute can be any of the following items:

- Any data point within a sweep has attribute sweep variables; for a family of curves, the step value is an attribute
- Device related parameters, such as W, L, Area, and so on
- Global information, such as temperature, pressure, and any nonzero stress conditions
- Physical locations can be considered an attribute, such as HCI, the W, L, and VDrain

There are two ways to set an attribute and it depends on the test module type. For example, an RTM or other types of test modules. The following will give you a detailed introduction on how to set an attribute for these types.

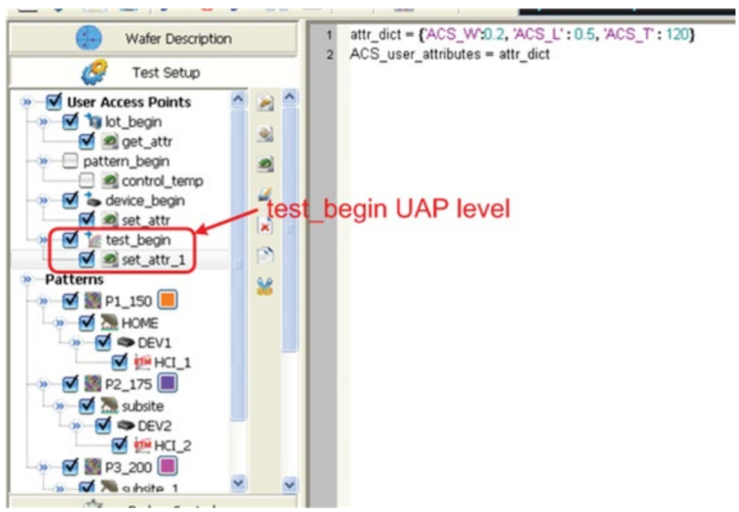
Add RTM attribute

For the RTM, you can use the ACS use attributes. The UAP can be added at the test begin level, as shown in the following figure.

To add an attribute for RTM

1. Assign an ACS user attribute. For example, the following figure shows a fixed attribute value: attr_dict={'ACS_W':0.2, 'ACS_L':0.5,'ACS_T':120}.
2. Format the attribute as {'attribute_name1':attribute_value1, 'attribute_name2':attribute_value2,…}. The attribute_value1 must be single data.

Figure 347: Set attribute UAP for RTM



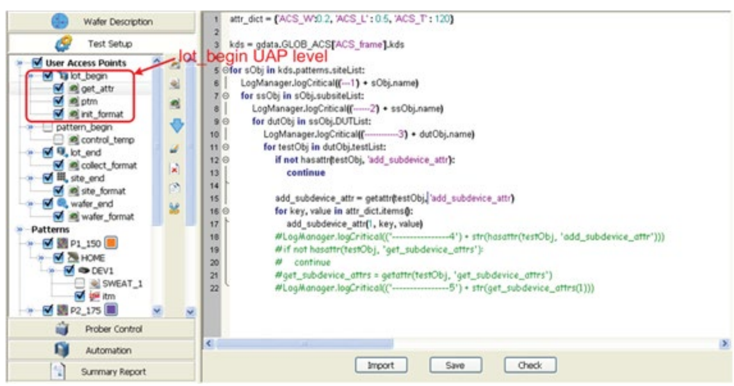
Add an attribute to test modules

Except for RTMs, you can add attributes to ITMs, STMs, CTMs, and PTMs. In this case, the UAP must be under the lot_begin UAP level. There is also an example showing how to set an attribute.

To add an attribute to other test modules

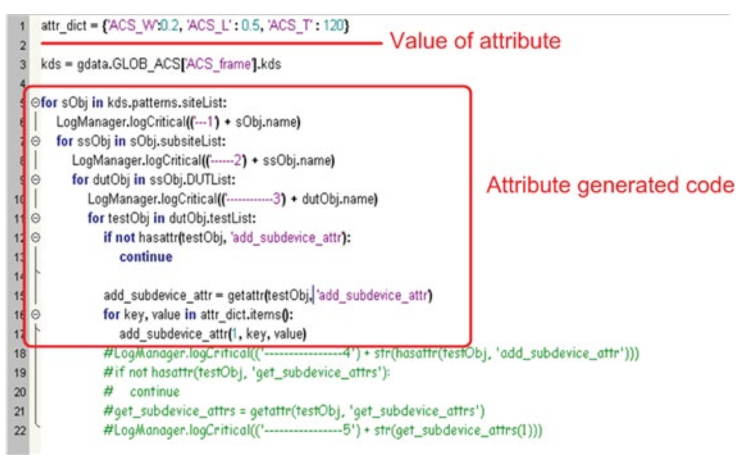
1. Open this UAP, you will see the code lines. As shown in the following figure, the first line is for setting a fixed attribute value: attr_dict={'ACS_W':0.2, 'ACS_L':0.5,'ACS_T':120}.
2. Format the attribute value as {'attribute_name1':attribute_value1, 'attribute_name2':attribute_value2,…}. The attribute_value1 must be single data.

Figure 348: UAP tree for other test modules



The lines indicated in the following figure are attribute generated code. They have the same function as other attribute values, as shown in the following figure.

Figure 349: Attribute setting information



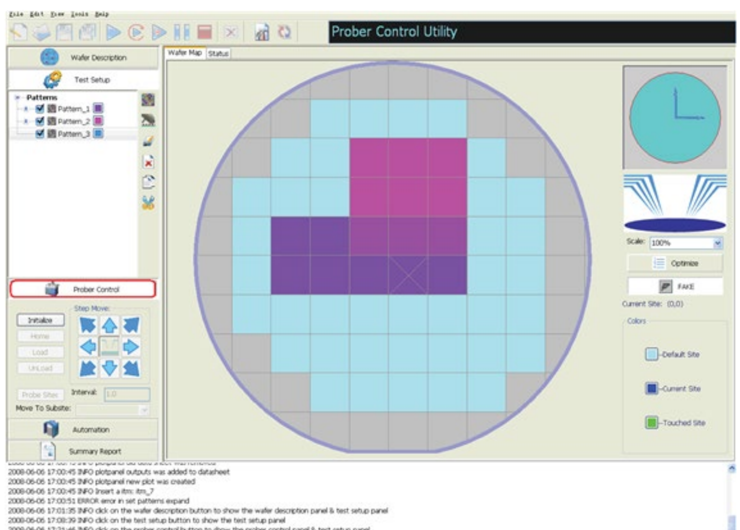
Prober control

In this section:

Prober control functions

The prober control functions allow you to control prober station movements and display the prober sites information in real time. Select **Prober Control** from the edit panel. The Wafer Map tab opens in the work area, as shown in the following figure.

Figure 350: Prober Control GUI



Wafer map indicator in Prober Control mode

In prober control mode, the wafer map indicator opens in the work area, as shown in the following figure.

Figure 351: Wafer Map Indicator in Prober Control mode

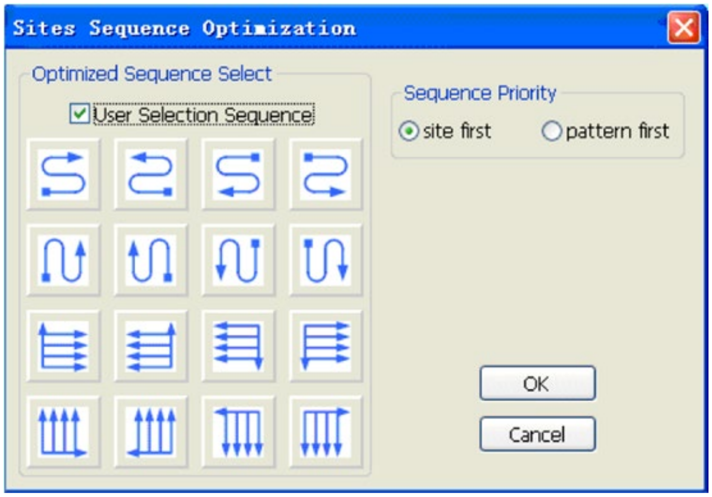
The wafer map indicator:

- Optimizes the order of sites that are to be probed.
- Sets the evaluation or standard mode of the prober station.
- Sets and shows the color settings.
- Shows the chuck-up or chuck-down state of the prober.
- Shows the coordinates of the current site.

To optimize the sites list:

1. Select **Optimize**. The Site Sequence Optimization dialog opens, as shown in the following figure.

Figure 352: Sites Sequence Optimization



2. You can choose from sixteen different sequences. Also, the default Sequence Priority is set to site first, as shown in the following figure.

If you want to print a pattern ID in the log window display, you will need to select pattern first in the sequence priority (see Automation functions (on page 7-1) for more details).

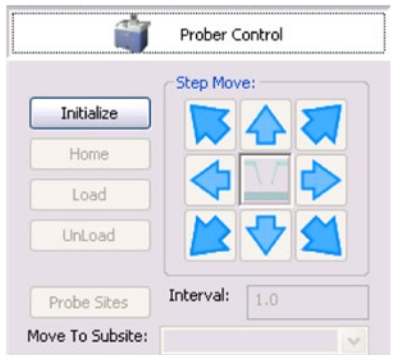
When a sequence is selected, the tester starts at the site closest to the chosen location, and then proceeds to all selected sites on the wafer, following the selected serpentine sequence. Select **OK** when finished.

- **Fake:** Sets the evaluation or standard mode of the prober station configured using the configure hardware command in the Tools menu. You can toggle between the two different modes. When the configured prober model is offline, the function is displayed as FAKE. When online, the function is the model name, for instance, TSK9.
- **Chuck Up/Chuck Down:** Indicates the raised or lowered chuck state of the prober.
- **Current Site: (X; Y):** Indicates the coordinates of the present site captured by the mouse movement.
- **Colors:** Shows and sets the color of sites. There are three status indicators: Default site, Current site and Touched site.
- **Default:** Shows and elects the color of the default setting.
- **Current:** Shows and selects the color of the present site of the prober.
- **Touched:** Shows and selects the color of sites that have already been touched by the prober.

Prober control panel

To access the prober control panel, select **Prober Control** from the edit panel.

Figure 353: Prober Control panel



- **Initialize:** Select this function to initialize the specified prober station. When the prober is initialized, all the prober control panel functions are enabled. If the initialization fails, all the functions are disabled.
- **Home:** This function is used to move the prober to the target site defined in the wafer description.
- **Load:** This function is used to load a wafer from the cassette onto the prober station.
- **UnLoad:** This function is used to unload the wafer from the prober station back to the cassette.
- **Probe Sites:** This function is used to move the prober to the selected sites continuously in the optimized sequence.
- **Interval:** Determines the time interval between two continuous movements of the prober. The time unit is in seconds.
- **Step Move:** There are eight blue arrow functions in Step Move, which can be used to control the prober station to move in different directions. Also, the green Chuck Up/Down function is used to control the prober station to raise or lower the chuck.

NOTE

The next two topics must be done in the following order Move to a site (on page 6-5), then Move to a subsite (on page 6-5).

Move to a site

To move to a site:

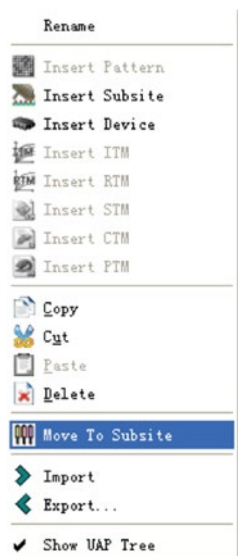
1. Select **Initialize** on the Prober Control panel. If the prober is configured correctly, there will be no errors and the other functions on the prober control panel enable.
2. Double-click the site you want to move to on the wafer map, or you can select the arrow functions on step move until you drive the prober to the destination site.

Move to a subsite

To move a subsite:

1. Move to the site on the wafer map.
2. From the configuration navigator, locate the site you moved to and right-click the subsite.
3. Select **Move to Subsite**, as shown in the following figure.

Figure 354: Right-click Menu of Subsite Node in Test Setup



Section 7

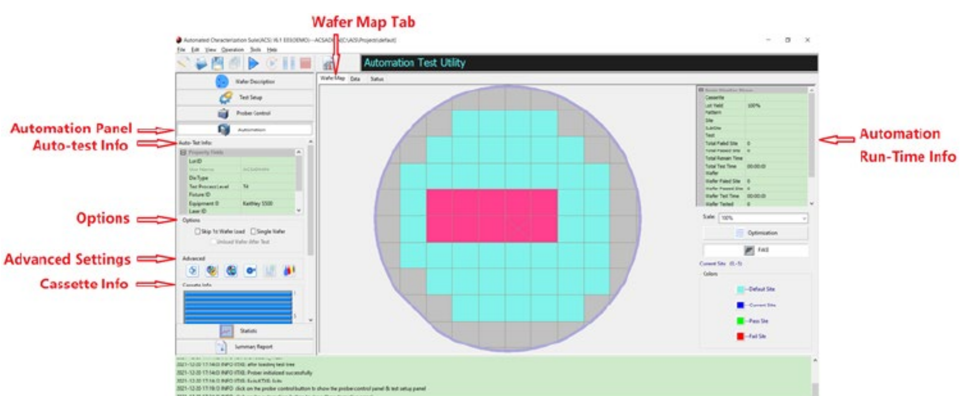
Automation

In this section:

Automation functions

After your configuration navigator has been set up and your wafer has been drawn, you can enter Automation mode to test all wafers in a cassette. To enter Automation mode, select **Automation** from the edit panel. The Automation panel displays (see the following figure). The Automation functions are discussed below.

Figure 355: Automation GUI



There are three tabs in the automation dialog:

- The Wafer Map tab is the primary interface and allows you to configure an automation test and display the present configuration and automation runtime information.
- The Data tab displays the test results in spreadsheet format and in real time as the test executes.
- The Status tab monitors the configuration status of automation. The Automation functions include:
  - **Auto-test information:** Auto-Test information is used to input some information related to the automation test. The information items include the following: Lot ID, User Name, Die Type, Test Process Level, Fixture ID, Equipment ID, Laser ID, Test Plan Version, and Remark.
  - **Options:** Used to customize the automation process. There are three options: Skip 1st Wafer Load, Single Wafer, and Unload Wafer after Test.
  - **Cassette information:** Used to indicate the wafer mounted status in the cassette, and wafer test status during automation, which slot is under the testing process and which slot has been tested completely. The Bottom-up option specifies how wafers are arranged in the cassette. By default, the arrangement of wafers is from top to bottom. You can reverse the arrangement by selecting the **Bottom-up** option.
  - **Runtime information:** Used to indicate the runtime information, present site, present subsite, and present test module, and so on.

Automation panel

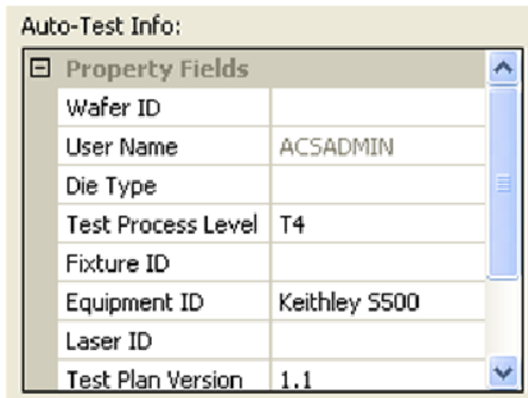
While in the Automation panel, select **RUN** to execute an automation test. The automation test runs through the entire cassette. Test progress is refreshed in real time in the Run Time information table, which is in the upper-right corner. The following section describes the parts of the automation panel.

Figure 356: Automation panel

Automatic test information

The Auto-Test information table allows you to input some necessary details related to the automation test process, such as: Lot ID, User Name, Die Type, Test Process Level, Fixture ID, Equipment ID, Laser ID, Test Plan Version, and Remark field. This information can be accessed in a UAP to generate customized data files.

Figure 357: Auto-Test Info box



You can see in the previous figure that the Lot ID, User Name, Die Type, Test Process Level, Fixture ID, Equipment ID, Laser ID, Test Plan Version, and Remark items have been defined.

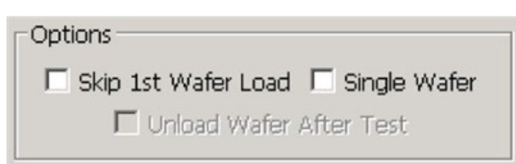
NOTE

All items are for information logging only, except two required fields: Lot ID and User Name. Information logging is optional. The meaning of the information logging can be user-defined, and will have no impact on the data file or other ACS operations.

Automatic test options

ACS provides three option to customize the automation process: Skip 1st Wafer Load, Single Wafer, and Unload Wafer After Test.

Figure 358: Options in the Automation Panel



- **Skip 1st Wafer Load:** Applicable for some prober Models such as TEL, which requires the first wafer on chuck to finish the manual alignment operation before the launch the automation test. Check this function when the first wafer is already loaded.
- **Single Wafer:** This tests the first wafer only. The load and unload commands will not be performed when the Single Wafer is selected. This is specially designed for semi-automated probers.
- **Unload Wafer After Test:** Only applicable when a single wafer is selected. Checking this box unloads the wafer automatically upon completion of the wafer test.

Advanced toolbar

The Advanced toolbar is used for advanced automation control features and configurations.

Figure 359: Advanced toolbar

Advanced Settings: Used to define advanced wafer settings.

Binning Settings: Defines and enables real-time binning. The binning result displays in the wafer map during automation.

Wafer Level Plot: Offline binning map, raw data, and plotting curve.

Real Time Plot Settings: Defines the critical test items in test plan that should be monitored in real time.

Show Real Time Plot Window: Activates when the Realtime Plot setting is configured. When selected, predefined plotting curves are updated in real time.

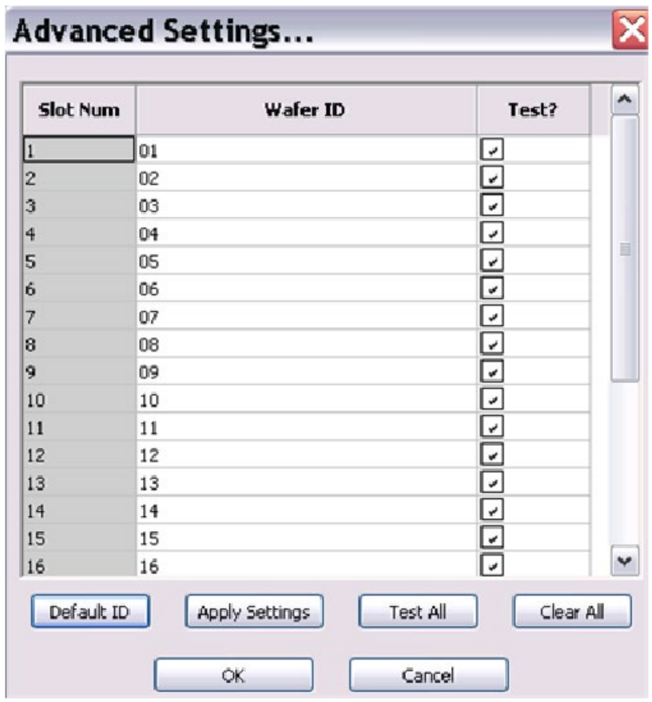
Post-inking Settings: Enables the post-inking function.

Advanced Settings

In the Advanced Settings dialog, you can rename the wafer ID and define the wafer to be tested in automation.

When you select **Advanced Settings**, the software automatically retrieves the slot status from the prober and shows all in-slot wafers in the Advanced Settings table, as shown in the following figure.

Figure 360: Advanced Settings

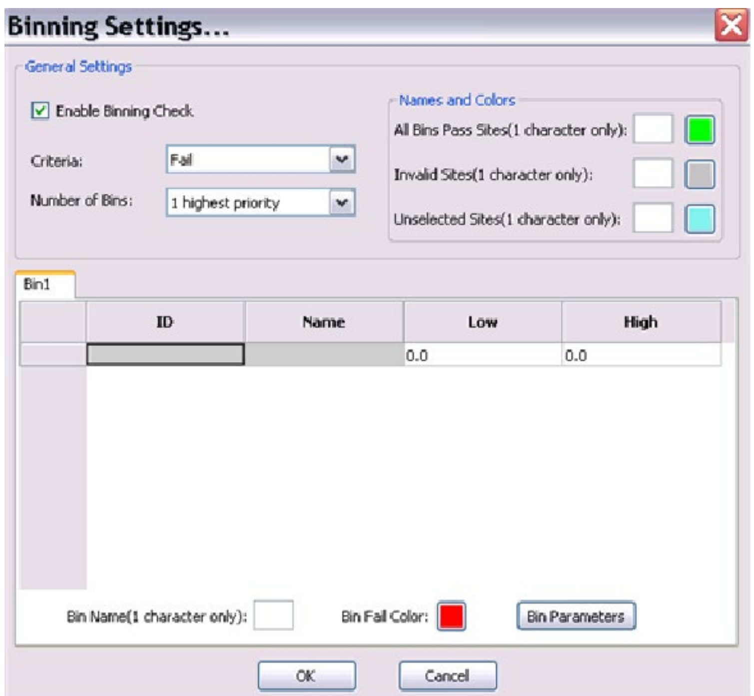


- **Default ID:** This function sets all wafer IDs (column Wafer ID) to the default value. The default value is 01 to 25 (depending on the previous information loaded).
- **Apply Settings:** The wafer ID can be edited. This function enables all wafer IDs to the customized value after editing them.
- **Test All:** This function selects all the in-slot wafers (Test Column). Once selected, all the wafers are ready for testing.
- **Clear All:** This function is used to clear all the in-slot wafers (Test Column). All the wafers are cleared from testing.

Binning Settings

After running the automation test, the binning method measures certain user-specified parameters (limits) against the results of each semiconductor device test, and then places the device test into an appropriate category (or "bin"). For example, if the device tests below the user-specified limit, the device would be placed into a bin labeled, "Failed" (see the following figure). These binned wafers are then performance-tested. The remaining wafers are certified according to the results of their performance level and placed in a wafer bank for later use.

Figure 361: Binning Settings



There are two parts to the binning system setup:

1. Binning Settings
2. Wafer Level Plot

Select the **Binning Settings** icon to enter data.

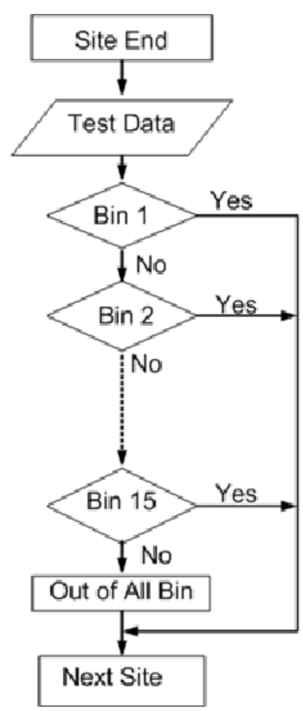
General Settings

Enable Binning Check: If selected, the Enable Binning Check and all the following functions activate. When activated, the Enable Binning Check feature runs a test on the devices, compares the results to the limits set by you, and places them in specific bins. Sites can be color-coded to distinguish between the categories (bins). In addition, a distinct color can be set for every site on the binning wafer map.

Criteria: You can select Pass or Fail. Pass means that if all test results are within the Bin range (Low < = X < = High), the tested die is placed in the Pass bin. Fail means that if the tested die does not fall within the Bin range (X < Low or X > High), the tested die is placed in the Fail bin.

Number of Bins: You can select 1 (highest priority) to 15 (lowest priority). When selecting a number, the same amount of Bin tables are generated. There are a total of 15 Bins supported. Each Bin has a priority level. When binning, the software checks if it is in Bin 1 first, if not, then Bin 2, then Bin 3 and so on up to and including Bin 15. If a tested die does not fall into any Bin category, it enters All Bin Pass or All Bin Fail, as shown in the following table.

Figure 362: Binning flow chart



Names and Colors:

- **All Bins Fail/Pass Sites (1 Character only):** Sets the name and color for all bin Fail/Pass criteria sites. For pass criteria, it shows All Bins Fail Sites. The default value is blank.
- **Invalid Sites (1 Character only):** Sets the name and color for invalid sites. The default value is blank.
- **Clear Sites (1 Character only):** Sets the name and color for unselected sites. The default value is blank.

Bin Tables:

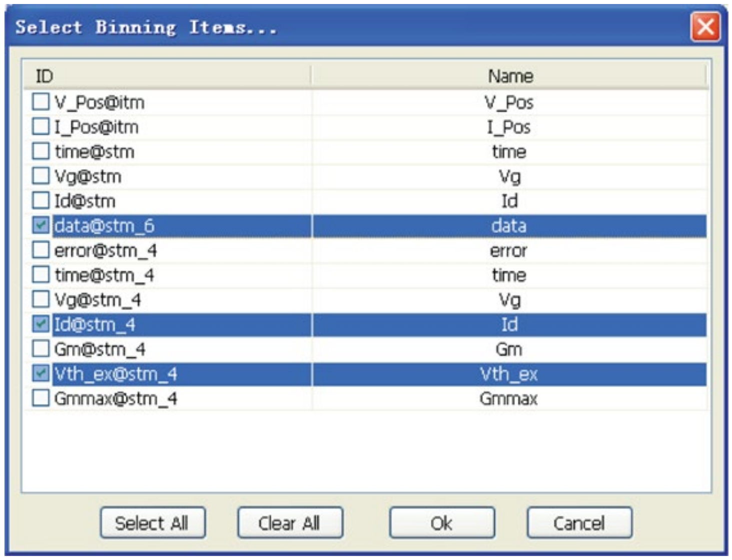
- **ID:** Parameter id, defined as `parameter_name` + `@` + `test_name`. An `@test_name` string is appended to the end of the parameter name to distinguish between parameters from different tests.
- **Name:** Parameter name.
- **Low/High:** Defines the binning range for the corresponding parameter. The binning operation uses this range.
- **Spec High/Low in limits table:** Defines the low and high values that will determine if a die passes or fails during automation. The rest is shown on the wafer map in automation. This range is also used to generate a summary report.
- **Valid High/Low in limits table:** Sets an Exit Condition for a tested die that falls out of the valid high/low range. If a test is out of range, a corresponding Exit action activates. This range is also used in the summary report.
- **High/Low in Binning table:** Defines the high and low parameters when binning is enabled. All wafer sites are assigned different bins according to the binning settings. When Binning is enabled, wafer map in Automation shows the binning result, rather than the pass/fail information with Spec High/Low.

Bin Parameters: Select **Bin Parameters** to select the parameters for the binning operation. The Select Binning Items dialog opens.

NOTE

To avoid possible errors when setting parameters, remember to consider the differences between Valid High/Low, Spec High/Low in the limits table, and High/Low in the Binning table.

Figure 363: Select Binning Items



The listed parameters in the Select Binning Items dialog originate from the test modules in Test Setup.

After checking the parameters that will be used for binning, select **OK** to activate them. The selected parameters are shown in the corresponding Bin table. The Low and High values are loaded with the default settings (the first time) or from previously saved binning settings.

- **Bin Name (1 character only):** Sets the name and color for current bin.
- **Bin Pass/Fail color:** Sets bins to distinct colors.

Bin setting procedure

If binning is enabled, the binning information shows on the wafer map during automation. In addition, an optional binning file generates that transfers the results to the wafer level binning plot to detect locations with problems.

To use the bin setting procedure:

1. Build one automation test project that runs properly.
2. From the Automation panel, select **Advanced Binning Settings**.
3. Select **Enable Binning Check**.
4. Choose the binning criteria based on the Pass and Fail requirements.
5. Choose the number of bins, from 1 to 15; Bin 1 has the highest priority, Bin 15 has the lowest. When binning, the software first checks if the results fit into Bin 1. If the results fit into Bin 1, the binning process advances to the next site to be tested. If the results from the test do not fit into Bin 1, the software sequentially checks the remaining bins for a fit. If the results do not fit into any bin, an All Bin Pass or All Bin Fail statement appears.

NOTE

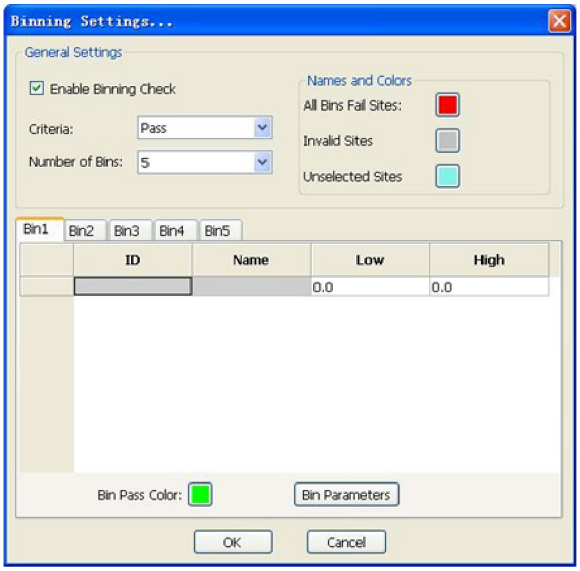
If the testing results of a site are placed into a bin, the binning process advances to the next site to be tested.

6. Select the bin parameters and choose the parameters for accessing the bin.
7. Set the binning low and high ranges. You must set each bin individually.
8. Set the binning colors. Every bin must have a color and name.
9. After bin setting is complete, select **OK** to run automation. The binning results are displayed in real time on the wafer map, and any location on the binning wafer that is invalid or not selected is detected and displayed.

Enable the Binning Check example:

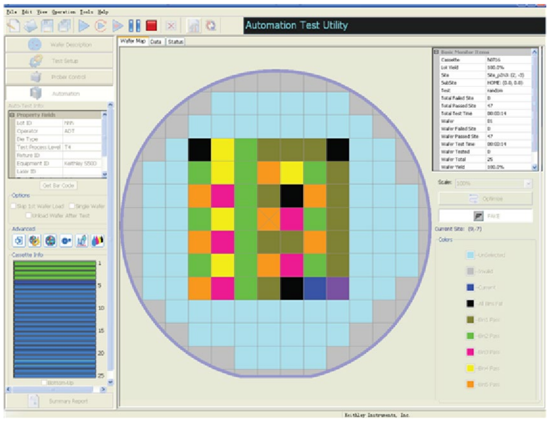
1. Build a project by selecting the STM named **random**. The *random* STM generates random data smaller than one and calculates an average value using the Formulator.
2. Select **Binning Settings** to access the binning settings panel.
3. Select **Enable Binning Check**, then select the following: Criteria, Pass, Number of Bins: 5.

Figure 364: Binning Settings example



4. Select **Bin Parameters** and then select the output *data@random* for every bin.
5. Set the limits for every bin to: bin1 [0,0.2], bin2 [0,0.4], bin3 [0,0.6], bin4 [0, 0.8], bin5 [0,0.95].
6. Set and choose a distinct color for each bin.
7. Select **OK**.
8. Select **Run**. The binning results display in real time on the wafer map.

Figure 365: Binning real-time results (example)

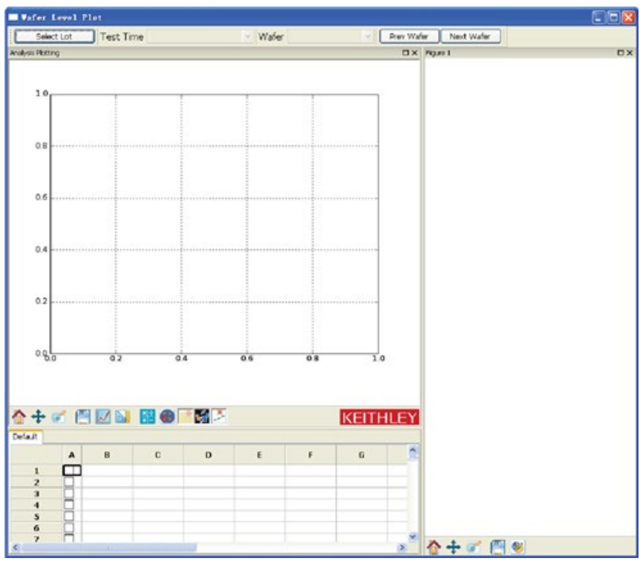


Wafer Level Plot

As a result of the tests performed during binning, the wafer plots are assigned to specific categories for later use. Binning wafer plots are important to trace root causes for higher future yields. For example, from binning characteristics, engineers can collect data and correlate them with specific processes. This saves time and aids in finding root causes of problems.

In this part, you can retrieve binning test results, trace the unwanted data, and get a site plot. Select the Wafer Level Plot icon.

Figure 366: Blank Wafer Level Plot



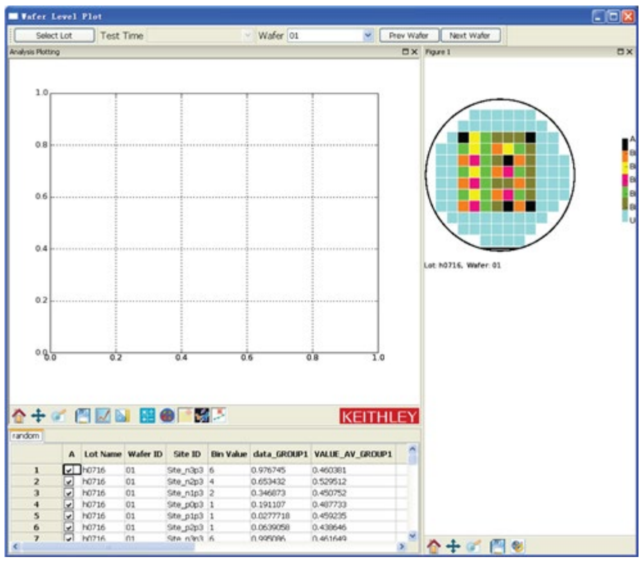
Wafer Level plot definition

The items in the Wafer Level plot dialog are as follows.

Select Lot: Open the `.kdf` file. The selected `.kdf` file should be generated from the present test plan and should match the present test plan.

After the file is loaded, the bin map of the first tested wafer is displayed.

Figure 367: Wafer Level Plot



Test time: Disabled in `.kdf` version.

Wafer: From the list, select the binning tested wafer. The Prev Wafer button moves to the wafer before the one selected in the list. The Next Wafer button moves to the next wafer.

Analysis Plotting and Analysis data sheet: This plotting supports the data plotting and site analysis. In data plotting, data is retrieved from all sites that belong to the present wafer. In site analysis plotting, data is retrieved from one selected site. Here, using the above case, “random,” is an example of one of the two types of plotting.

The items in data sheet of the wafer level plot dialog:

- Lot Name
- Wafer ID
- Site ID
- Bin value

There are a total of 15 Bins supported, and the Bins are prioritized. Bin 1 has the highest priority, and Bin 15 has the lowest. When binning, the software checks if it is in Bin 1 first, if not then Bin 2, then Bin 3... then Bin 15. If not in any Bin, it enters All Bin Pass or All Bin Fail (see the following table).

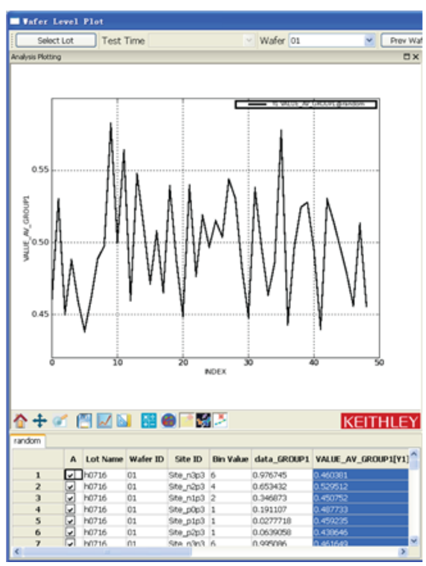
Bin Value	Comment
1,2,3,…15	Bin 1, Bin2,…Bin 15
0	All bin pass (in fail criteria) or all bin fail (in pass criteria)
<0	Abnormal result

NOTE

The erased site has no value in `.kdf` file, so it does not show here. In the binning file (`.nbf` file), the value of the erased site is 255. It is a virtual number.

For data plotting, set the average value of *random* data `VALUE_AV` as axis Y1, then select the plot function icon. Select **Plot1**, and the average value of the *random* data from the wafer 1 test site is plotted (see the following figure).

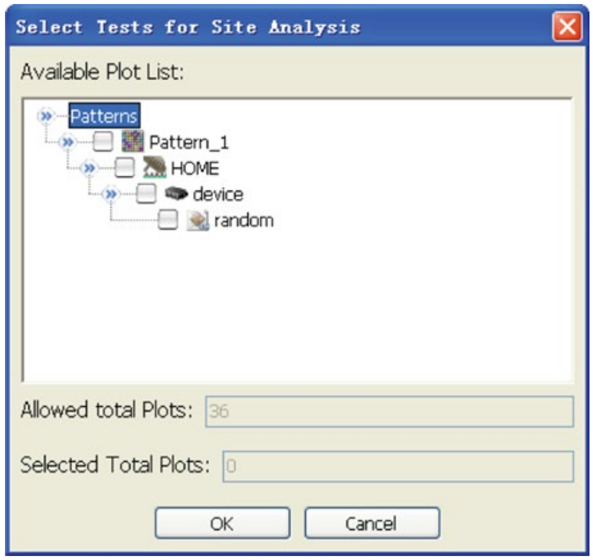
Figure 368: Data plotting example



Site analysis plotting:

Click **Select Test** to execute Site Analysis for the test module you want. A dialog opens, as shown in the following figure.

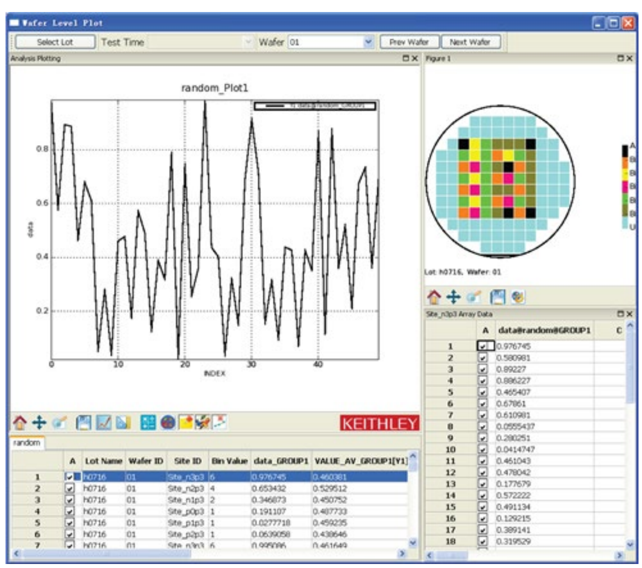
Figure 369: Site Analysis plotting



Select the test module you want to perform the site analysis on, then select **OK**.

Select the site in the bin map to show the plotting. The data of this plotting is shown to the right, and the selected site in the Analysis data sheet highlights in the data sheet at the bottom (see the following figure).

Figure 370: Site Analysis plotting example



If you do not see the following figure plot, verify whether a plot has been defined in the Data Sheet of Test Setup. After all the above steps have been taken, you can trace data and do site analysis.

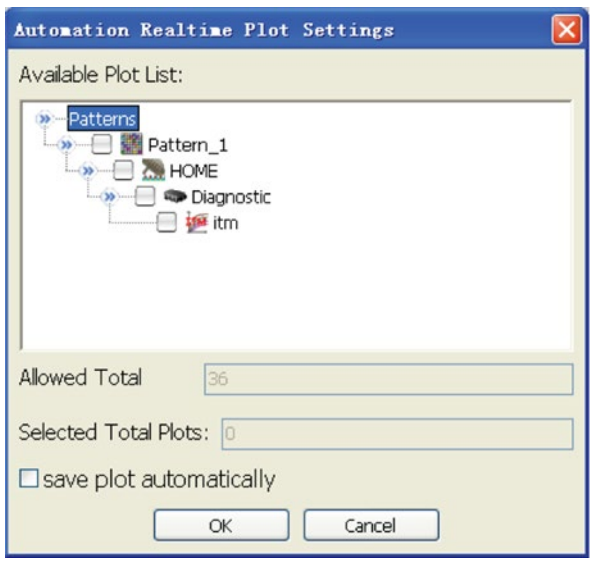
Real-time plot settings

The real-time plot settings define plots in the data sheet.

To define the plots:

1. Select Realtime Plot Settings. The dialog shown in the following figure opens. The configuration navigator shows the tests included in the automation.

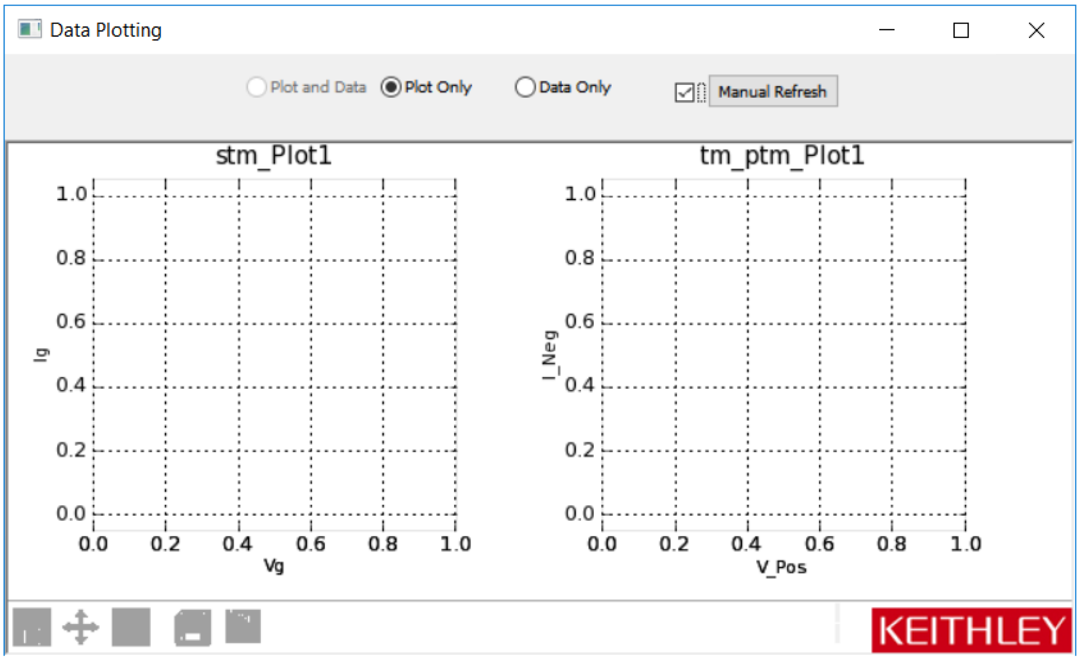
Figure 371: Automation Realtime Plot Settings



2. Select the tests whose plots need to be displayed in real time.
   - **Allowed Total**: Indicates that the maximum number of plots is 36.
   - **Selected Total Plots**: Shows the number of plots that have been selected.
3. If save plot automatically is selected at the end of each site, the Realtime Plot Settings dialog automatically saves the plotting curve to the kdf folder of the present project. Select **OK** to finish this step.

Select **Show Real Time Plot** and a dialog opens. You can now choose the view style and do basic operations. Note that you can manually refresh the data plotting graph for the STM and PTM by selecting manual refresh during testing.

Figure 372: Real-time Plot



Post-inking

This feature is used for automation wafer inking after automation test. Only EG probers are supported for posttest inking.

The post-inking feature performs automation and inking separately. Before accessing the post-inking function, you must run the automation test and generate a data file and binning file.

To set up post-inking:

1. Select the post-inking settings icon. The following dialog is displayed.

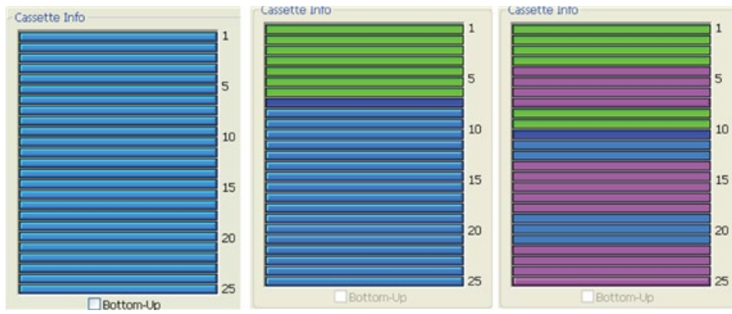
Figure 373: Post Inking Settings

2. Under **Binning File**, select **Browse**.
3. Select the binning file.
4. Set the Delay Time.
5. Select Ink Bad (Erased) Die to set the erased die inking number.
6. Choose Manual or Auto Load Wafer.
7. Under Wafers in binning file, review the binning wafer IDs.
8. Use Run, Pause, Stop, and Exit to control the inking process.

Cassette information box

Cassette information shows the runtime information of the wafers/slots in a cassette. There are four types. Move your mouse over the wafer to display the wafer status.

Figure 374: Cassette Info box



- Selected Wafer (dark blue)
- Unselected wafer or nonexistent wafer (purple)
- Current Test Wafer (light blue)
- Tested Wafer (green)

Preference settings

The Preferences setting is accessed from the Tools menu option. Preferences are used to apply customized settings before manual or automation testing. You have the option to make changes to modify the path for saving files, instrument, prober, and automation settings. Plus, you can modify the `.kdf` file and there are many items in the Advanced tab to configure your testing.

There are six tabs in Preferences:

1. Path
2. Instruments
3. Prober
4. Automation Settings
5. KDF File
6. Advanced

Preference settings path tab

The following is a brief description of the areas in the Path tab.

Project Repository: You can also select browse to select a different path for projects.

Auto-Load Project: When launching ACS, this is the path that automatically loads the default project or the last loaded project. The project loaded depends on if you have previously loaded a project.

Default Project: This field and the associated browse button are only accessible when the Default Project option is selected in the Auto-Load Project list.

Summary Reports Path: You can also select browse to select a different path for projects.

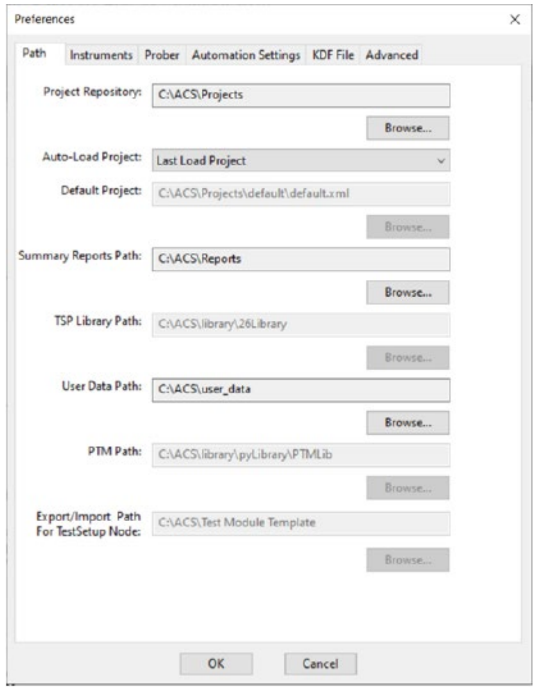
TSP Library Path: You can select browse to select a different path for projects.

User Data Path: Used to set the binning file and data file path. The user data path is saved as a software parameter. This path can be accessed in a UAP, which means you can save the data file in this path. It may also be saved to the Convert Data dialog as the default output data file path. You can also select browse select a different path for projects.

PTM Path: You can select browse to select a different path for projects.

Export/Import Path For TestSetup Node: You can select browse to select a different path for projects.

Figure 375: Preferences Path tab



Instruments tab settings

NOTE

If any of the matrices in the system are set to Remote Sense, all the Series 2600B and Series 2400 instruments in the system will be set to four-wire (4-W) mode when executing a test. Also, the SMU Remote Sense option box in the Preferences settings Instruments tab will be disabled. When all the matrices are set to Local Sense, all the Series 2600B and Series 2400 instruments in the system will be set to two-wire (2-W) mode when executing a test. It is highly recommended that all the matrices in the system are in the same mode. For more information on how to configure a matrix, refer to ACS hardware configurations (on page 10-1).

The following is a brief description of the areas in the Instruments tab.

Check all: Selects all the SMU options (these are cleared by default). Or you can manually select the SMUs (1, 2, or 3) indicated in the tab.

Source Auto Delay for 26xx: Select to enable the source auto delay function for the Series 2400 SMU in an ITM.

Measure Auto Delay for 26xx: Select to enable the measure auto delay function of the Model 2635A or 2636A in an ITM or RTM.

Automatic Contact Check for 26xx: Select to enable automatic contact check for the Series 2600B SMU in an ITM before testing.

Initiate Unused SMU: Only used for RTM and ITM SMUs. There are three states for the initiate unused SMU.

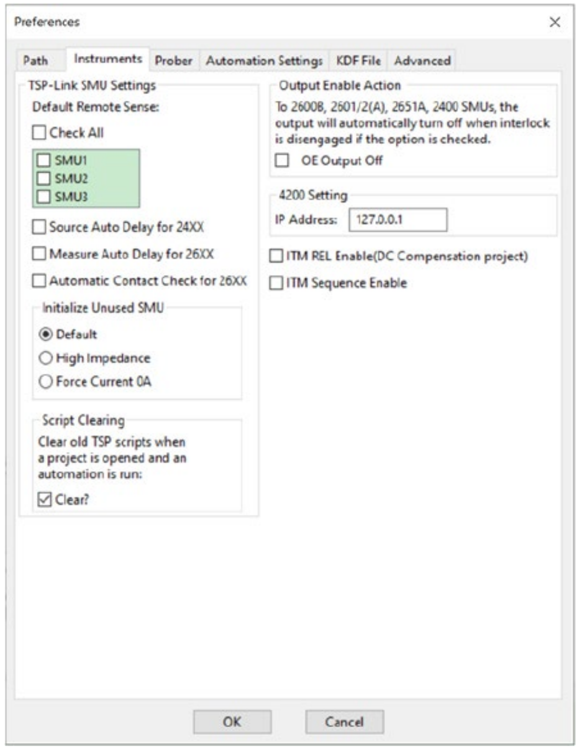
- Default: OUTPUT_NORMAL OFF mode. Selected by default and it outputs 0 V when the output is turned off.
- High Impedance: HIGH_Z OFF mode. Zero the output (in either volts or current) when off.
- Force Current 0A: 0 A in ON mode. Forces the output current to 0 amps and turns on the output.

Script Clearing: Used to clear old scripts in the Series 2600B SourceMeter instruments when a project is opened and running automation.

Output Enable Actions:

- 4200A Setting IP address: Model 4200A-SCS IP address (when S4200A remote control is required).
- ITM REL Enable (DC compensation project): Select to show the Enable REL Function setting in the ITM. Refer to DC compensation for more detail.
- ITM Sequence Enable: Select to enable the On Sequence for "xxx" (pad name) and the sequence delay settings in the ITM. For more detail refer to Model 26xxB and 24xx SMU parameters (on page 5-61).

Figure 376: Instruments tab



Prober tab settings

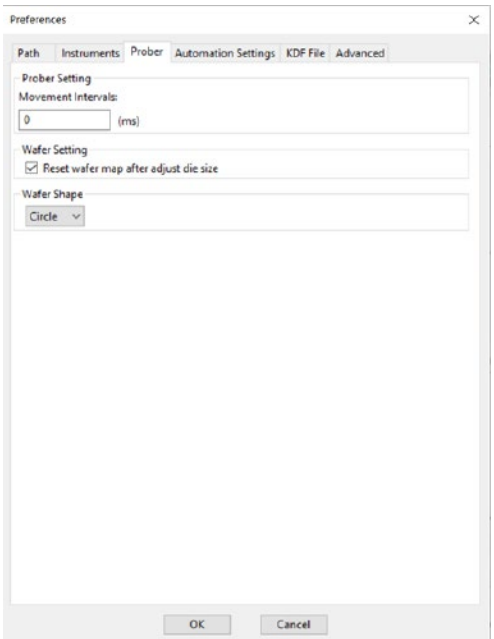
The following is a brief description of the areas in the Prober tab.

Movement intervals: The interval time between each prober movement during an automation test. The default value is 0 µs.

Wafer Setting: This is selected by default. This setting resets the wafer map if the size of the die is changed.

Wafer Shape: There are two options: circle or square. The default is circle.

Figure 377: Prober Tab settings



Automation Settings tab

The following is a brief description of the areas in the Automation Settings tab.

User mode: Allows you to select either Engineer or Operator.

NOTE

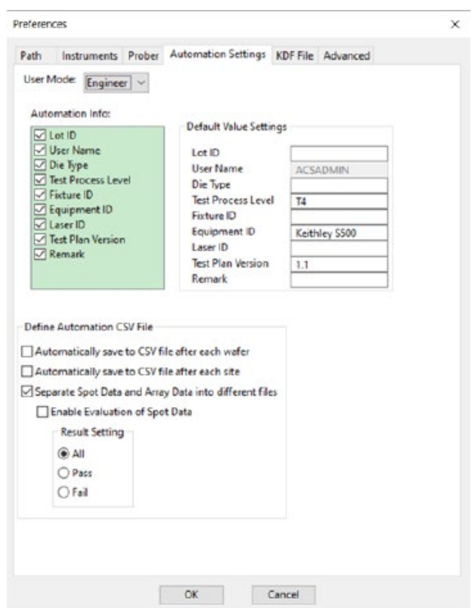
The Automation Information items can only be modified in engineer mode.

Automation Information: You can select or clear the automation information you want for your test in this section. If you do not select any items, the default value settings will be applied to your test.

Define Automation CSV File:

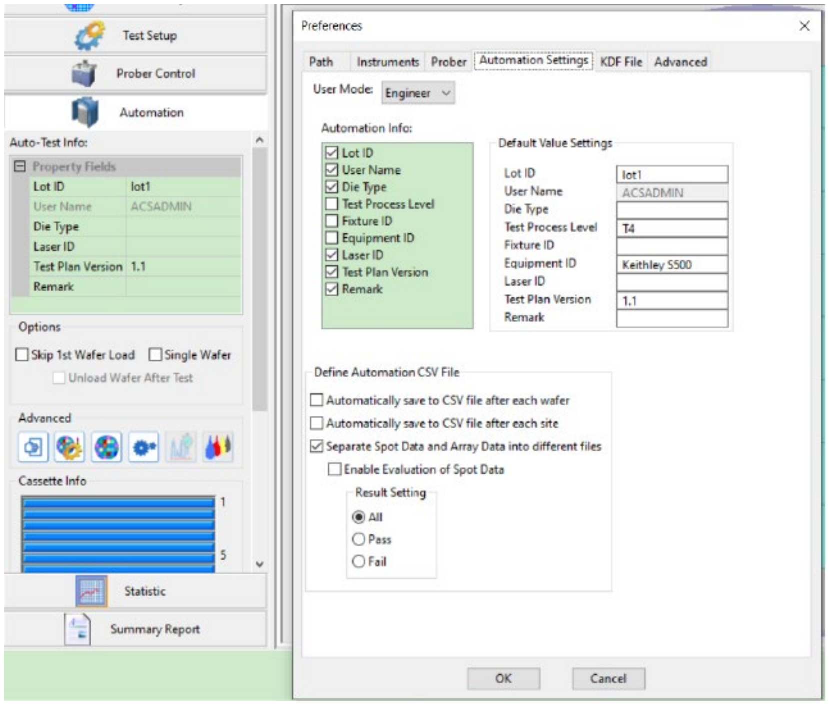
- **Automatically save to CSV file after each wafer**: Select this option to automatically save test data of each wafer to a `.csv` file after testing.
- **Automatically save to CSV file after each site**: Select this option to automatically save test data of each site to a `.csv` file after testing.
- **Separate Spot Data and Array Data into different files**: This is selected by default. It saves spot data and array data to each file.
- **Enable Evaluation of Spot Data**: Select this option so the evaluation results are shown in the data file for the spot data.
- **Result Setting**: You can choose any combination of All, Pass, and Fail so the spot data will be shown in data file.

Figure 378: ACS Automation settings tab



An example of the automation information that is selected is in the following figure.

Figure 379: Automation tab information



KDF File tab settings

The following is a brief description of the areas in the KDF File tab.

Save Test Data to KDF file: This is selected by default. This saves the automation test results in a `.kdf` file.

Do Not Remove Old KDF Folder When Overwrite: Select this option to prevent the `.kdf` file from getting overwritten with a new project.

Generate Test Data File (.KDF file) Each Wafer: Select this option to generate a `.kdf` file for each wafer.

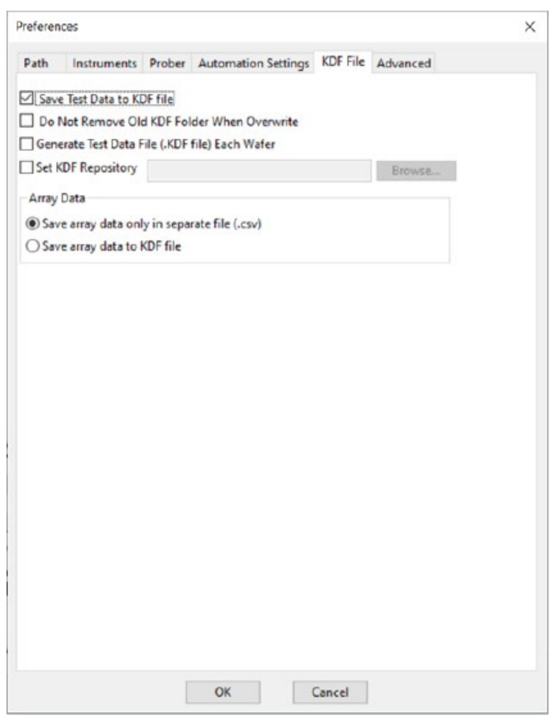
Set KDF Repository: Select this option to choose a directory to save your `.kdf` files.

Array Data:

Save array data only in separate file (.csv): This is selected by default. This saves the two-dimensional array data of the automation test to a `.csv` file.

Save array data to KDF file: Select this option to save the two-dimensional array data of the automation test results to a `.kdf` file.

Figure 380: KDF File tab settings



Advanced tab settings

The following is a brief description of the options in the Advanced tab settings.

Parallel Execute Group In: You must choose one of three choices: Test, Device, or Subsite.

Working Mode: Select local mode or network mode. Local mode is the default. The network option is for a server.

Log Window Display Level: Determines which level of messages are displayed in the log window, INFO, WARNING, or ERROR.

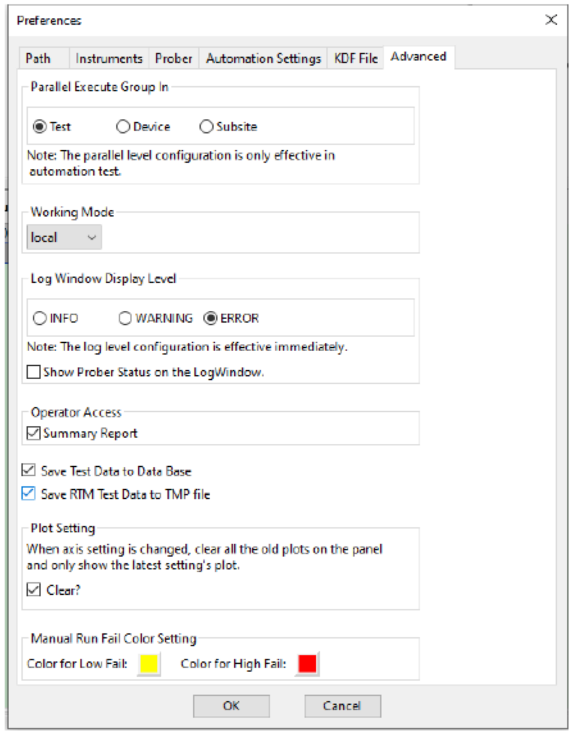
Show Prober Status on the LogWindow: This option displays the prober control information.

Operator Access: Select this option to get a summary report of the operator information testing. If this option is clear, there is no report.

Plot Setting: Select **Clear?** to clear all plots on the panel and only show the newest plot.

Manual Run Fail Color Setting: If a test result is out of specifications for Spec Low or the Spec High range, as defined in the Pattern-Limits page, the background color of the data cell changes to the Manual Run Fail Color Setting. For more information, refer to Fail data (on page 5-249).

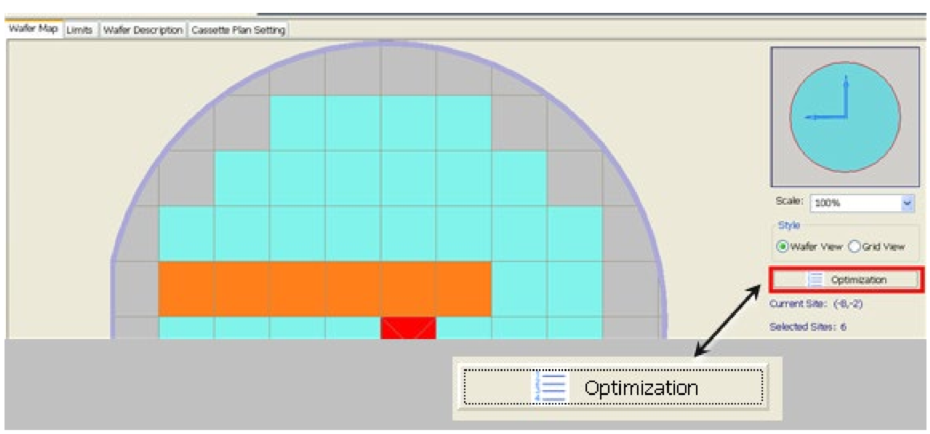
Figure 381: Advanced tab settings



NOTE

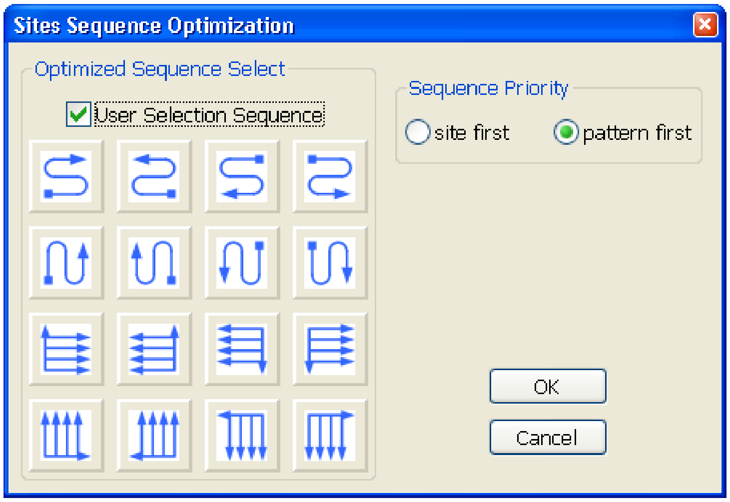
If you are using the UAPs in a module and you want to print wafer ID and pattern ID information in ACS, you must choose INFO in the Log Window Display Level. You must also select Optimization on the Wafer Map tab, shown in the following figure.

Figure 382: Optimization function on Wafer Map tab



Once you select Optimization, the Site Sequence dialog opens. This is where you select pattern first in the Sequence Priority to ensure that your pattern ID is displayed in the log window when you print from the UAPs of your module (see the following figure). The default Sequence Priority is set to site first.

Figure 383: Pattern first in Sites Sequence Optimization function



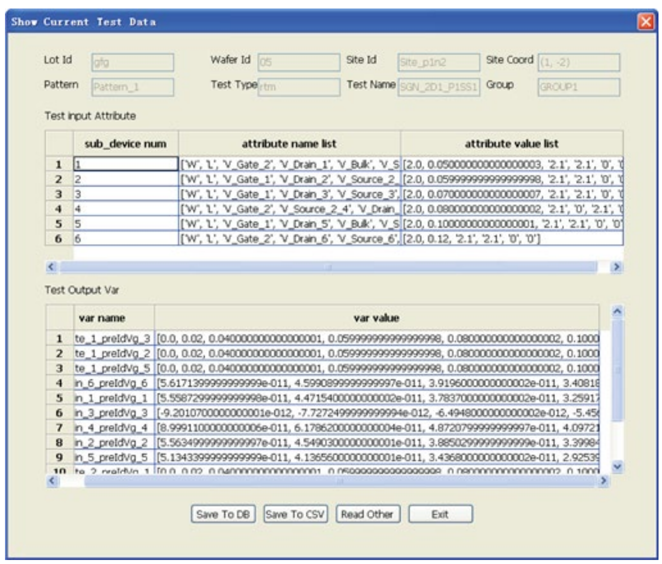
Read current test data

To read current test data:

1. From Tools, select **Read Current Test Data**.
2. Select **Browse** to open the test data `.tmp` file (in the directory `C:\ACS\Projects\project_name\RTM_temp`). When the target file is found, select **Read** to display the Show Current Temporary Data dialog.
3. You can save your data using the following options:
   - **Save to DB:** Saves the data to a `.tmp` file in the database.
   - **Save to CSV:** Save the data to a `.tmp` file in the database.

You can use Read Other to find other files to open.

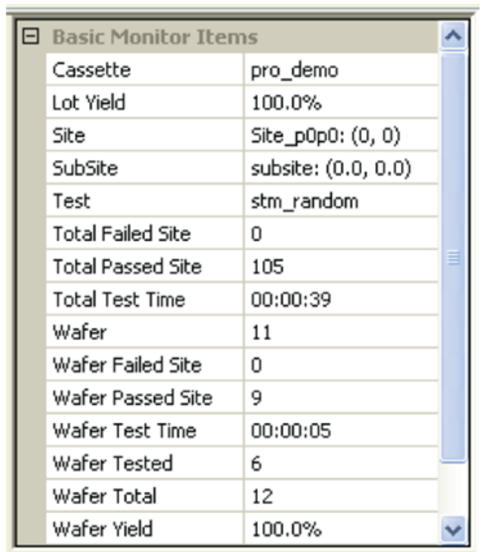
Figure 384: Show Current Temporary Data



Wafer map indicator

During the Automation Test, the information in the Basic Monitor Items table will be updated automatically to show the current running information.

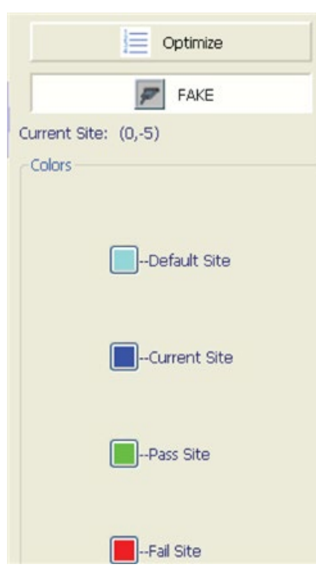
Figure 385: Basic Monitor Items table



Basic Monitor Items	
Cassette	pro_demo
Lot Yield	100.0%
Site	Site_p0p0: (0, 0)
SubSite	subsite: (0.0, 0.0)
Test	stm_random
Total Failed Site	0
Total Passed Site	105
Total Test Time	00:00:39
Wafer	11
Wafer Failed Site	0
Wafer Passed Site	9
Wafer Test Time	00:00:05
Wafer Tested	6
Wafer Total	12
Wafer Yield	100.0%

The Wafer Map Indicator is used to optimize the sites list, set the Offline/Online mode of the prober Model, change, and show the color settings, and to show the coordinates of the current site. These functions are discussed below:

Figure 386: Wafer map indicator



Optimization – Optimizes probing path of selected sites to minimize the prober moving time. You can choose from 16 different sequences. When a sequence is selected, the tester starts at the site closest to the chosen location and then proceeds to all selected sites on the wafer, following the selected serpentine sequence.

This function is used to set the Offline/Online mode of the prober Model configured using the Configure Hardware command from the Tools menu. You can toggle the function to switch between the two different modes. When the configured prober Model is Offline, the function is displayed as FAKE. When Online, the function is the Model name, for instance TSK9.

Current Site (x, y) – This information is used to indicate the coordinates of current site pointed to by the mouse/cursor.

Current Site – This function is used to show and set the color of the current site on which the prober is probing. You can change the color setting of the current site by selecting this function.

Pass Site – This function is used to show and set the color of those sites that have already passed testing. In the passed state, all test modules in the site have qualified output values validated by the pre-defined limits. You can change the color setting of a passed site by selecting this function. Fail Site – This function is used to show and set the color of those sites that fail to pass testing. In the failed state, all test modules in the site do not have qualified output values validated by the predefined limits. You can change the color setting of failed sites by selecting this function.

Start an automation test

To run an automation test:

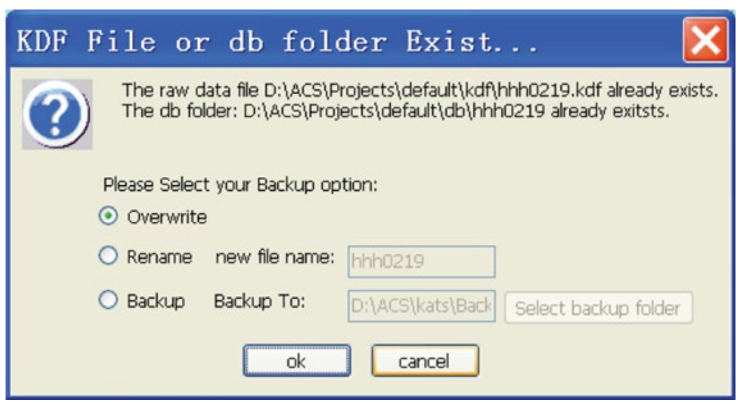
1. Select **Run** on the toolbar to start an automation test. The automation test is launched.
2. Select **Pause** on the toolbar to pause the automation test.
3. Select **Stop** on the toolbar to stop the automation test.

You can view a Summary Report of the test execution using the ACS Summary Report functions. These functions are discussed in Summary report panel (on page 9-1) later in this section.

NOTE

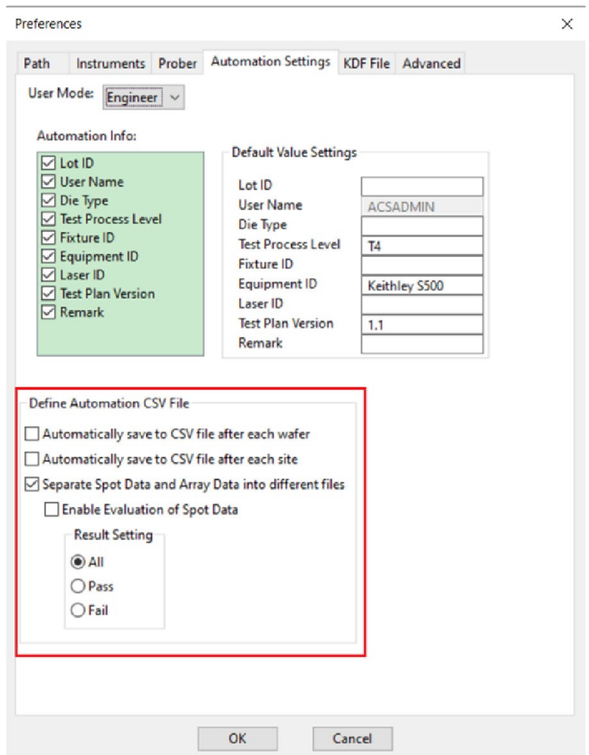
The data file or data file folder name is derived from the Lot ID that is assigned by you before the automation test. Therefore, if the data file already exists, the KDF File or db file Exist dialog opens and you can choose to back up the file (see the following figure).

Figure 387: KDF File or db folder Exists



The results of an automation test are saved as a `.csv` file, `.kdf` file (Keithley Data File) or a database file (db folder file) automatically (refer to the KDF File tab and the Advanced tab in the following figure). You can choose how to save your data in the **Tools** > **Preferences** dialog, as shown in the following figure.

Figure 388: Automation Settings tab



A Keithley data file (`.kdf`) contains the following items:

typ	type and version
lot	lot ID
prc	process ID
dev	device ID
tst	test program name
sys	tester name
tsn	tester number
opr	user name

Data part contains:

wafer_id	split
boat	slot
site_id	coordinate_x
coordinate_y	bin_value
data_id	test_value

NOTE

The data_id part can contain from three to four parts, which are separated by the following symbols: ¡°@¡± sign. For example, output_name@test_name@sub_site[@GROUP_ID], and the group id (the fourth part) is optional.

If more than one group is in the test, the group_id will be added into the data_id.

Automation process example

The next examples demonstrate binning and post-inking, stress migration, and how to create a new project and save the `.csv` file.

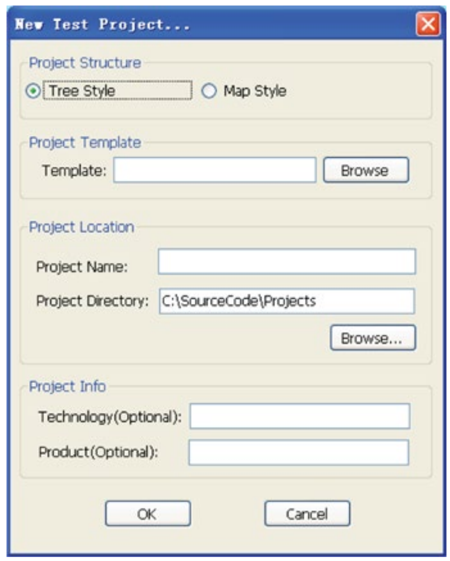
Example 1: Binning and post-inking

This example shows a typical operation flow of automation with real-time binning and post-inking.

To set up and run the binning and post-inking example:

1. Select **File** > **New**, then select **New Project**.

Figure 389: New Test Project

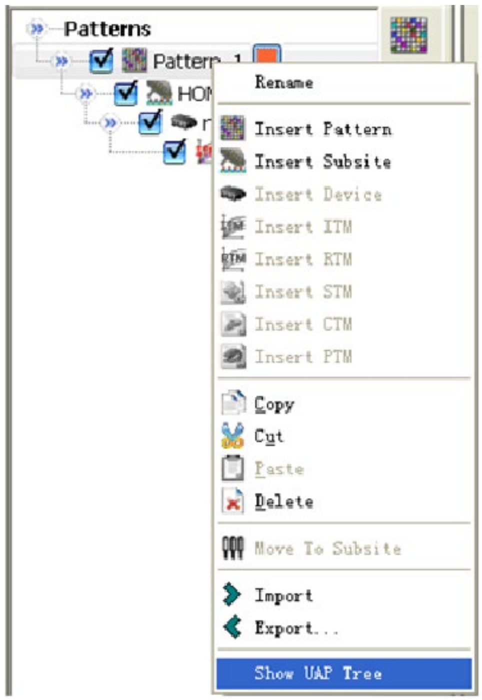


2. Input the name of the project and appropriate information, then select **OK**.
3. Select **Wafer Description**. Input the wafer parameter information as the wafer under test.

Figure 390: Wafer Description panel

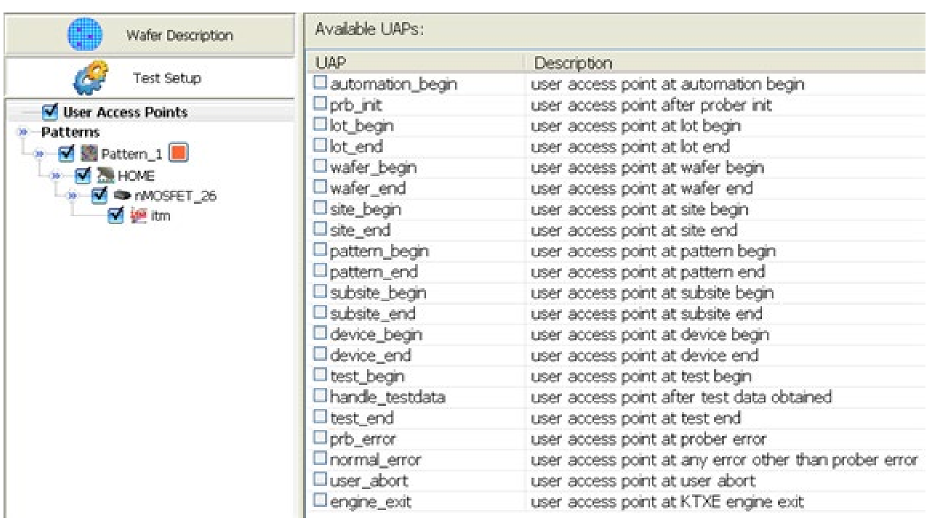
4. Select **Test Setup**. You can now set up your own test plan. There are several test modules available: ITM, STM, CTM, and PTM. Select the module that you want to run.
5. Right-click the module on the configuration navigator and select Show UAP Tree.

Figure 391: Access UAP tree



6. The UAP dialog opens. Select the **lot_end** UAP.

Figure 392: Available UAPs



7. Add two Python modules to the configuration navigator.

Figure 393: UAP and test tree management

8. Import binning modules from the ACS UAP library: `NBF.py` and `NDF.py`. Also, if needed, you can write your own binning customized PTM module, then select **Save**.

NOTE

You should use the NBF to generate binning files for the wafers in a cassette. It must be put at LOT end UAP. Also, you should use the `.ndf` to generate data files for wafers in the cassette. It must be put at LOT end UAP.

An `.nbf` file contains the following items.

Wafer header part contains:

```
<WAFER>
WAFER ID: wafer01
TEST SESSION NAME: operator
LOT ID: lot01
EQUIPMENT ID:    Keithley S500
FIXTURE ID:
TESTPLAN NAME:   demo
TESTPLAN VERSION:      1.1
TEST PROCESS LEVEL:    T4
FLAT REGION:     180
DIE SIZE X (mm): 5.65
DIE SIZE Y (mm): 0.75
COORDINATE QUADRANT:    1
REFERENCE DIE X: 4
REFERENCE DIE Y: 15
TEST OPERATOR:   TL
TEST START TIME: 07-May-2009 14:36
TEST END TIME:   07-May-2009 14:36
```

Bin value part contains:

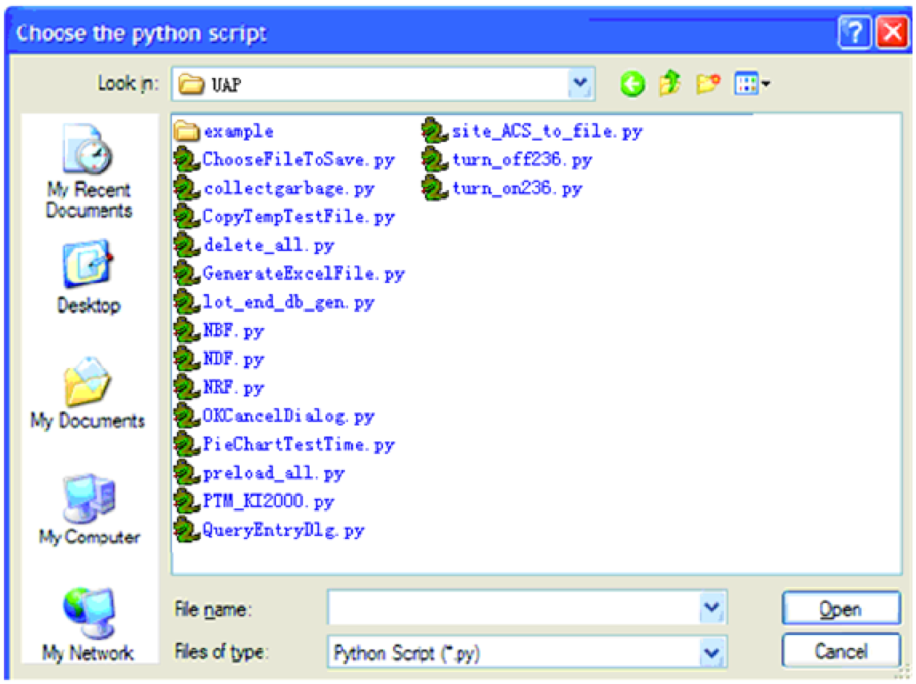
```
site_coordinate_x, site_coordinate_y, bin_value
...
</WAFER>
...
```

An `.ndf` file contains the following items:

```
Header part contains:
<HEADER>
TEST SESSION NAME:
LOT ID:   lot1
WAFER ID: wafer01
EQUIPMENT ID:    Keithley S500
FIXTURE ID:
TESTPLAN NAME:   36mm FET Jun
TESTPLAN VERSION:      1.1
TEST PROCESS LEVEL:    T4
FLAT REGION:     180
DIE SIZE X (mm): 5.65
DIE SIZE Y (mm): 0.75
COORDINATE QUADRANT:    1
REFERENCE DIE X: 4
REFERENCE DIE Y: 15
TEST OPERATOR:   TL
TEST START TIME: 07-May-2009 14:36
TEST END TIME:   07-May-2009 14:36
</HEADER>
Data part contains:
<DATA>
<WAFER>
wafer_id
   <SITE>
   site_coordinate_x, site_coordinate_y
```

```
    output_name,test_value
    ...
    bin, bin_value
    </SITE>
    ...
</WAFER>
 ...
</DATA>
```

Figure 394: Importing a PTM file for UAP

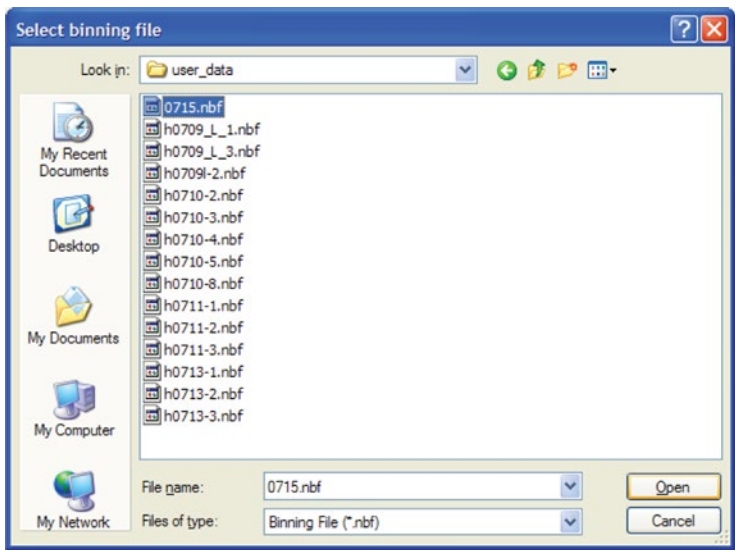


9. After you have built the test setup, navigate to the Automation panel. Input the appropriate operation information, such as Lot ID, User Name, Die Type, and Remark. All the operation information prints in the resulting data file (`.kdf`, `.ndf`).

Figure 395: Automation setup panel

10. Select **Save**, then select **Run** to initiate the Automation process. ACS automatically loads all the wafers and runs the tests. All test data and the binning results are saved in the `.kdf` file.
11. Because lot_end UAP is enabled, after the test ends, the two PTM modules execute. The `.nbf` and `.ndf` files are written according to the data in the `.kdf` file. This takes some time, depending on your computer speed. Wait until ACS finishes this procedure, which is signified by the Run function turning from a color to gray.
12. The `.nbf` and `.ndf` files are ready for post-inking. Navigate to the post-inking dialog and browse for the `.nbf` file.
13. In the Manual Load Wafer mode, a dialog displays to inform you to load the wafer with a specified wafer ID. Follow the dialog prompts, loading the correct wafer.

Figure 396: Browsing for the .nbf file



Example 2: Stress migration

In the stress migration (SM) test, another challenge is that you would like to see the data in real time during automation, and determine whether to stop the test automatically if several consecutive sites failed on one parameter, for instance, due to a bad probe pin). So the automation logic needs to be changed as well. In ACS, you can change the process flow (or operation flow) to see the data in real time.

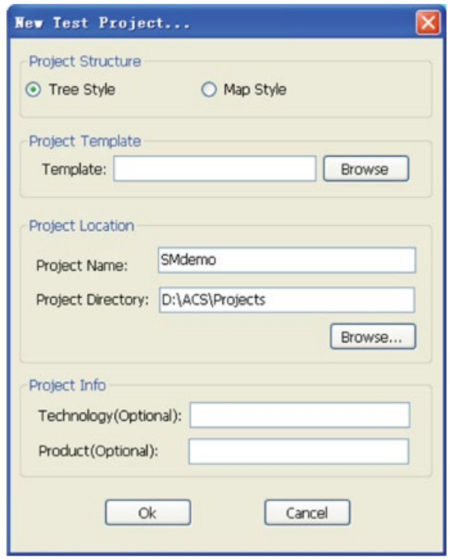
To start operation flow:

1. Create a new project.
2. Import the SM comma separated value (`.csv`) setting file.
3. Confirm the `.csv` data.
4. Set the wafer map and select the site to be tested in the wafer map. Save the project.
5. Run automation.
6. Temporarily halt the test when x number of consecutive die fails.
7. Check the data in wafer map data tab.
8. Check the Pass/Fail in wafer map.
9. The first four steps are detailed in the previous section Importing a custom setting project file.

To create a new project for SM:

For each `.csv` file, it is better to create a new project. From the ACS menu, choose **File->New**, and ACS will show the dialog below. Enter the Project Name to make a new project in ACS.

Figure 397: Create a new project for SM



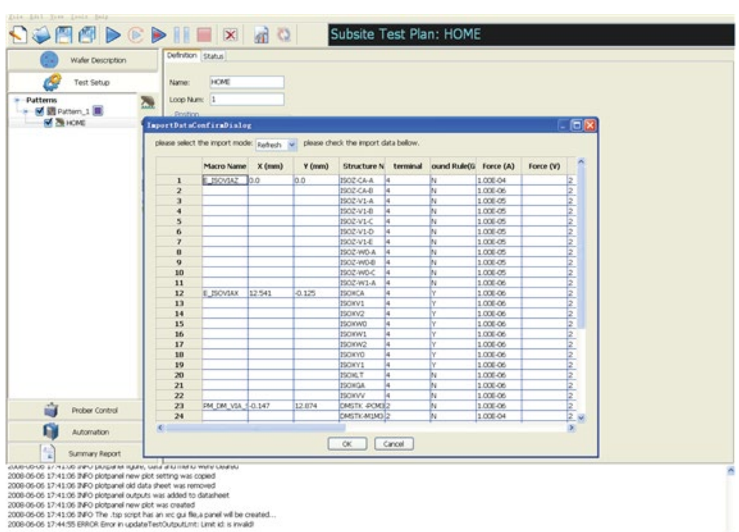
To import the SM .csv setting file:

From the ACS **File** menu, select **Import > Test Plan File(*.csv)**.

To confirm the .csv data:

After the `.csv` file is selected, ACS displays the ImportDataConfirm dialog. Confirm the data file is the correct test file, then select **OK** to move to the next step.

Figure 398: Import Data Confirmed

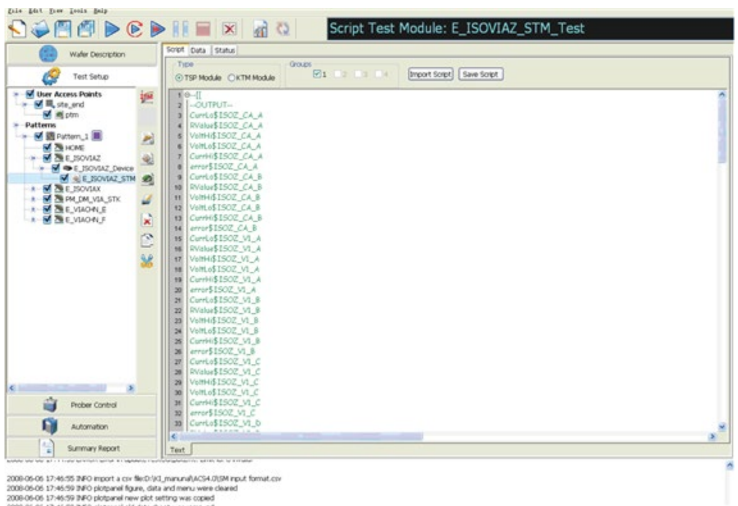


The import modes are:

- Refresh - replace existing one if they are same in test configuration navigator.
- Append - append in test configuration navigator.

After you import the **.csv** data file, the test configuration navigator, including the SM script test modules, is created automatically.

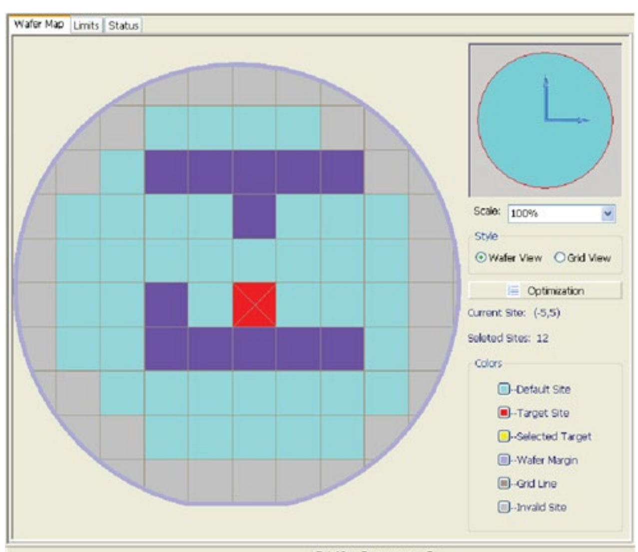
Figure 399: Test setup tree including SM script test modules



To set up the wafer map:

Next, you will be ready to set up the wafer map and the site to be tested.

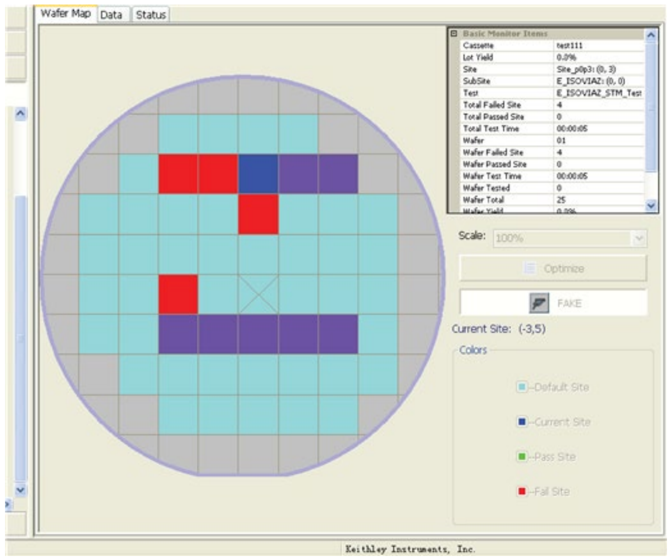
Figure 400: Setup wafer map and site to be tested



To run automation:

After you have set up the wafer map, you can run automation.

Figure 401: Run Automation

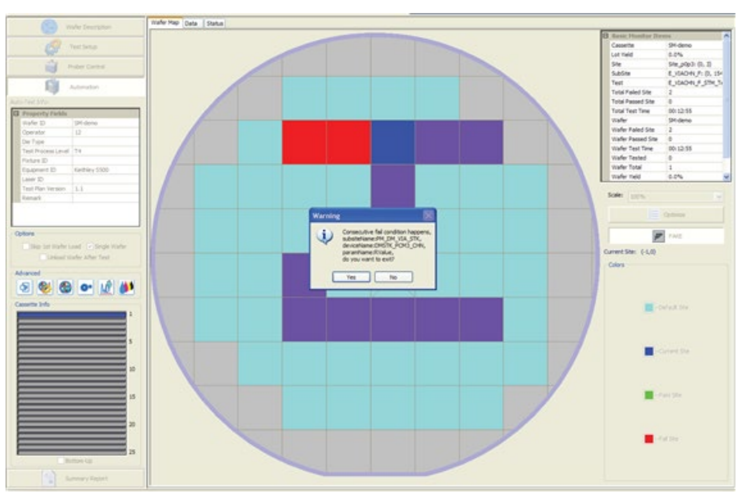


To temporarily halt the test when x number of consecutive die fails:

An x number of consecutive die is read from the `.csv` file. If that x number is greater than 0, ACS will check the rule automatically.

When consecutive dies fail and reach the setting number x, you will see a dialog. Select **Yes** to exit the test, select **No** to continue.

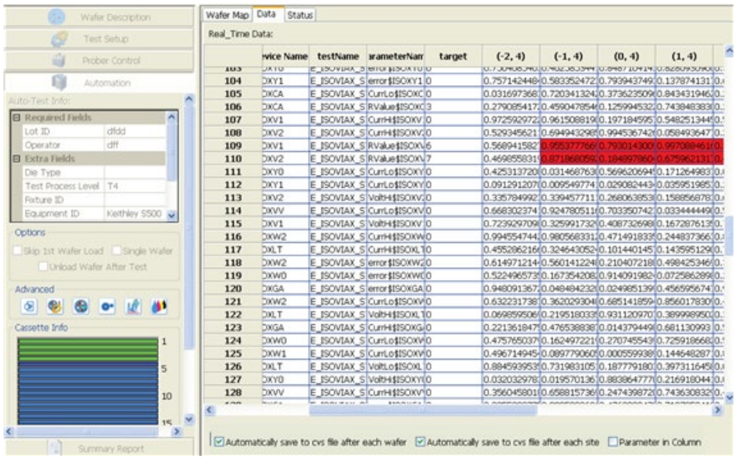
Figure 402: Run Automation complete



To check testing data in the wafer map Data tab:

The displayed testing data can be updated when switching from wafer map to data tab. You can select the Refresh function to update data. This tab can display one wafer test data.

Figure 403: Check testing data in Wafer Map Data tab

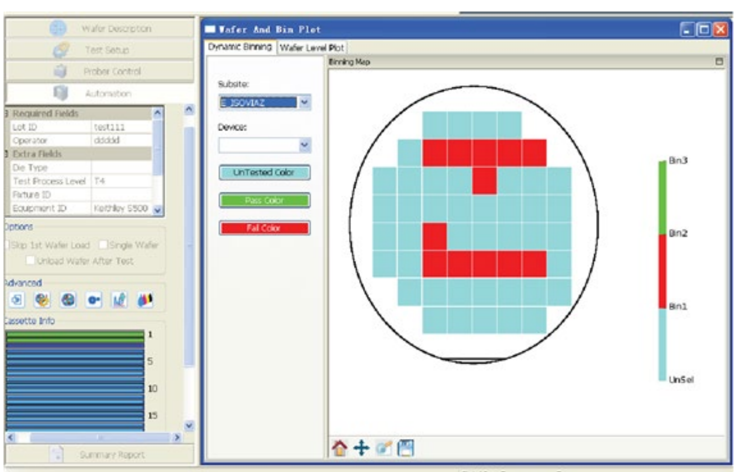


You can select to save the whole wafer data to a `.csv` file when that wafer test is complete in the Preference Settings. You also can select **Save To .csv**.

To check pass/fail in the wafer and bin plot panel:

You can check pass/fail based on Subsite or Device selection when testing is in process. When you select Subsite or Device, the wafer map will be updated, however, it cannot be updated dynamically without selection.

Figure 404: Check the Pass or Fail in the wafer bin plot

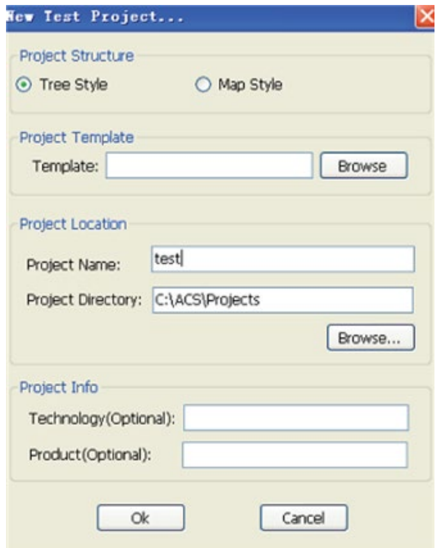


Example 3: Save a .csv file

To create a new project:

1. From the ACS menu, select **File > New**. The ACS software displays a dialog where you must enter a Project Name to make a new project.

Figure 405: Create a new project

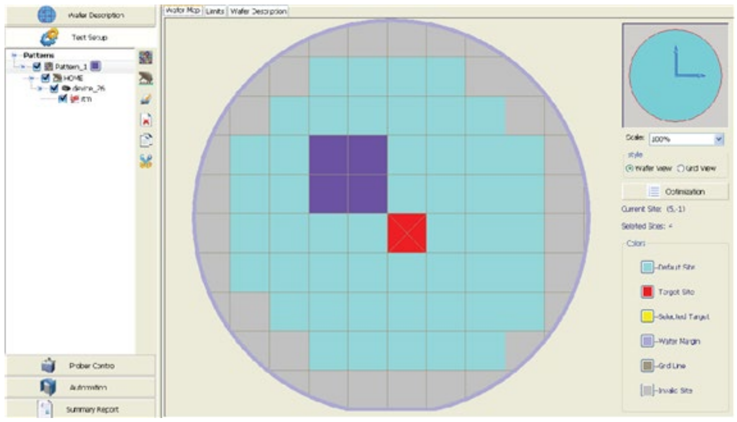


2. Add a new device and test module.
3. Add a new device that is a Series 2600B SourceMeter instrument.
4. Add a new ITM test module and configure its settings.

To select test sites:

Select **Patterns** and select some sites in the Wafer Map.

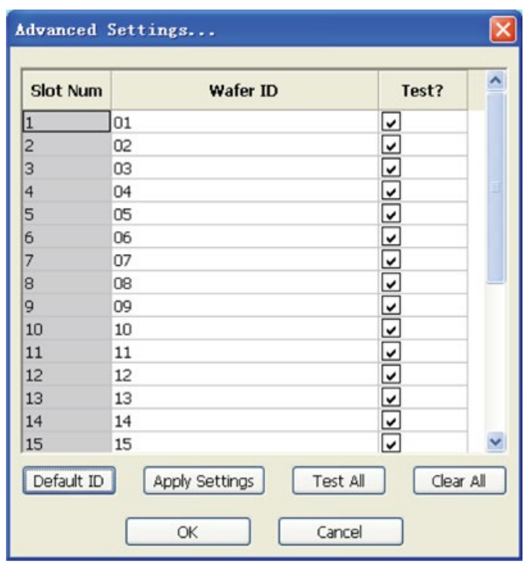
Figure 406: Selected test sites



To select test wafers:

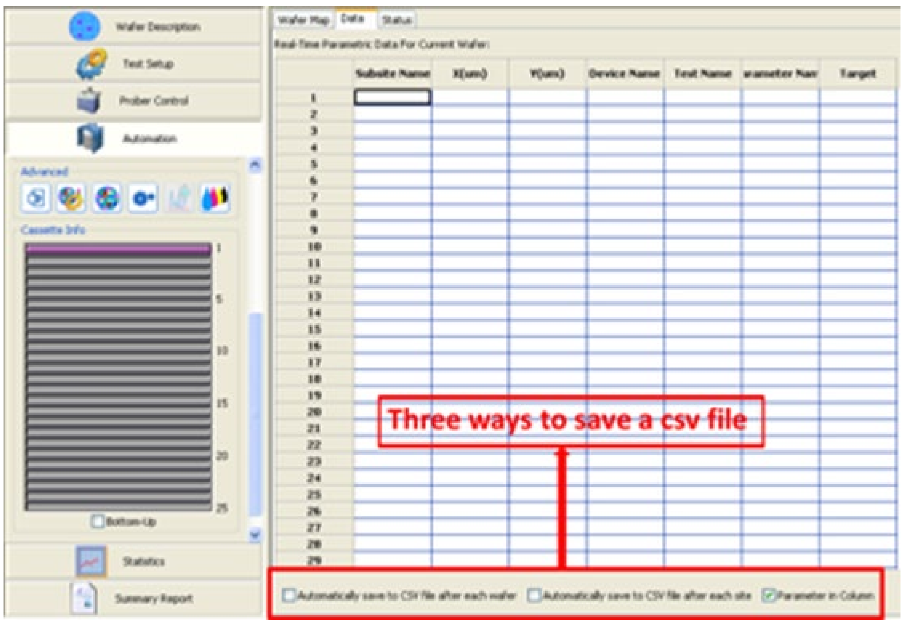
1. Select **Automation** and open **Advanced Settings**.
2. Select **Wafer 01** and **Wafer 02**.
3. Select **OK** and save this project.

Figure 407: Selected Test Wafer



4. Set the Lot ID and Operator.
5. Select the method for saving the `.csv` file. You can save the `.csv` file using any of the following methods (see the following figure):
   - Automatically save to `.csv` file after each wafer
   - Automatically save `.csv` file after each site
   - Parameter in Column

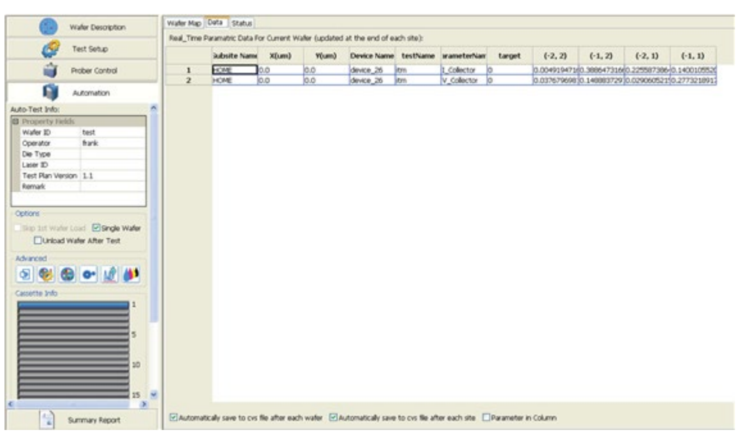
Figure 408: Three ways to save a .csv file



To run the project:

Run this project. You can see that the displayed testing data can be updated when switching from the wafer map to the data tab.

Figure 409: Testing data in the Wafer Map Data tab



To confirm test results are saved in a .csv file:

1. After the run is finished, in the …`\ACS\Projects\test\kdf`, there is a folder with a name like `xxx_WaferInfo` (*xxx* is the name of the Wafer ID). Open this folder. There are two folders and two `.csv` files.

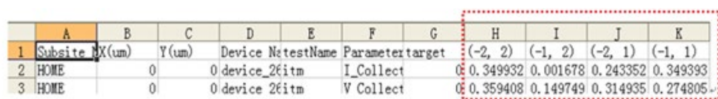
Figure 410: The save data folder

NOTE

The `.csv` files are created by selecting Automatically Save to .csv File after each wafer.

2. Select two wafers so two `.csv` files are created. The data is saved after creating each wafer. Open **Wafer_1_20080529092635.csv**. This wafer contains four test sites (the H, I, J, and K columns), as shown in the following figure.

Figure 411: The .csv file saved after each wafer



	A	B	C	D	E	F	G	H	I	J	K
1	Subsite N	X(um)	Y(um)	Device Na	testName	Parameter	target	(-2, 2)	(-1, 2)	(-2, 1)	(-1, 1)
2	HOME	0	0	device_26	itm	I_Collect	0	0.349932	0.001678	0.243352	0.349393
3	HOME	0	0	device 26	itm	V Collect	0	0.359408	0.149749	0.314935	0.274805

NOTE

The folder Wafer_1 and Wafer_2 are created by checking the Automatically save .csv file after each site.

3. Open folder Wafer_1. There are four `.csv` files. Because you selected four test sites in the previous steps, there are four `.csv` files, which are created after each site.

Figure 412: Open the Wafer_1 folder

4. Open a `.csv` file in …`\ACS\Projects\test\kdf\test_WaferInfo\Wafer_1`.
   This `.csv` is saved after each site, so there is only one site.

Figure 413: The .csv file saved after each site

	A	B	C	D	E	F	G	H
1	Subsite N	X(um)	Y(um)	Device Na	testName	Parameter	target	(-1, 1)
2	HOME	0	0	device_26	itm	I_Collect	0	0.349393
3	HOME	0	0	device_26	itm	V_Collect	0	0.274805

Check the Parameter in the column:

1. Check the Parameter in column in the data tab, then save and run this project again.
   Open `Wafer_1_20080529104534.csv`
   in …`\ACS\Projects\test\kdf\test_WaferInfo`.

Figure 414: The .csv file saved after each wafer and parameter

	A	B	C	D	E	F	G	H
1	Site	Subsite N	X(um)	Y(um)	Device Name	testName	V_Collect	I_Collector
2	(-2, 2)	HOME	0	0	device_26	itm	0.298966	0.09586225
3	(-1, 2)	HOME	0	0	device_26	itm	0.329251	0.34757908
4	(-2, 1)	HOME	0	0	device_26	itm	0.134039	0.04842533
5	(-1, 1)	HOME	0	0	device_26	itm	0.160677	0.27285038

NOTE

The Parameter in column F in the next figure is the same as rows 1 to 5 of column A in the previous figure.

2. Open
   `Site_n1p1_20080529104534.csv\\ACS\Projects\test\kdf\test_WaferInfo\Wafer_1`.

Figure 415: The .csv file saved after each site and parameter

	A	B	C	D	E	F	G	H
1	Site	Subsite N	X(um)	Y(um)	Device Name	testName	V_Collect	I_Collector
2	(-1, 1)	HOME	0	0	device_26	itm	0.160677	0.272850385

The Parameter in the column (see next two figures) is the same as the rows in the "Clear a parameter in a column" figure.

Figure 416: Select a parameter in a column

A	B	C	D	E	F	G	H
Site	Subsite N	X(um)	Y(um)	Device Name	testName	V_Pos	I_Pos
(-3, 2)	subsite	0	0	Resistor_1k	itm	1	0.000998
(0, 2)	subsite	0	0	Resistor_1k	itm	1	0.000998

Figure 417: Clear a parameter in a column

A	B	C	D	E	F	G	H	I
Subsite N	X(um)	Y(um)	Device Na	testName	Parameter	target	(-3, 2)	(0, 2)
subsite	0	0	Resistor_	itm	V_Pos	0	1	1
subsite	0	0	Resistor_	itm	I_Pos	0	0.000998	0.000998

All saved test results reported are based the times checked in the limits page.

Statistics

In this section:

Statistics introduction

Statistics is the science of making effective use of numerical data relating to groups of individuals or experiments. It deals with all aspects of this, including not only the collection, analysis, and interpretation of such data, but also the planning of the collection of data, in terms of the design of surveys and experiments.

Statistical methods can be used to summarize or describe a collection of data. This is useful in research, when communicating the results of experiments. In addition, patterns in the data may be modeled in a way that accounts for randomness and uncertainty in the observations, and are then used to draw inferences about the process being studied.

Using the ACS statistics feature, you can create an output file in a Keithley data file (`.kdf`). This feature also supports multiple `.kdf` files. By using this feature, you can do simple analysis on test results and make necessary adjustments and modify test conditions.

You can view the online data plot results while ACS testing is in progress.

Runtime monitoring

When ACS is running, you can do the statistical analysis at the same time. This saves you time and allows you to find any problems in the runtime monitoring. While in online mode, it allows you to automate all reports (including the statistical analysis) and notify you of abnormal yield levels or parameters that are not within specifications. Parameter specifications that are monitored in runtime include semiconductor test results, mean, sigma, and range levels.

Broad statistical analysis range

ACS can perform statistical analysis on your semiconductor data, from analyzing a mean, median, or sigma. Using the built-in editor, which has many filtering options, you can change the data and immediately see the effect on mean, median, sigma, or range. The built-in report helps you find the optimal parameter test limits.

Easy data import and control

You can start the semiconductor data analysis by either manually selecting the test data files from ACS for semiconductor characterization statistical data analysis, or using the running data automatically. ACS software interface offers a flexible interface to select test data files and filter the information to process for your semiconductor characterization or production analysis. When leveraging the ACS built in `.kdf` file, you can review the data file headers at insertion time.

Required files

Keithley data files (`.kdf`) for all plots when ACS is not running.

Statistics limitations

- Only supports the `.kdf` files for statistical analysis and does not support database.
- Only supports the common statistical modeling analysis.
- Does not support the ability to compare the history data with the online data.
- Supports all the test modules except PTM when ACS enters demo mode.
- Not compatible with previous versions; only supports ACS version 4.2 and higher.

Statistics settings

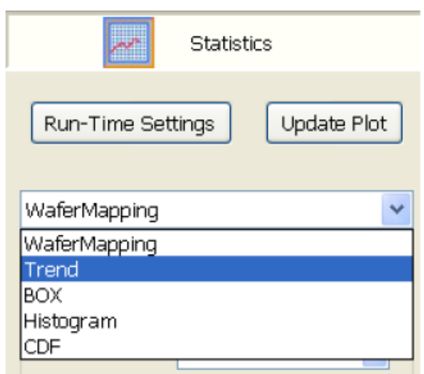
The statistics function is a comprehensive feature that covers statistics configuration and operations, including wafer mapping plots, trend plots, and BOX plots. It also includes a Histogram and cumulative distribution function (CDF).

Statistics configuration

To configure statistics:

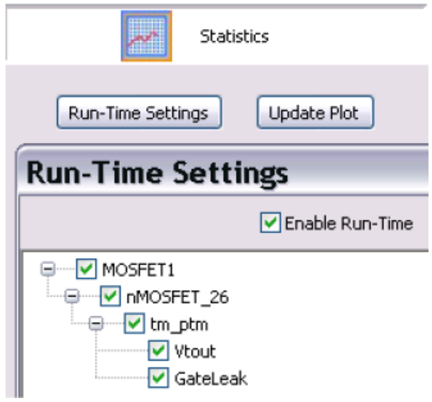
1. Select the plot type. You can select Wafer Mapping, Trend, BOX, Histogram, or CDF, as shown in the following figure.

Figure 418: Plot types



2. Configure the plot settings. The settings depend on the type of plot.
3. To set up runtime analysis, select **Enable Run-Time**, as shown in the following figure. Specify the variables you want to analyze. If you want to create statistics offline, you do not need to enable Run-Time Settings.

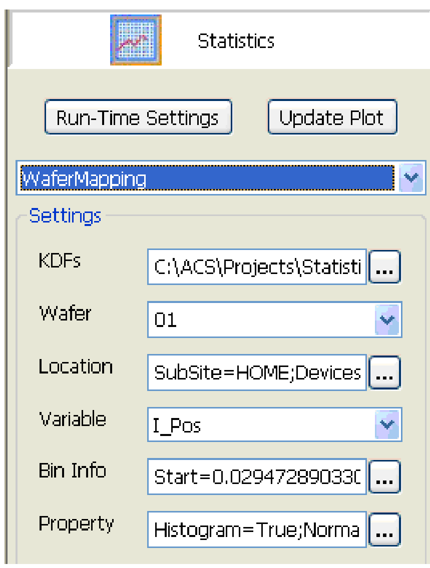
Figure 419: Run-Time Settings and Enable Run-Time



To run a statistical analysis:

1. Select the plot type and configure the plot settings.
2. Click **Update Plot**.
3. Repeat as necessary.

Figure 420: Statistics panel



Statistical analysis can be run in runtime mode or offline mode. When Enable Run-Time is selected, the plot is refreshed from memory data when ACS is running and you use the Run-Time Settings function. When Enable Run-Time is not selected, analysis is offline. The plots show the results of the selected `.kdf` files and you need to select the `.kdf` files that you want to analyze. In either mode, the plot is refreshed after clicking **Update Plot**.

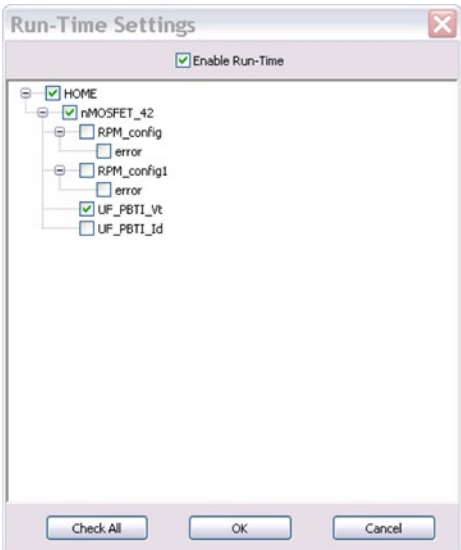
Statistics operations

The following steps show you how to do simple statistical analysis and must be done in the order as described.

NOTE

If you want to analyze a test in Runtime mode, you must select **Run-Time Settings** and choose **Enable Run-Time**. Then you must choose the devices you want to analyze in the configuration navigator.

Figure 421: Run-Time Settings of Statistics



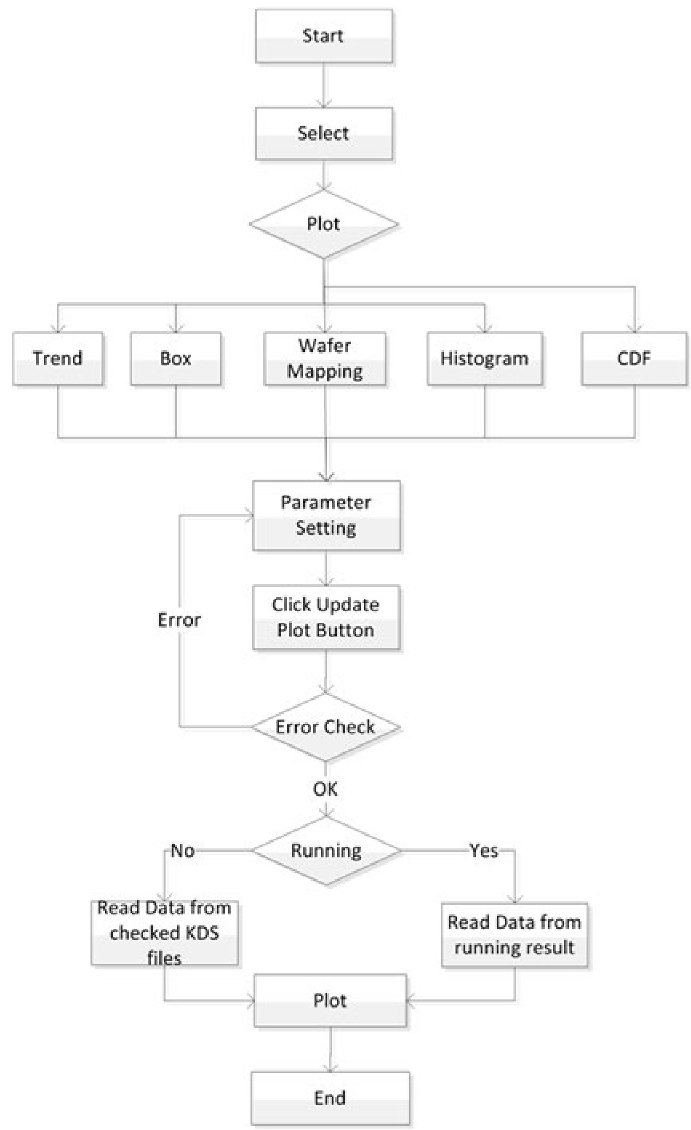
NOTE

If you are in demo mode, you must import the data file (`.kdf`) that you want to analyze.

To do simple statistical analysis:

1. Select the plot type you want to analyze.
2. Choose the Settings that you need, based on the plot type.
3. Select the Update Plot function.
4. Repeat these steps as many times as necessary.

Figure 422: Statistical operation flow chart

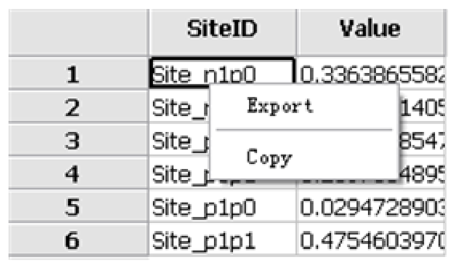


Plot types

There are five plot types available in the statistics feature. The parameter settings depend on the plot type selected (for example, Wafer Mapping, Trend, BOX, Histogram, and CDF).

There are different tabs available for each plot type. For example, the Wafer Mapping plot contains WaferMapping and data. If you select the Histogram or NormalCDF on property settings, you have these plots are available with the Update Plot function. You can export data from the data tab to a `.csv` file by right-clicking the data (see the following figure). You can also copy the selected data.

Figure 423: data tab Export



Wafer Mapping plot

The Wafer Mapping plot rapidly and automatically creates a map of the entire wafers with the testing results displayed. This enables rapid process feedback early to avoid adding expense to wafers or portions of wafers that may have poor yields. It also provides rapid feedback to the fabrication process, allowing you to adjust the process to improve the overall yield.

The Wafer Mapping plot works at the site level. One site on the wafer map is displayed by one state. That means only a single data point comes from every site for the variables that are used in the wafer mapping statistic function. You must locate the variable value by location, such as lot, wafer, or device, for the RTM, including the subdevice, and then switch the wafer to view the site distribution.

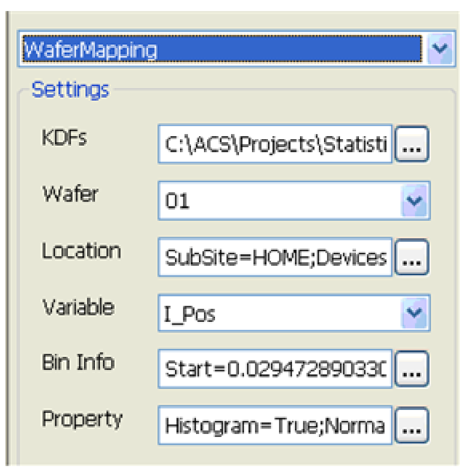
You can view the Histogram and CDF plot at the same time. Use the settings to determine your parameters.

After choosing the Wafer Mapping plot type, the parameters you need to configure are KDFs, Wafer, Location, Variable, Bin Info, and Property.

To set the wafer mapping settings:

1. For KDFs, select **...** to add a file.

Figure 424: Wafer Mapping GUI settings



2. Select **Add**.
3. Select the `.kdf` file that you want to upload. You can select more than one `.kdf` file by using **Ctrl** + click to select one file at time or **Shift** + click to select several files that are in sequence.
4. Choose the **Wafer**.

Figure 425: Wafer Mapping parameters Wafer selection

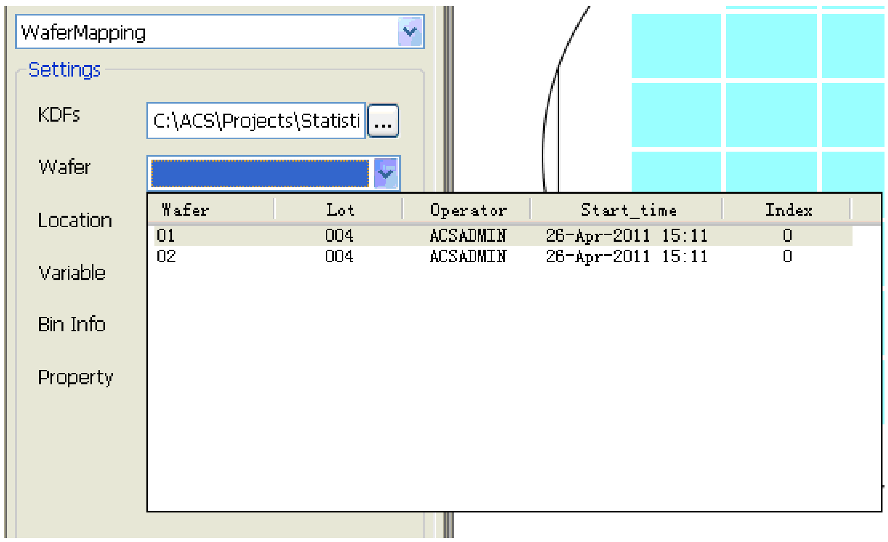
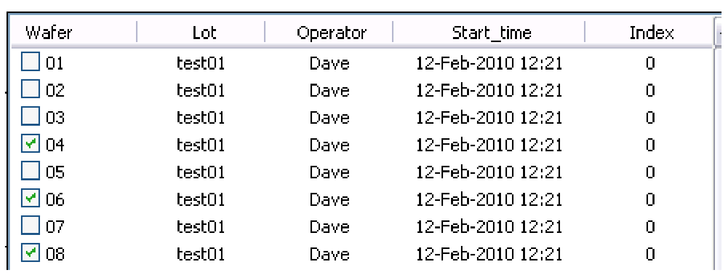
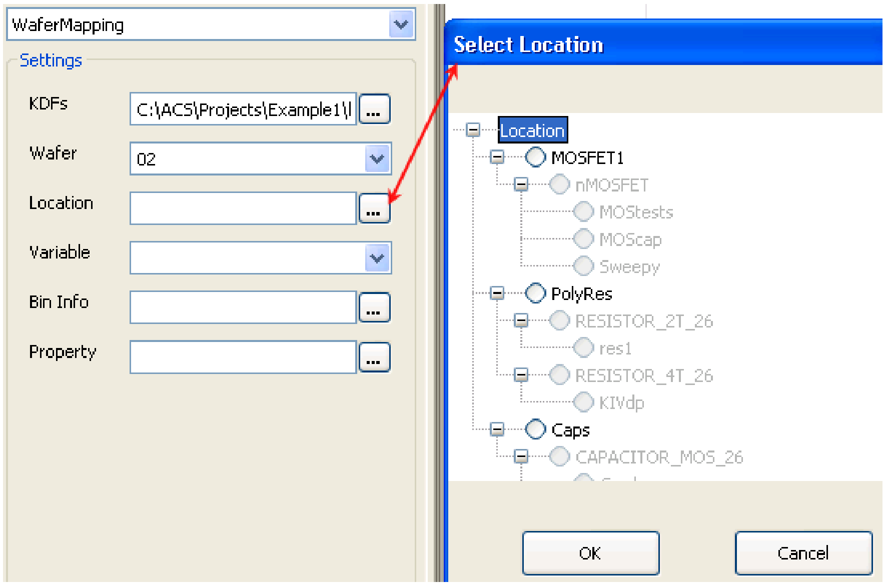


Figure 426: Trend parameters Wafers selection



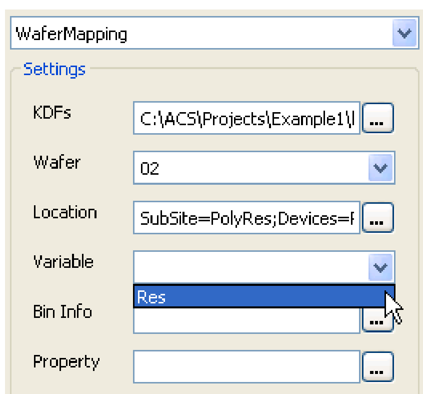
5. Choose the test module **Location**. You can only select one location.

Figure 427: Wafer Mapping parameters Location selection



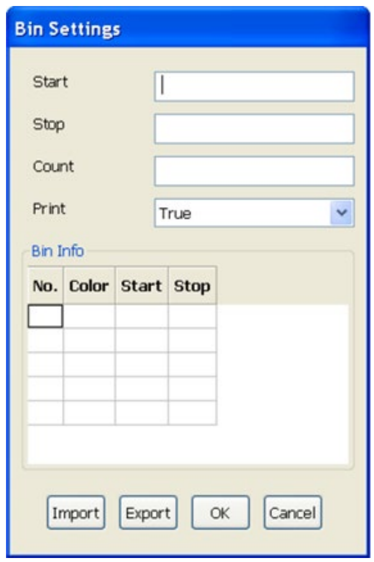
6. Choose the **Variable**.

Figure 428: Wafer Mapping parameters Variable selection



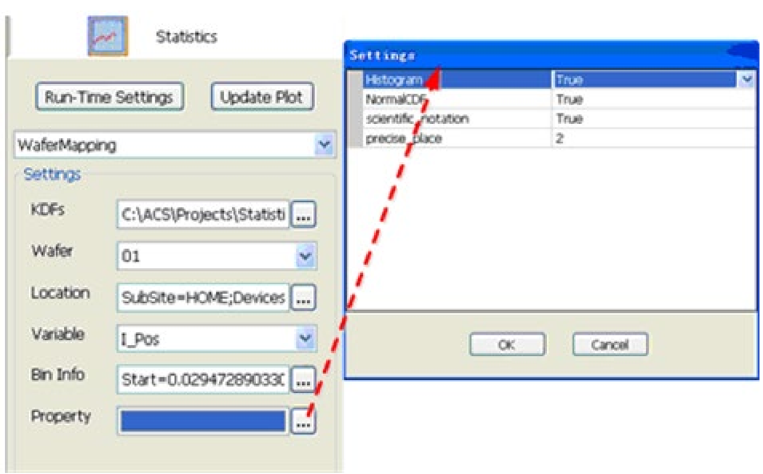
7. Select the **Bin Info**. The Bin Settings dialog is displayed. Enter the following information:
   - Start point value.
   - Stop point value.
   - Print. Set to True to print the data on the WaferMapping.
   - Import settings or export settings that you have created. Export saves the bin settings to `.csv` file that can be uploaded later for testing.

Figure 429: Wafer Mapping Bin info parameter



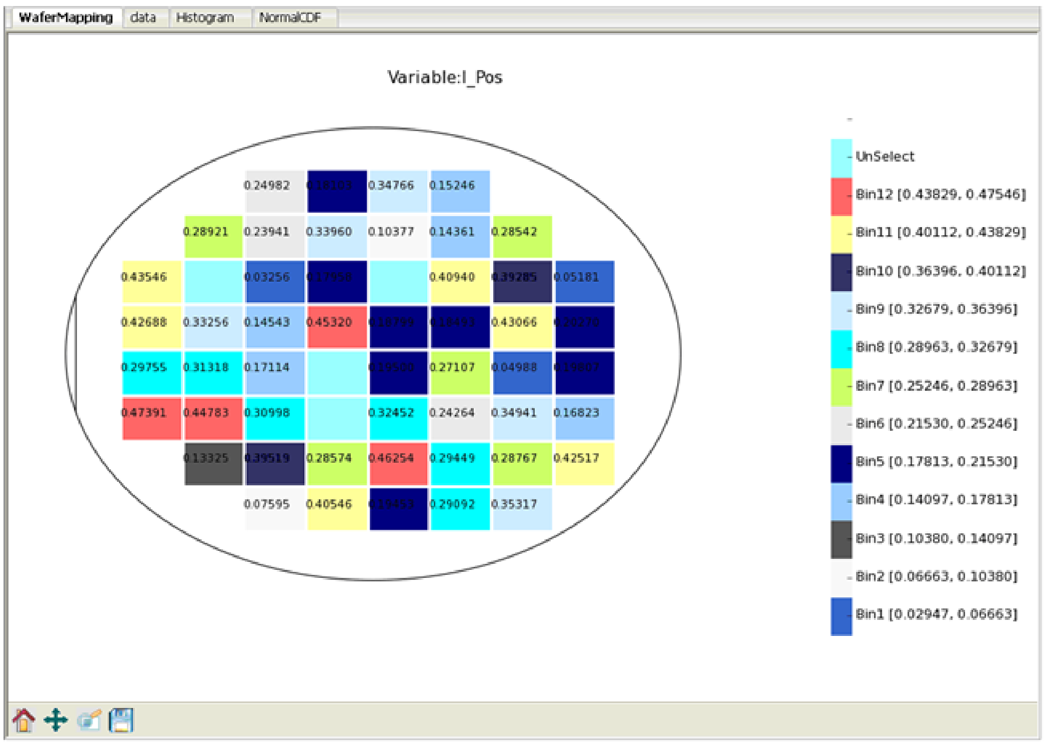
8. Define the Bin Info.
9. Select **OK**.
10. Select the ellipsis next to **Property**.

Figure 430: Wafer Mapping parameters Property



11. To generate a histogram of the data when you update the plot, set Histogram to **True**.
12. To generate the cumulative distribution function (CDF) plot when you update the plot, set NormalCDF to **True**.
13. To display Wafer Mapping data in scientific notation, set the scientific notation to **True**.
14. To set the number of effective digits of the data displayed in the Wafer Mapping, set to **precise_place**.
15. After you have set all the parameters, select **Update Plot** to see the newest plots. See the following figures.

Figure 431: Wafer Mapping plot



NOTE

You can select both the Histogram and NormalCDF plot. After you update the plots, the Wafer Mapping viewing area contains a tab for each plot. See the following figures.

Figure 432: Histogram plot pane

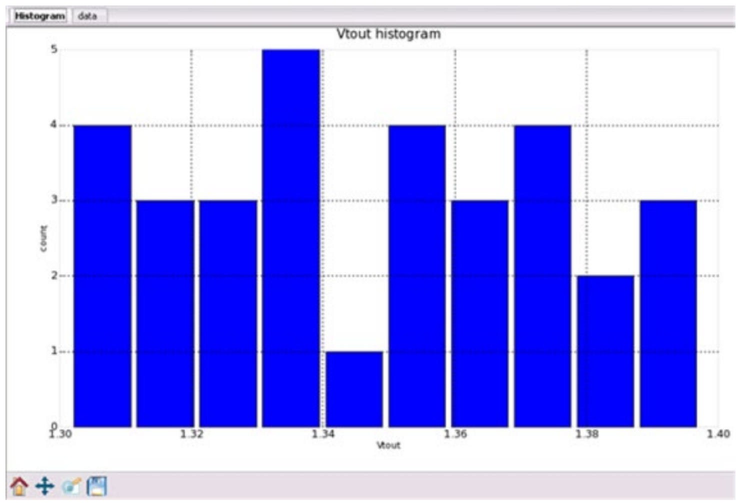
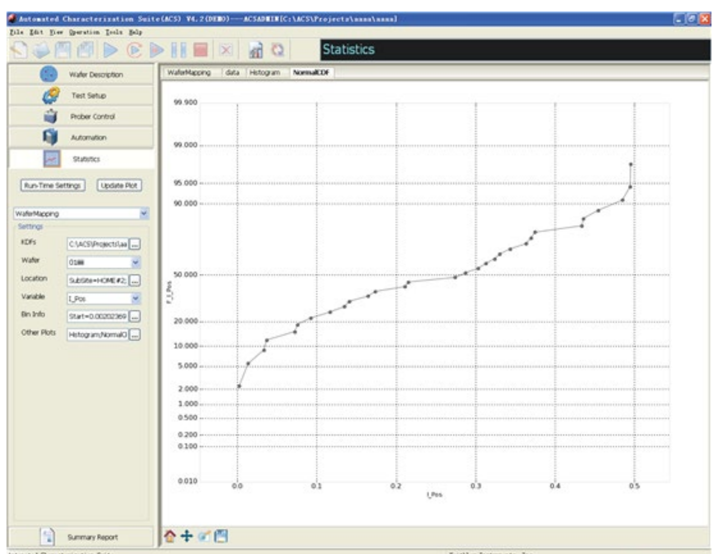


Figure 433: NormalCDF plot pane



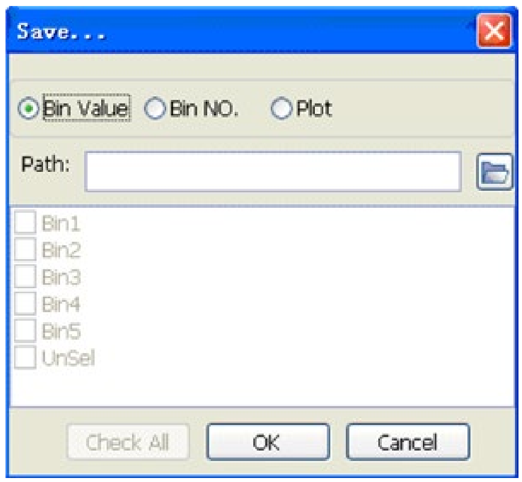
In the wafer mapping plot, you can save the plot by selecting save.

Figure 434: Wafer Map Plot icons

- **Bin Value:** Saves the coordinates and values of all the sites to a `.vbf` file. If the site contains no data, the value is −1 by default.
- **Bin NO (Bin Number):** Saves the coordinates and bin numbers of the sites in the selected bins to a `.nbf` file. If the site is not in any bin, the bin number is −1 by default.
- **Plot:** Saves the plot to a `.png`, `.eps`, or `.svg` file.

The following is the Save dialog for the Bin value, Bin NO. or Plot.

Figure 435: Wafer Mapping Save



Trend plot

The Trend plot is at the wafer level. That means a single point comes from every wafer for the selected variable. The mean, minimum, and maximum values, which are calculated from all site values, are shown in the plot for each wafer.

The limit information of the variable and current test data is loaded automatically when ACS is testing. Select multiple `.kdf` files by selecting the `.kdf` files parameter settings if ACS is not running. You see the updated plot when you select the Update Plot function.

The following figure shows the Trend plot GUI settings.

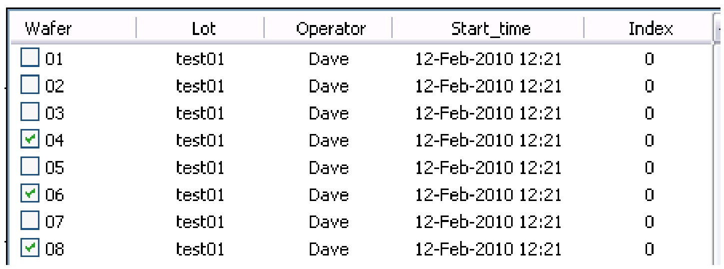
Figure 436: Trend Plot GUI settings

Choosing the parameter settings is like the Wafer Mapping settings. The parameters you need to configure are KDFs, Wafers, Location, Variable, Filter, and Limit.

To set the trend settings:

1. Select **...** next to the KDFs function to add a file.
2. Select **Add**, as shown in the following figure.
3. Choose the `.kdf` file that you want to upload. You can select more than one `.kdf` file by selecting Ctrl+click and selecting one file at time, or by selecting Shift+click to select several files that are in sequence.
4. Choose the Wafer (or more than one) from the list.

Figure 437: Trend parameters Wafers selection



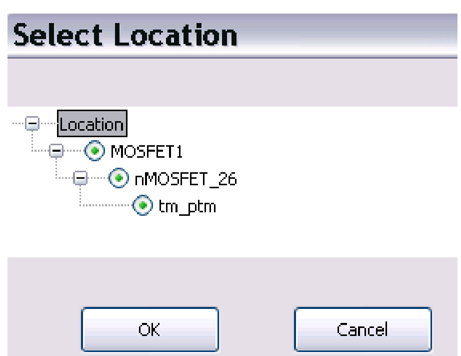
Wafer	Lot	Operator	Start_time	Index
☐ 01	test01	Dave	12-Feb-2010 12:21	0
☐ 02	test01	Dave	12-Feb-2010 12:21	0
☐ 03	test01	Dave	12-Feb-2010 12:21	0
☑ 04	test01	Dave	12-Feb-2010 12:21	0
☐ 05	test01	Dave	12-Feb-2010 12:21	0
☑ 06	test01	Dave	12-Feb-2010 12:21	0
☐ 07	test01	Dave	12-Feb-2010 12:21	0
☑ 08	test01	Dave	12-Feb-2010 12:21	0

5. Choose the test module Location.

NOTE

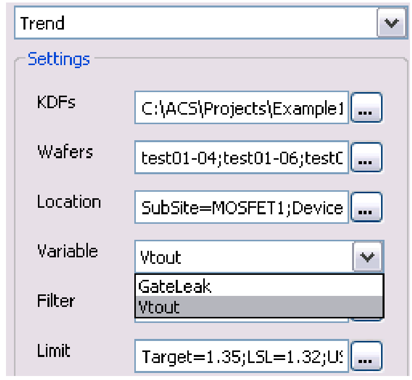
Only one Location can be selected.

Figure 438: Trend parameters Location selection



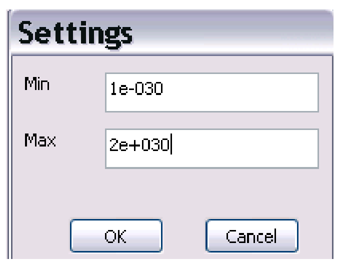
6. Choose the Variable from the list.

Figure 439: Trend parameters Variable selection



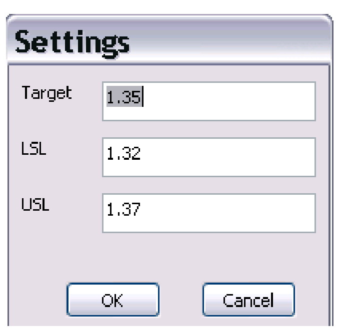
7. Next to Filter, select **...** The Settings dialog opens.
8. Set the minimum and maximum of the variable that you choose based on the `.kdf` file selected. For example, you must enter the minimum setting and maximum setting that you want to measure in your plot.

Figure 440: Trend parameters Filter Settings



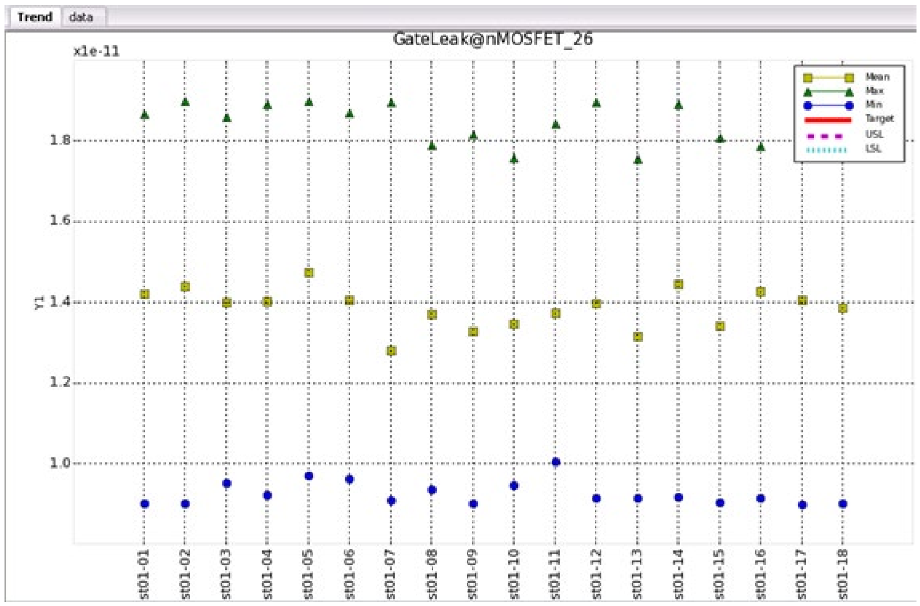
9. Next to Limit, select **...**
10. Set the Target specification limit, the lower specification limit (LSL), and the upper specification limit (USL). The default value for the Target, LSL, and USL is determined by the limit files or limit settings in ACS.

Figure 441: Trend parameters Limit Settings



11. Select **Update Plot** to see the newest plots. There are two tabs where you can view the plot information: Trend and data.

Figure 442: Trend plot tabs

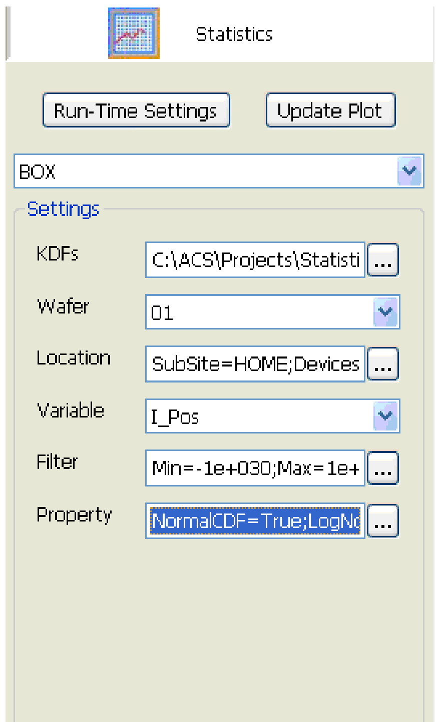


BOX plot

The BOX plot is at the lot level. The BOX plot helps you to display differences between populations without making any assumptions of the underlying statistical distribution. In other words, you can see differences between lots for selected variables. You can select the lots by selecting multiple `.kdf` file parameter settings and you can select different devices and different variables to see the differences between the selected wafers.

The following figure shows the BOX settings dialog.

Figure 443: BOX GUI settings

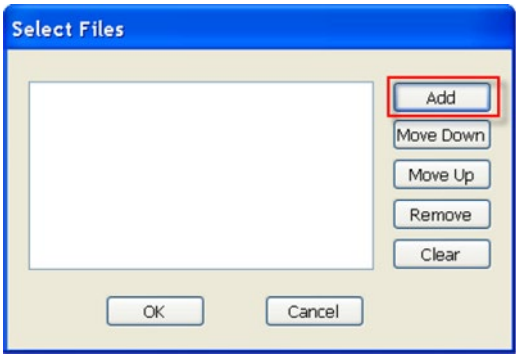


Choosing the parameter settings is like the Wafer Mapping settings. The parameters you need to configure are KDFs, Wafer, Location, Variable, Filter, and Property.

To set the BOX Settings:

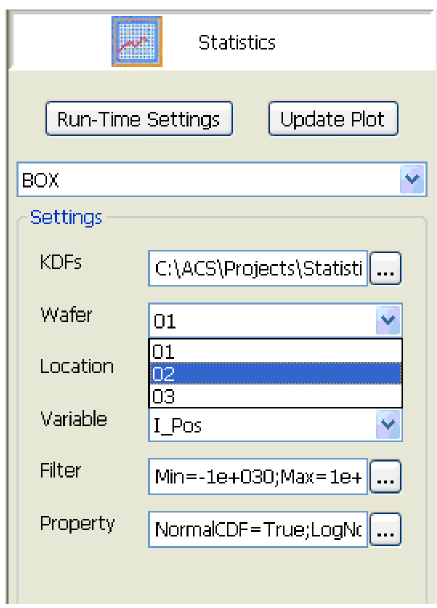
1. Next to KDFs, select **...**.
2. Select **Add**. The Select Files dialog is displayed, as shown in the following figure.

Figure 444: Add ,kdf files



3. Select the `.kdf` files that you want to upload. To select more than one `.kdf` file, use Ctrl+click to select one file at a time, or use Shift+click to select several files that are in sequence.
4. Choose the Wafer from the list.

Figure 445: BOX parameters Wafer selection

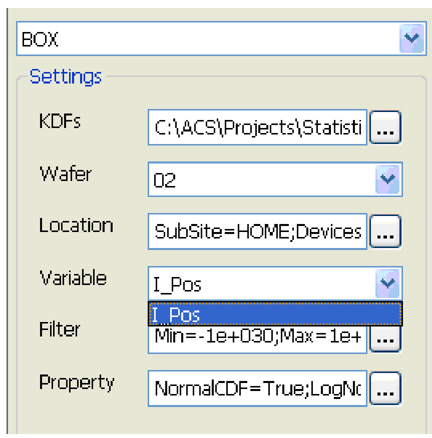


5. Choose the test module Location. Only one location can be selected.

Figure 446: Trend parameters Location selection

6. Choose the Variable from the list.

Figure 447: BOX parameters Variable Settings

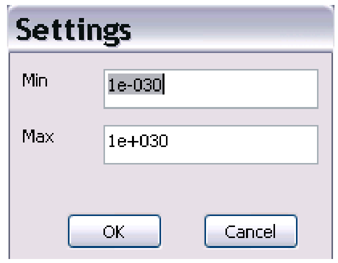


NOTE

When you choose True, you see a percentage change from the previous lot to the last lot. When you choose False, you see all lots used in the plot.

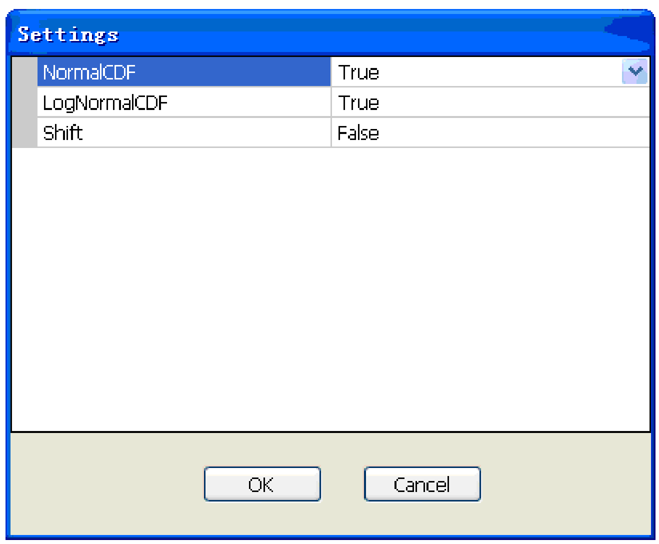
7. Next to Filter, select **…**. The Settings dialog is displayed.
8. Set the minimum and maximum of the variable that you chose based on the `.kdf` file selected. For example, you must enter the minimum setting and maximum setting that you want to measure in your plot, as shown in the following figure.

Figure 448: Filter Settings



9. Next to Property, select **…**. The Settings dialog is displayed.

Figure 449: BOX parameters Property



10. Select **NormalCDF**, **LogNormalCDF**, or Shift:
    - **NormalCDF:** (True) Generates the NormalCDF plot.
    - **LogNormalCDF:** (True) Generates the LogNormalCDF plot.
    - **Shift:** (True) Computes and plots the percentage shift of each selected lot (`.kdf`) as compared to the lot with the earliest time stamp. For example, if three `.kdf` files are selected, the file with the earliest timestamp is used as the baseline. The other two lots are compared to the baseline lot and the equation below is used in the calculation:

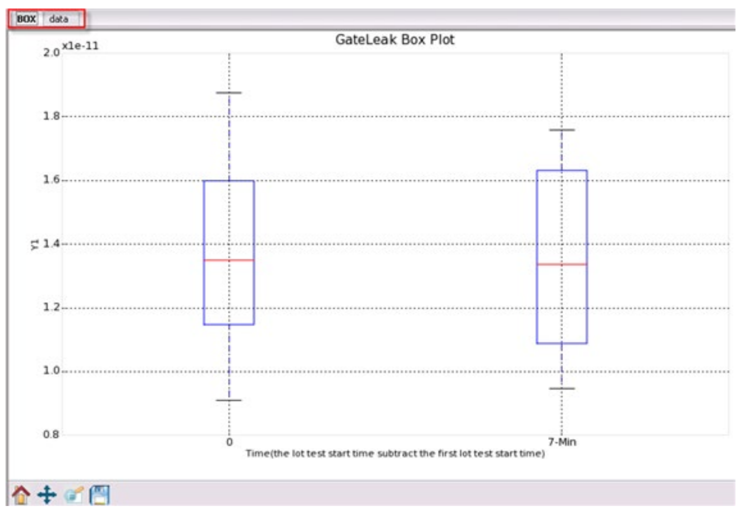
$$\%\ \text{difference} = \frac{(\text{lot 2 result} - \text{baseline result})}{\text{baseline result}} \ \text{X}\ 100$$

11. Select **Update Plot** to see the newest plots. You can view the plot information in the BOX and data tabs, as shown in the following figure.

NOTE

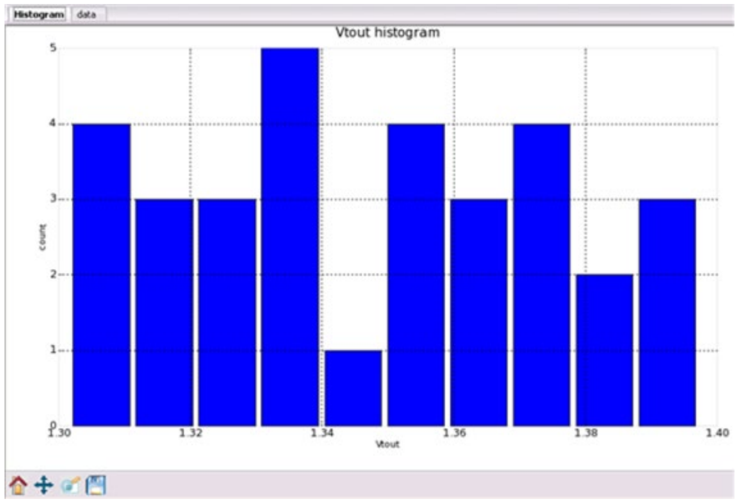
The bottom of the BOX is the 25th percentile of the lot and the top is the 75th percentile. The red band represents the 50th percentile (the median). The whiskers of the lot define the length of the whiskers as a function of the inner quartile range. They extend to the most extreme data point. The default is 1.5 IQR (interquartile range).

Figure 450: BOX plot tabs



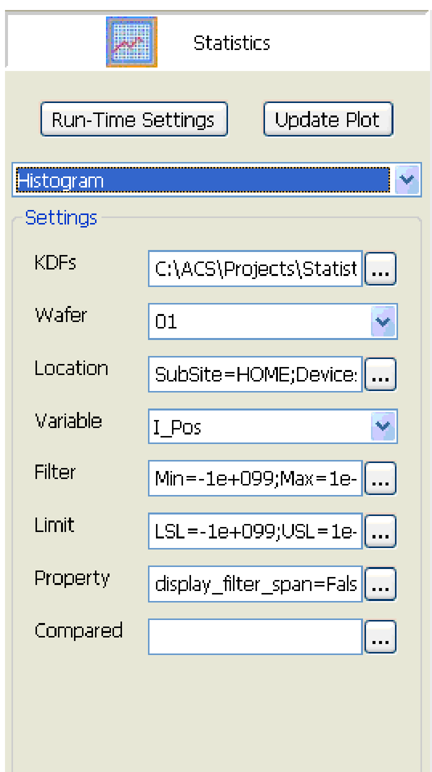
Histogram

The Histogram feature is a graphical display of tabulated frequencies indicated with a bar graph, as shown in the following figure.



The following figure shows the Histogram GUI parameter settings.

Figure 451: Histogram GUI settings

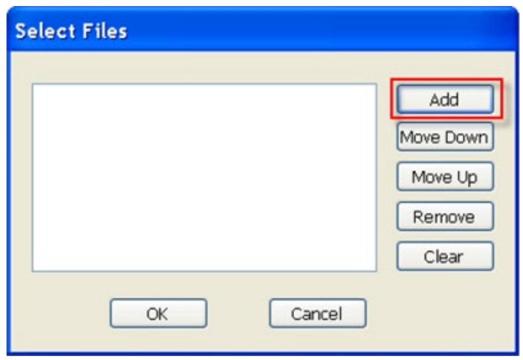


Choosing the parameter settings is like the Wafer Mapping settings. The parameters you need to configure are KDFs, Wafer, Location, Variable, Filter, Limit, Property, and Compared.

To set the Histogram Settings:

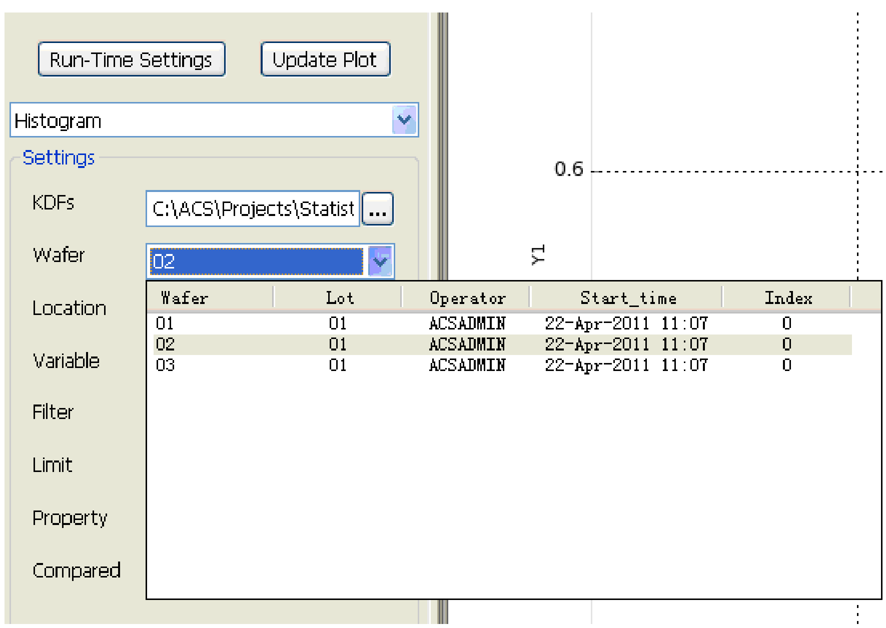
1. Next to KDFs, select **...**.
2. Select **Add**.

Figure 452: Add ,kdf files



3. Choose the `.kdf` file that you want to upload. You can select more than one `.kdf` file using **Ctrl+click** to select one file at time, or by using **Shift+click** to select several files that are in sequence.
4. Choose the Wafer from the list.

Figure 453: Histogram parameters Wafer selection

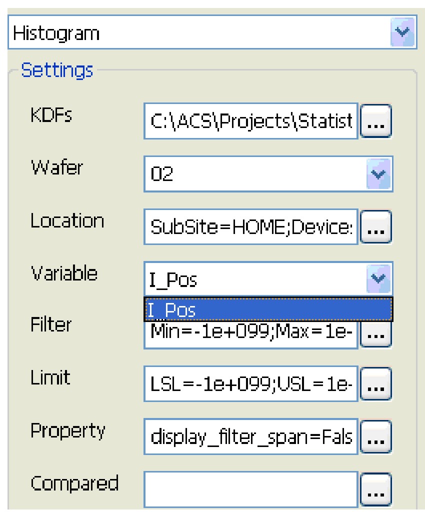


5. Choose the test module Location. Only one Location can be selected.

Figure 454: Trend parameters Location selection

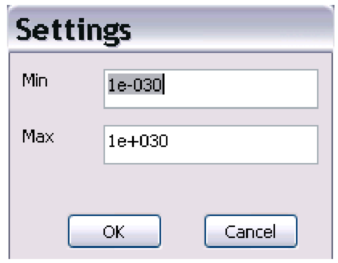
6. Choose a Variable.

Figure 455: Histogram parameters Variable Settings



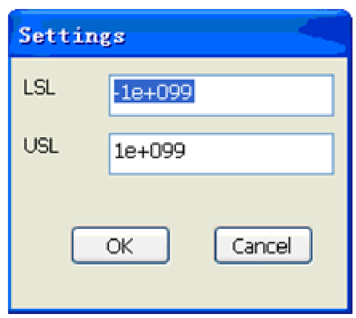
7. Next to Filter, select **...**. The Settings dialog is displayed.
8. Set the minimum and maximum of the variable based on the `.kdf` file selected. For example, you must enter the minimum setting and maximum setting that you want to measure in your plot.

Figure 456: Filter Settings



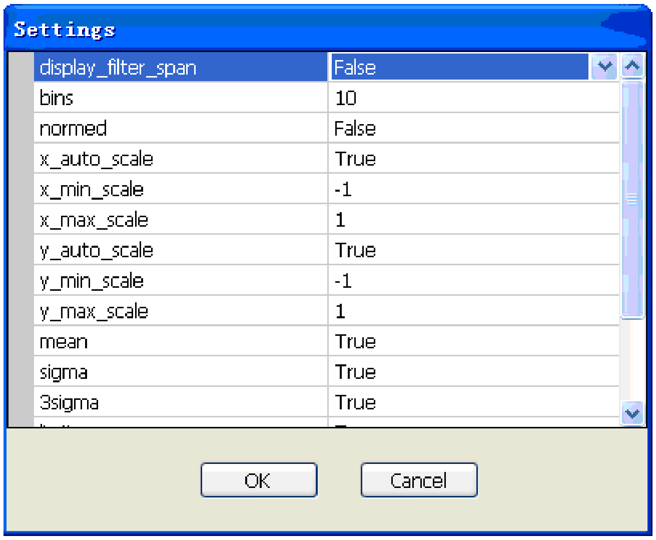
9. Next to Limit, select **...**.
10. Set the lower specification limit (LSL), and the upper specification limit (USL).

Figure 457: Histogram Limit Settings



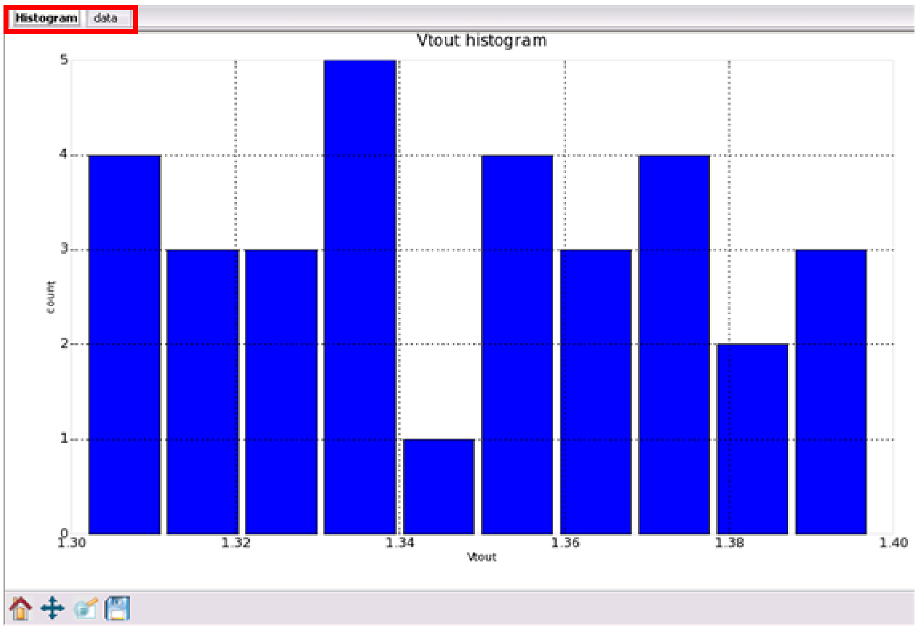
11. Next to Property, select **....**

Figure 458: Histogram Property Settings



12. Enter the following settings.
    - **display_filter_span:** If this is False and x_auto_scale is True, the histogram is generated according to the bins number. In other conditions, the histogram is generated according to the bins span.
    - **bins:** Sets the bins number.
    - **normed:** (True) sets counts to normalized.
    - **x_auto_scale:** (True) x-axis is auto-scaled.
    - **x_min_scale:** Minimum value of x-axis when x_auto_scale is False.
    - **x_max_scale:** Maximum value of x-axis when x_auto_scale is False.
    - **y_auto_scale:** (True) y-axis is auto-scaled.
    - **y_min_scale:** Minimum value of y-axis when y_auto_scale is False.
    - **y_max_scale:** Maximum value of y-axis when y_auto_scale is False.
    - **mean:** (True) displays mean value on the histogram.
    - **sigma:** (True) displays 1σ sigma limit on the histogram.
    - **3sigma:** (True) displays 3σ sigma limit on the histogram.
    - **limit:** (True) displays limit values in the histogram.
    - **PDF:** (True) displays probability density function overlayed on the histogram.
    - **Cpk:** (True) displays the process capability overlayed on the histogram.
13. Next to Compared, select **…**.
14. Add a lot to compare. The histogram displays the differences. Leave the Compared option blank when you do not want to compare the differences.
15. Select **Update Plot** to see the newest plots. You can view plot information on the Histogram and data tabs, as shown in the following figure.

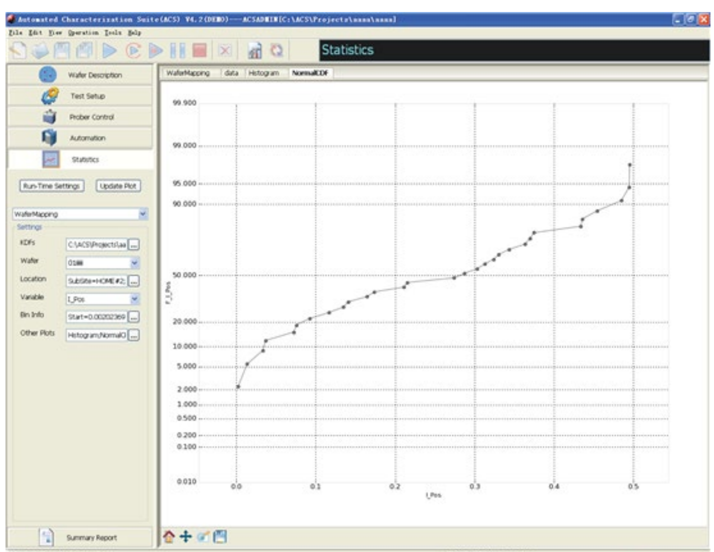
Figure 459: Histogram plot tabs



Cumulative distribution function (CDF)

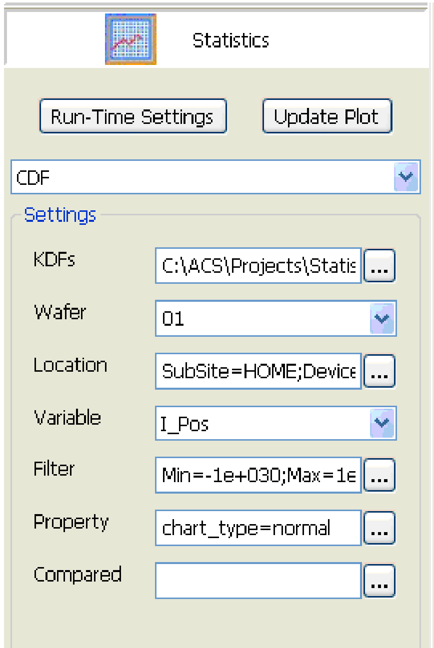
In ACS software, the CDF feature is a graphical display of probability distribution that is indicated with a line plot, as shown in the following figure.

Figure 460: Normal CDF plot pane



The following figure shows the CDF GUI parameter settings dialog.

Figure 461: CDF GUI parameter settings

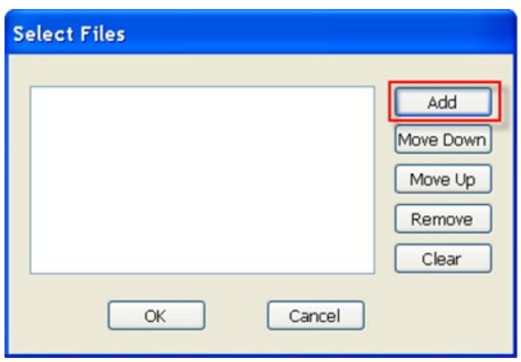


Choosing the parameter settings is like the Histogram settings. The parameters you need to configure are KDFs, Wafer, Location, Variable, Filter, Property, and Compared.

To set the CDF settings:

1. Next to KDFs, select ….
2. Select Add, as shown in the following figure. To select more than one `.kdf` file, use **Ctrl+click** and select one file at time, or use **Shift+click** to select several files that are in sequence.

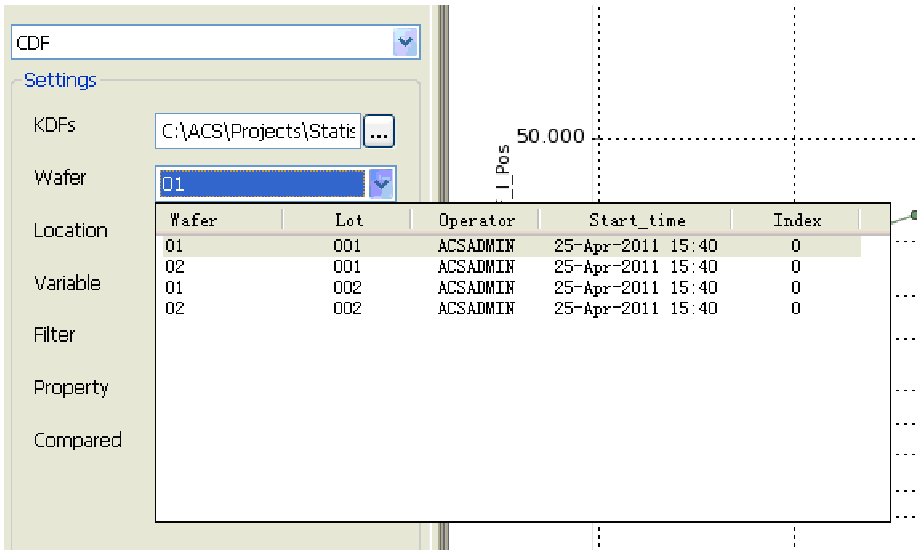
Figure 462: Add ,kdf files



3. Choose the `.kdf` file that you want to upload.

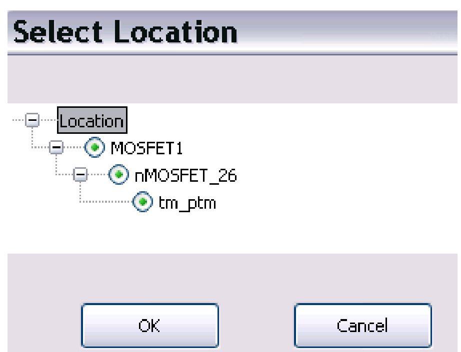
4. Choose the Wafer from the list (see the following figure).

Figure 463: CDF parameters Wafer selection



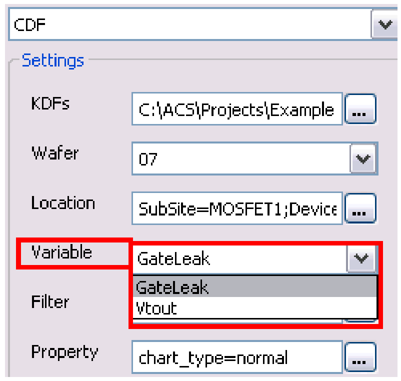
5. Choose the test module Location. Only one location can be selected.

Figure 464: Trend parameters Location selection



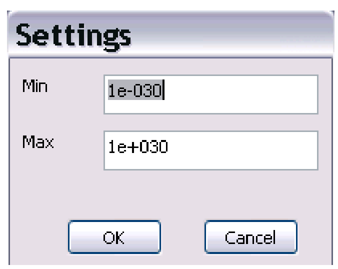
6. Choose the Variable from the list.

Figure 465: CDF parameters Variable Settings



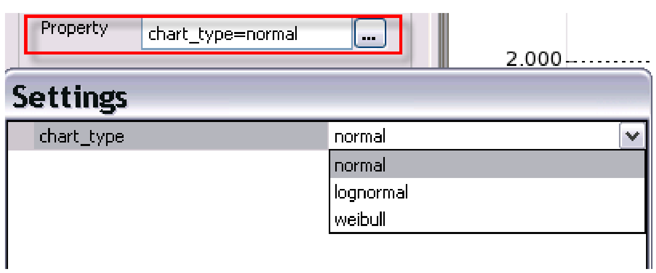
7. Next to Filter, select **...**. The Settings dialog is displayed.
8. Set the minimum and maximum of the variable that you chose based on the `.kdf` file. For example, you must enter the minimum setting and maximum setting that you want to measure in your plot.

Figure 466: Filter Settings



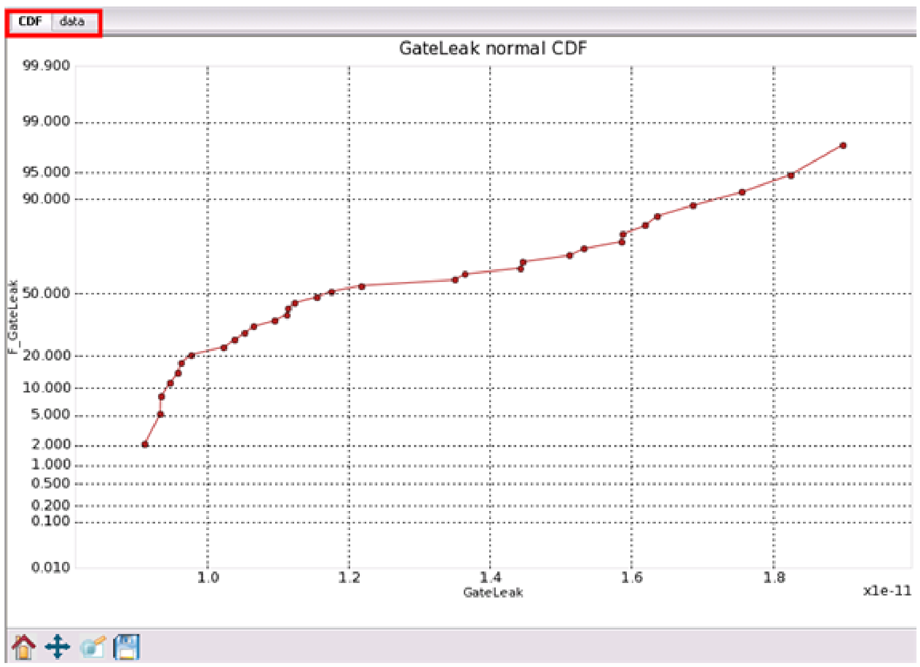
9. Next to Property, select **...**. The Settings dialog is displayed.
10. Select the type of CDF: normal, lognormal, or Weibull, as shown in the following figure.

Figure 467: Types of CDFs



11. Next to Compared, select **...**.
12. Add a lot to compare. The histogram displays the differences. Leave the Compared option blank if you do not want to compare the differences.
13. Select Update Plot to see the newest plots. You can view information on the CDF and data tabs, as shown in the following figure.

Figure 468: CDF plot tabs



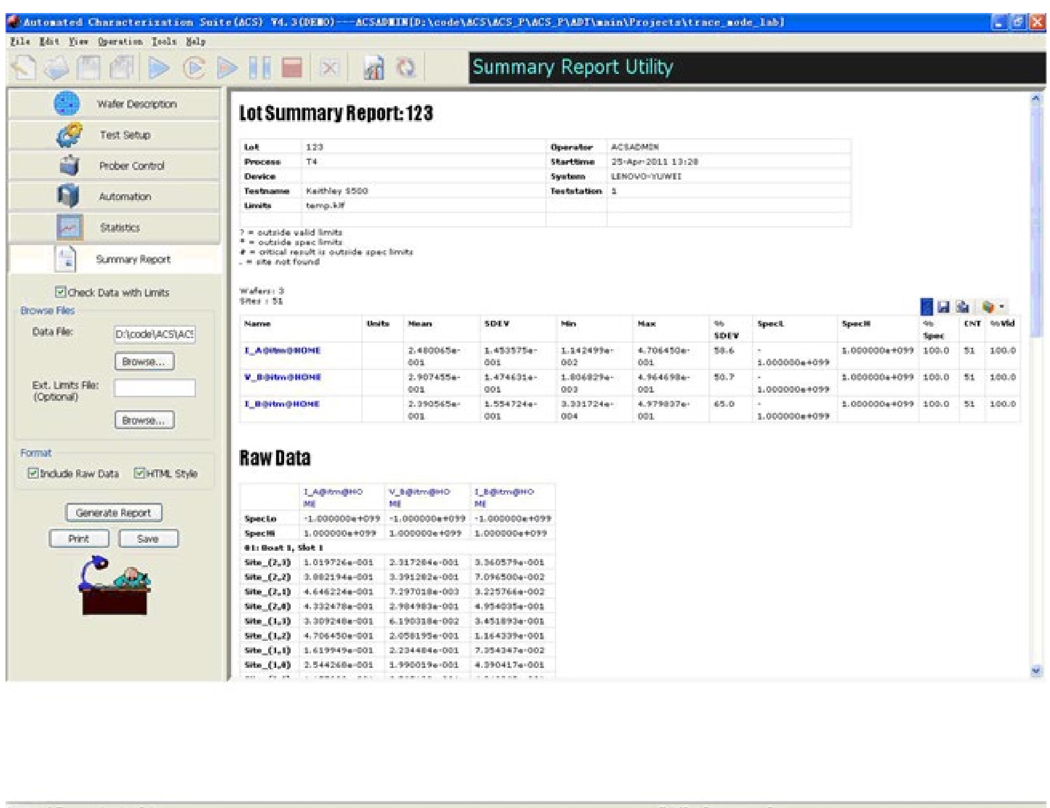
Summary report

In this section:

Summary report panel

Select the Summary Report function from the View panel, and the Summary Report panel displays in the Edit panel. A blank Lot Summary Report displays in the work area.

Figure 469: Lot Summary Report



- **Check Data with Limits:** Used to choose whether to check the testing results with the limits setting in the test plan.
- **Data File:** Used to choose a Keithley data file (`.kdf`). Select the Browse function and choose the target data file (`.kdf`).
- **Ext. Limits File (Optional):** Used to choose an external Keithley limits file (`.kdf`). This option is available only after the Check Data with Limits option is selected. Select the **Browse** function and choose a limits file to replace the test plan limits setting.
- **Include Raw Data:** Determines if the report contains Raw Data. By default, this checkbox is cleared.
- **HTML Style checkbox:** Used to select the style of report, HTML, or plain text.
- **Generate Report function:** Select to start generating a summary report.
- **Print function:** to print the generated report shown in the work area.
- **Save function:** Select to save the current report.

Generating a summary report

After testing is completed, you can generate a summary report and view the results of a test in the summary report. The summary report contains such information as testing time, user name, and measured data from a Keithley data file `.kdf`, and pass/fail information based on test data and limits setting. The Summary Report function provides the following capabilities:

- You can generate a Summary Report and a Raw Data report. The Summary Report is a summarized report that includes means, standard deviations, and other statistics. The Raw Data report is a detailed report that includes all the raw data.

Figure 470: Lot Summary Report demo

Wafer Summary Report: 1

Lot	1	UserName	ACSADMIN
Process	T4	Starttime	12-Nov-2012 11:11
Device		System	JOJO-8D28F7D7FE
Testname	Keithley S500	Teststation	1
Limits	temp.klf		

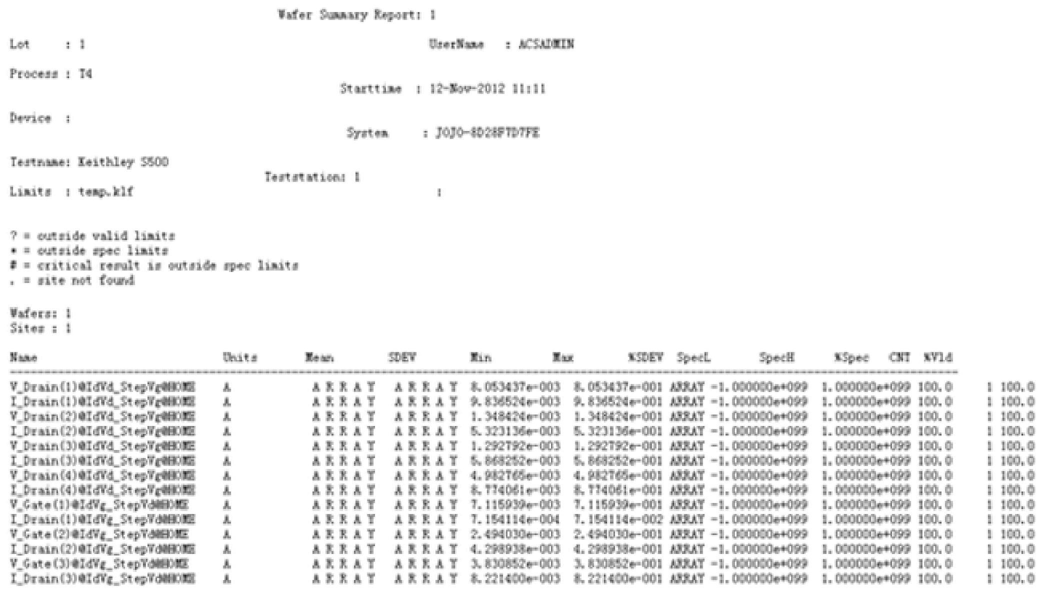
? = outside valid limits
* = outside spec limits
# = critical result is outside spec limits
. = site not found

Wafers: 1
Sites : 1

Name	Units	Mean	SDEV	Min	Max	%SDEV	SpecL	SpecH	%Spec	CNT	%Vld
V_Drain(1) @IdVd_StepVg@HOME	A	ARRAY	ARRAY	8.053437e-003	8.053437e-001	ARRAY	-1.000000e+099	1.000000e+099	100.0	1	100.0
I_Drain(1) @IdVd_StepVg@HOME	A	ARRAY	ARRAY	9.836524e-003	9.836524e-001	ARRAY	-1.000000e+099	1.000000e+099	100.0	1	100.0
V_Drain(2) @IdVd_StepVg@HOME	A	ARRAY	ARRAY	1.348424e-003	1.348424e-001	ARRAY	-1.000000e+099	1.000000e+099	100.0	1	100.0
I_Drain(2) @IdVd_StepVg@HOME	A	ARRAY	ARRAY	5.323136e-003	5.323136e-001	ARRAY	-1.000000e+099	1.000000e+099	100.0	1	100.0
V_Drain(3) @IdVd_StepVg@HOME	A	ARRAY	ARRAY	1.292792e-003	1.292792e-001	ARRAY	-1.000000e+099	1.000000e+099	100.0	1	100.0
I_Drain(3) @IdVd_StepVg@HOME	A	ARRAY	ARRAY	5.868252e-003	5.868252e-001	ARRAY	-1.000000e+099	1.000000e+099	100.0	1	100.0

- You can generate an HTML script report and a plain text report. In an HTML-style raw report, a hyperlink is set on each row label of the summary table and on the array results. If you click the row label, the cursor will jump to the corresponding column in the Raw Data table. If you click an array that is shown as a hyperlink, it will expand.

Figure 471: Plain Text format report



```
                                   Wafer Summary Report: 1

Lot     : 1                                        UserName   : ACSADMIN

Process : T4
                                        Starttime  : 12-Nov-2012 11:11

Device  :
                                         System     : JOJO-8D28FTD7FE

Testname: Keithley S500
                                 Teststation: 1
Limits  : temp.klf                                    :

? = outside valid limits
* = outside spec limits
# = critical result is outside spec limits
. = site not found

Wafers: 1
Sites : 1

Name                          Units     Mean       SDEV       Min           Max           %SDEV  SpecL           SpecH          %Spec  CNT  %Yld
------------------------------------------------------------------------------------------------------------------------------------------------
V_Drain(1)@IdVd_StepVg@HOME   A         A R R A Y  A R R A Y  8.053437e-003 8.053437e-001 ARRAY -1.000000e+099  1.000000e+099 100.0     1 100.0
I_Drain(1)@IdVd_StepVg@HOME   A         A R R A Y  A R R A Y  9.836524e-003 9.836524e-001 ARRAY -1.000000e+099  1.000000e+099 100.0     1 100.0
V_Drain(2)@IdVd_StepVg@HOME   A         A R R A Y  A R R A Y  1.348424e-003 1.348424e-001 ARRAY -1.000000e+099  1.000000e+099 100.0     1 100.0
I_Drain(2)@IdVd_StepVg@HOME   A         A R R A Y  A R R A Y  5.323136e-003 5.323136e-001 ARRAY -1.000000e+099  1.000000e+099 100.0     1 100.0
V_Drain(3)@IdVd_StepVg@HOME   A         A R R A Y  A R R A Y  1.292792e-003 1.292792e-001 ARRAY -1.000000e+099  1.000000e+099 100.0     1 100.0
I_Drain(3)@IdVd_StepVg@HOME   A         A R R A Y  A R R A Y  5.868252e-003 5.868252e-001 ARRAY -1.000000e+099  1.000000e+099 100.0     1 100.0
V_Drain(4)@IdVd_StepVg@HOME   A         A R R A Y  A R R A Y  4.982765e-003 4.982765e-001 ARRAY -1.000000e+099  1.000000e+099 100.0     1 100.0
I_Drain(4)@IdVd_StepVg@HOME   A         A R R A Y  A R R A Y  8.774061e-003 8.774061e-001 ARRAY -1.000000e+099  1.000000e+099 100.0     1 100.0
V_Gate(1)@IdVg_StepVd@HOME    A         A R R A Y  A R R A Y  7.115939e-003 7.115939e-001 ARRAY -1.000000e+099  1.000000e+099 100.0     1 100.0
I_Drain(1)@IdVg_StepVd@HOME   A         A R R A Y  A R R A Y  7.154114e-004 7.154114e-002 ARRAY -1.000000e+099  1.000000e+099 100.0     1 100.0
V_Gate(2)@IdVg_StepVd@HOME    A         A R R A Y  A R R A Y  2.494030e-003 2.494030e-001 ARRAY -1.000000e+099  1.000000e+099 100.0     1 100.0
I_Drain(2)@IdVg_StepVd@HOME   A         A R R A Y  A R R A Y  4.298938e-003 4.298938e-001 ARRAY -1.000000e+099  1.000000e+099 100.0     1 100.0
V_Gate(3)@IdVg_StepVd@HOME    A         A R R A Y  A R R A Y  3.830852e-003 3.830852e-001 ARRAY -1.000000e+099  1.000000e+099 100.0     1 100.0
I_Drain(3)@IdVg_StepVd@HOME   A         A R R A Y  A R R A Y  8.221400e-003 8.221400e-001 ARRAY -1.000000e+099  1.000000e+099 100.0     1 100.0
```

- You can sort tested data in the reports according to Limits, Critical, or Results.

Settings for each of these reports are discussed in more detail in Summary report panel (on page 9-1).

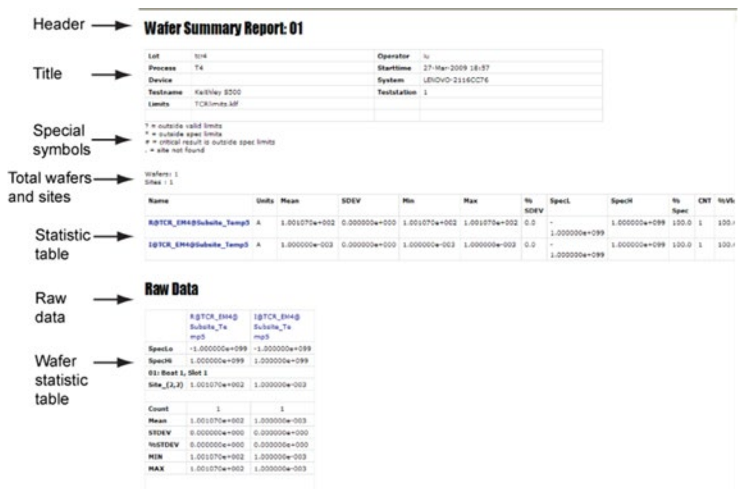
As seen in the previous figures, each report has two distinct parts – Lot Summary Report and Raw Data. Each is in an HTML-style and in plain text formats.

Each report is preceded by a report header that is called the Lot Summary Report. The report header displays information about the lot, process, device, testing time, and user name. The header lines are displayed on each report page. Raw Data consists of all raw tested data. You can choose whether you want to display Raw Data.

Understanding the lot summary report

Header: A header is placed at the top of the Lot Summary Report. An empty field in the header indicates that the corresponding field in the data file is empty at the time of searching. The data is listed with test names as row labels on the left and test information is in columns across the top. The test information consists of the mean, minimum, and maximum for the sites based on the executed testing plan.

Figure 472: Wafer Summary Report



Title: If only one wafer is saved in the presently loaded `.kdf` file, the title is "Wafer Summary Report: %wafer id%". If a multiwafer is saved in the `.kdf` file, then the title is "Lot Summary Report: %wafer id%".

Following the header is a series of column field labels. These fields contain data derived from the `.kdf` file. Some fields, such as %Spec and %Vld, may include an asterisk next to the data points to indicate the data is out of range, alerting you to notice potentially bad data points. The following paragraphs describe the fields that serve as the column indicators for the standard report shown in the above two figures.

- **Lot/Wafer:** If single wafer option has been selected before generating the `.kdf` file, the label is "Wafer," and the wafer id value is the same as the title. Otherwise, the label is "Lot." Lot id value is the one input in the Auto-Test Information box in Automation. This value is saved in the `.kdf` file.

NOTE

There are two conditions that generate a one-wafer `.kdf` file. One is in single wafer mode, the title is "Wafer Summary Report: %wafer id%", and the lot/wafer label is "Wafer." The other condition is when wafer is clicked to be tested in the automation or stop the test in the first wafer; the title is "Wafer Summary Report: %wafer id%", the lot/wafer label is "Lot."

- **User Name:** Defined in the Auto-Test Information box in Automation, input parameter, saved in the `.kdf` file.
- **Process:** Defined as Test process level in the Auto-Test Information box in Automation, input parameter, saved in the `.kdf` file.
- **Start time:** The test start time, saved in the `.kdf` file.
- **Device:** Defined as Die Type in the Auto-Test Information box in Automation, input parameter, saved in the `.kdf` file.
- **Sys:** The Computer name, saved in the `.kdf` file.
- **Test name:** Tester name, defined as Equipment ID in the Auto-Test Information box in Automation, input parameter, saved in the `.kdf` file.
- **Test station:** Test station number, default value is 1, saved in the `.kdf` file.
- **Limits:** The limits file name. The limits file used to generate the summary file. If external limits file has been specified, the value is the specified file; if not, a temp file named `temp.klf` is used during the generation. After generation, the temp file is deleted automatically.

Special symbols:

- "?" - If the raw data is outside valid limits, a "?" symbol is added beside the data, and the data is marked with red color.
- "*" - If the raw data is outside spec limits, a "*" symbol is added beside the data, and the data is marked with red color.
- "#" - If the raw data of a critical parameter is outside spec limits, a "#" symbol is added beside the data, and the data is marked with red color. The critical part is defined in the limits editor.
- "-" - If there is no corresponding value to the output parameter in the site, the cell value would be "-"

Total num of wafers/sites: Total number of wafers and total number of sites can be got from the `.kdf` file. See the next figure.

Statistic table: Following the header is a series of column field labels. These fields contain data derived from the `.kdf` file. Some fields, such as %Spec and %Vld, may include an asterisk next to the data points that indicates the data is out of range and gives alerts on the potentially bad data points. The following paragraphs describe these fields that serve as the column indicators for the standard report shown in the above two figures.

- **Name:** The unique test output id. This is the parameter name to which the data is corresponding.
- **Units:** The parameter units, defined in the limits editor. This field is blank if no limits file is specified in the Summary Report panel.
- **Mean:** Mean value of all the data for this parameter. The average for each parameter is calculated from the data of all testing sites. It is the arithmetic mean of all the results
- **Sdev:** SDEV value of all the data for this parameter, depend on the data and the target value defined in the limits editor. This is the standard deviation from the mean of the parameter's data points.
- **Min:** Minimum value of all the data for this parameter.
- **Max:** Maximum value of all the data for this parameter.
- **%sdev:** %SDEV value of all the data for this parameter. It depends on the data and the target value defined in the limits editor. This is the standard deviation expressed as a percentage offset from the mean.
- **SpecL and SpecH:** Spec low value and high value defined in the limits editor. These are the specification constraints from the limits file. These values show quick identification for data points that are out of range. If no limits file is specified, the default value is −1.00e+16 and 1.00e+16, respectively.
- **%spec:** Yield value of the data inside the spec limits, if the value is less than 50, a "*" is appended and marked with red color. This is the percentage of data points that are within SpecL and SpecH.
- **Cnt:** The total records number of this parameter(times this parameter has been tested in the automation).
- **%vld:** Yield value of the data inside the valid limits. if the value is less than 50, a "*" is appended and marked with red color.

Understanding raw data

Raw Data is an expanded part of the Lot Summary Report. The additional data found in Raw Data consists of the actual testing data derived from the `.kdf` file.

Raw Data uses the same header file as the Lot Summary Report. The parameters are listed in columns instead of rows. The information shown in a standard report is listed in the bottom rows on each raw report page. This information is calculated in the same way as in the Lot Summary Report.

In Raw data, separated to different sections with each wafer, array data is in a list. Distinct colors and symbols are used regarding the data value and limits setting.

Wafer statistic table:

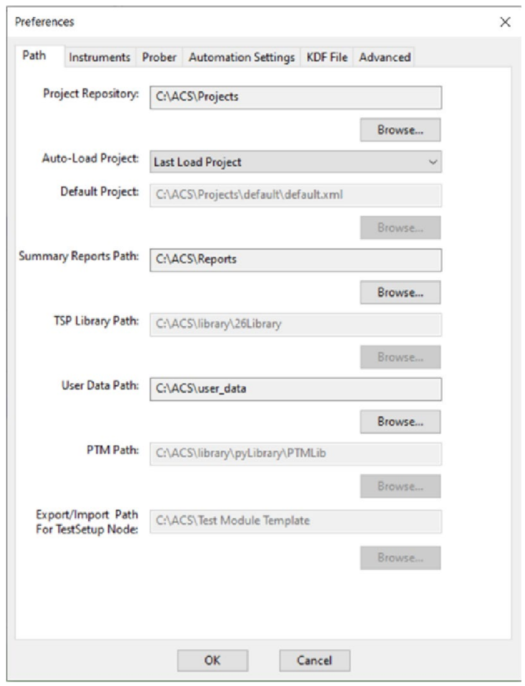
- **Count:** The amount of data inside the valid limits in the current wafer
- **mean/stdev/%stdev/min/max:** Mean/stdev/%stdev/min/max value of data in the current wafer

Customize user data path

In the **Tools** Menu, select **Preferences** to set the binning file and data file path.

This data path dialog allows you to select the directory path for the software parameter data.

Figure 473: User Data Path in Preferences setting



Section 10

Hardware configuration

In this section:

ACS hardware configuration introduction

The Hardware Management Tool is used to manage the hardware configuration of Keithley instruments (refer to Hardware Management Tool (on page 10-4) for more detail). The following instruments are supported:

- Keithley Series 2600A/B SourceMeter® SMU Instruments
- Keithley 2400 Graphical Interface SourceMeter® SMU Instruments
- Keithley 2400 SourceMeter® SMU Instruments
- Keithley Series 2650A High Power System SourceMeter® (includes 2651A and 2657A)
- Keithley 4200A-SCS Parameter Analyzer and supported cards/modules
- Keithley Model DMM7510 7½-Digit Graphical Sampling Multimeter
- Keithley 2010 Series: 7½-Digit Multimeter with Scanning
- Keithley 700 Series Semiconductor Switching Systems
- Keithley Series 3700A System Switch/Multimeter
- Keithley Series 3400 Pulse Pattern Generators
- PTM-only instruments

PTM-only instruments

The categories for PTM-only instruments are:

- Switch Matrix
- Capacitance Meter
- General-Purpose Test Instrument

All supported PTM-only instrumentation and equipment is controlled by Python Test Modules (PTM); however, they are not compatible with STM or ITM.

The software provides Python LPT libraries for each supported external instrument. You can use these libraries to create your own test modules.

The software also provides several standard Python user libraries to control external equipment that is typically used in semiconductor characterization applications. Standard user-module libraries are provided for the equipment (see the following table).

Supported PTM-only table

Category	Instrument	LPT Library[1]	User Library[2]
Switch matrix[3]	Model 707A or 707B Switching Matrix (DDC)	ki70xlpt	switchctrl[4]
	Model 708A or 708B Switching Matrix (DDC)		
Capacitance meter[5]	Model 590 CV Analyzer	ki4200cvulpt	CVITM[6]
	Model 595 CV Analyzer		
General-purpose test instrument[7]	Any IEEE-488 controlled instrument or equipment	created by user	TEKSCOPE[8]
			HiPower_24[9]

[1] LPT library is saved in the path `\\ACS\library\pyLibrary\ptmlpt`. For more about the ACS LPT library functions, refer to Python LPT library.

[2] The User library saved in this path: \\ACS\library\pyLibrary\PTMLib. For more about the PTMLib library, you can refer to the Python User Library. To learn more about how to import a PTM, refer to PTM with the .XRC GUI File (on page 5-163).

[3] ACS supports the Keithley Instruments Model 707A, 707B, 708A, and 708B Switch Matrices. The Model 708A and 708B accept a single matrix card. The Model 70A7 and 707B accept up to six matrix cards. Only one switch matrix can be present in the system configuration at a time.

[4] The switchctrl saved in `\\ACS\library\pyLibrary\PTMLib\ switchctrl`. To learn more about how to use the switchctrl library, refer to Switch_Control.

[5] Up to four supported capacitance meters may be added to system configuration.

[6] The CVITM saved in `\\ACS\library\pyLibrary\PTMLib\CVITM`. To learn more about how to use this library, refer to Configure a capacitor meter library.

[7] ACS supports up to sixteen general-purpose test instruments (GPI). Two-terminal and four-terminal types may be present in the system configuration simultaneously, but the total number of GPI cannot exceed sixteen.

[8] The TEKSCOPE is saved in the `\\ACS\library\pyLibrary\PTMLib\TEKSCOPE`. To learn more about how to use this library, refer to Configure a scope library.

[9] The HiPower_24 is saved in the `\\ACS\library\pyLibrary\PTMLib\HiPower_24`. To learn more about how to use this library, refer to Configure a Series 2400 SourceMeter instruments library.

Communication interface

All instruments are supported with standard GPIB, LAN, USB, and serial port communication connections. They are also available in mixed applications. Additionally, GPIB, LAN, and USB are available for primary communication connections with ITM, PTM, and STM. The default communication for all testing is LAN. RS-232 is also an option.

Using the hardware management tool, these instruments are easily added, configured, and removed from the system configuration.

ACS can also control multiple TSP instruments as a group connected to a single GPIB or ethernet interface. Each group uses TSP-Link® to connect instruments.

The following table indicates how instruments communicate with ACS.

Instrument	Platform	Communication interface
2010/2400	Computer or 4200A	GPIB
2400 Graphical Series SMU		GPIB, ethernet, USB, TSP subordinate
2600		GPIB, TSP subordinate
2600B		GPIB, ethernet, USB, TSP subordinate
4200A		Internal
3700A		GPIB, ethernet, USB, TSP subordinate
DMM7510		GPIB, ethernet, USB
Other PTM-only instruments		GPIB, ethernet, USB

NOTE

If you make changes to the hardware (for example, remove one instrument, change the 707B work mode, change node information), complete the following steps:

1. Close ACS software.
2. Select the **Hardware Management Tool** to scan the hardware.
3. Select **Save** to update the changes.
4. Start ACS.
5. Check the test setup to make sure the new hardware configuration is correct.

Hardware Management Tool

When the software is installed on your computer, the Hardware Management Tool (HMT) is installed at the same time. You see the HMT icon on your desktop (see the following figure) and the software icon. The HMT manages the hardware of your test configuration.

Select the HMT icon to scan your hardware and generate a `Hardware_Config_Online.kcf` (Keithley configuration file).

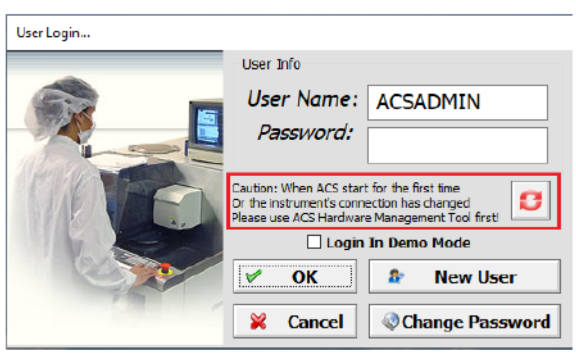
NOTE

Before you use the software, you must scan the hardware of your test configuration. If you do not scan the hardware you will see a warning indicating the hardware configuration file does not exist.

Figure 474: Hardware Management Tool

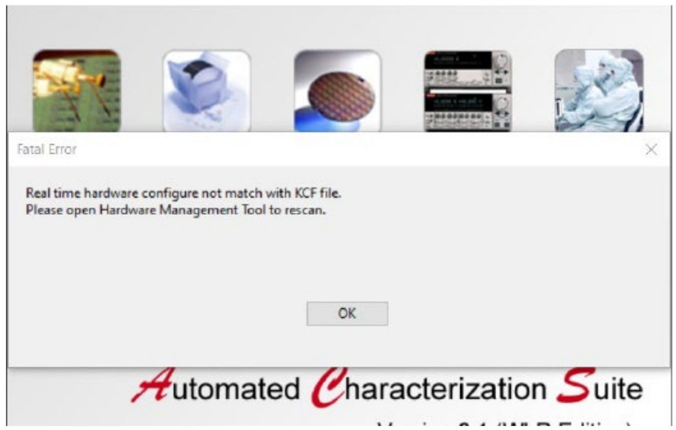
On the User Login interface, you may see a caution message. This may indicate that instrument connections have changed or that you are logging in for the first time to ACS software (see the following figure).

Figure 475: User login



If the hardware has not been scanned with the Hardware Management Tool before starting ACS, you will see the following message "Scan Hardware config file does not exist:," as shown in the following figure.

Figure 476: Hardware configuration fatal error



Scan for instruments

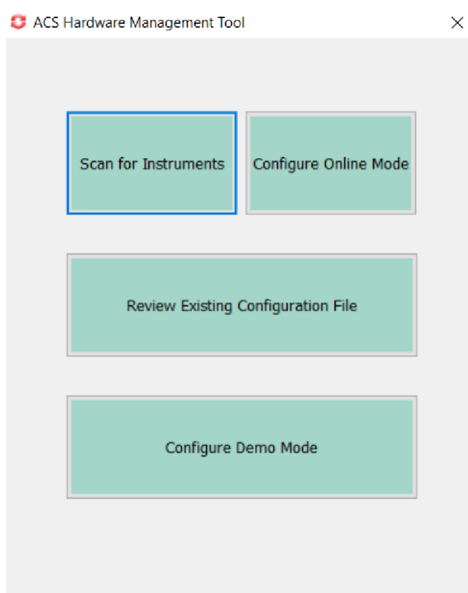
The following describes how to scan the hardware configuration.

When you open the Hardware Management Tool, the interface contains four options. You can scan the connected instruments, configure online mode, review the existing configuration file, or configure a demo mode test.

To scan for instruments:

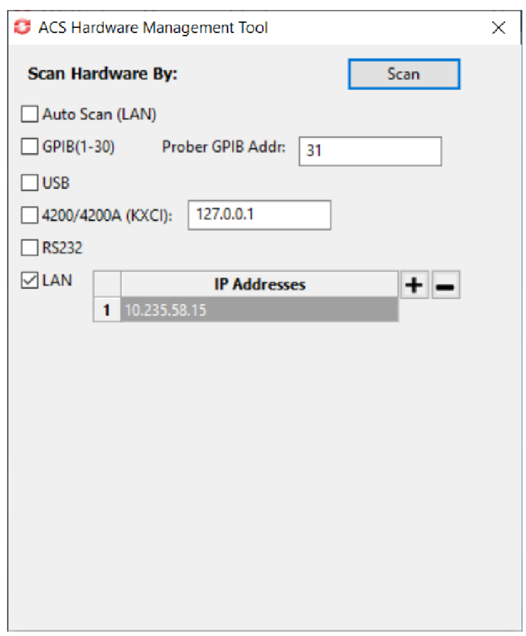
1. Select **Scan for instruments**.

Figure 477: Hardware Management Tool options



2. Select the options for your test based on the instrument connection, as shown in the following figure.
   - Auto Scan (LAN): instruments need to be on the same subnet as the system computer (this means the same first three fields of the IP address)
   - GPIB: instrument address from 1 to 30 is supported
   - Prober GPIB Address: default value is 31
   - USB
   - 4200/4200A (KXCI)
   - IP Address: add or delete an IP address by selecting the plus or minus buttons, and configure the IP address of 4200A-SCS connected LAN
   - RS-232
   - LAN
3. When you have selected your options, select **Scan**, as shown in the following figure. For a list of supported instruments, refer to ACS hardware configuration introduction (on page 10-1).

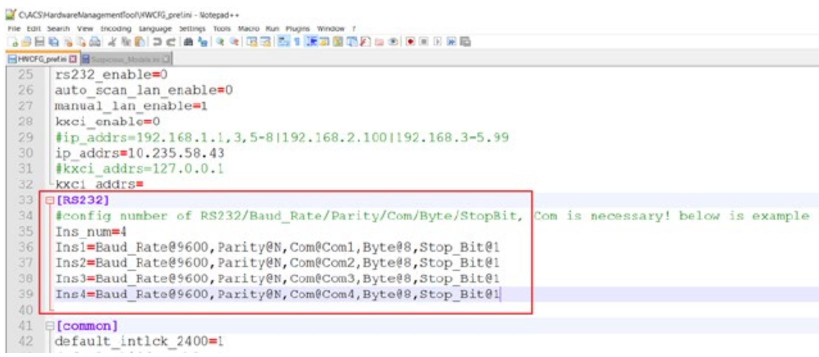
Figure 478: Hardware Management Tool



If you are using a RS-232 connection, the instrument is not automatically added to the hardware configuration. You must add instruments connected with RS-232 manually.

Change the hardware configuration file that is in the following directory on your computer: `C:\ACS \HardwareManagementTool\HWCFG_pref.ini`. In this file, you need to change the Baud rate, parity, byte, and stopBit settings. Review the following figure for details.

Figure 479: RS-232 settings



The system queries the instruments and the updated configuration information is shown in the comparison results. See the Comparison Results (Online Mode) (on page 10-8) for detail.

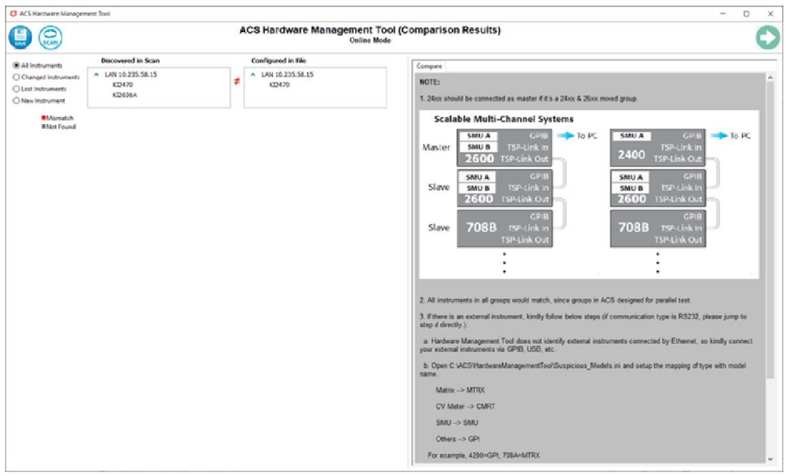
Comparison Results (Online Mode)

NOTE

Before you can compare results, you must scan the instruments connected to your system for testing. Refer to Scan for instruments (on page 10-6) for more detail.

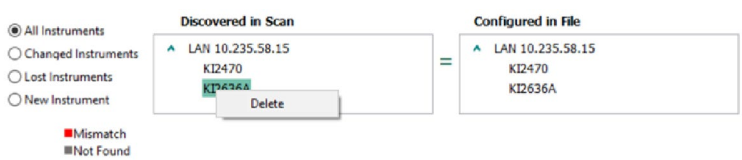
The comparison results, as shown in the following figure, shows the instruments connected to your system from the scan. See "Discovered in Scan" and "Configured in File" in the following figure. The scan compares the present online instrument configuration to the instruments that were previously configured. You can choose to apply the newly scanned instruments and add to the online configuration file by selecting **Save**.

Figure 480: Comparison results



You can delete instruments that are found during the scan, as shown in the following figure. Right-click to delete. Select **Save** to keep your configuration.

Figure 481: Delete in Online ode



When you save your configuration, the Configuration Page (Online Mode) opens. See the Configuration Page (Online Mode) (on page 10-10) for detail.

Configuration Page (Online Mode)

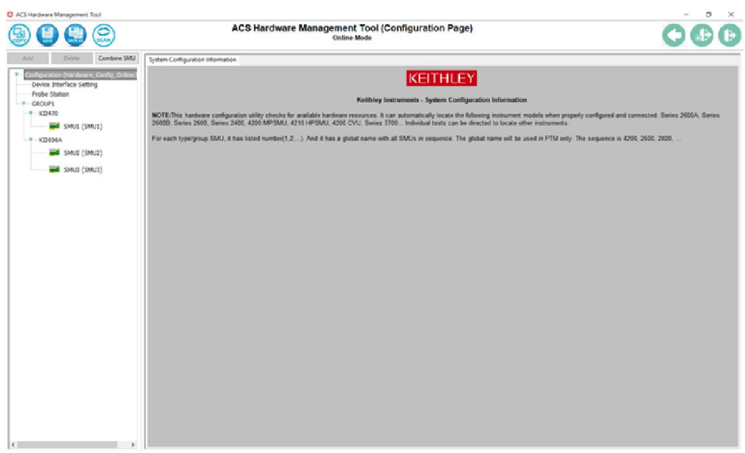
The configuration page shows the updated instrument configuration, as shown in the following figure. You can add an instrument in the configuration page by selecting **Add**.

Make sure you have the instrument connected to the test system to add it. Select an instrument, then select **Delete** to remove an instrument.

Also, to combine multiple SMU instruments to your test configuration, select the SMU instruments and select the **Combine SMU** option.

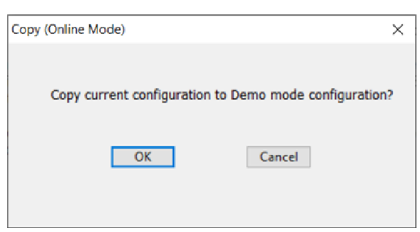
You also have the option to go back to the comparison results window, save and exit, or close the window using the arrows on the toolbar.

Figure 482: ACS Configuration Page (Online Mode)



Select copy in the toolbar to save your online configuration to the online mode configuration. Select **OK** to save the configuration, as shown in the following figure.

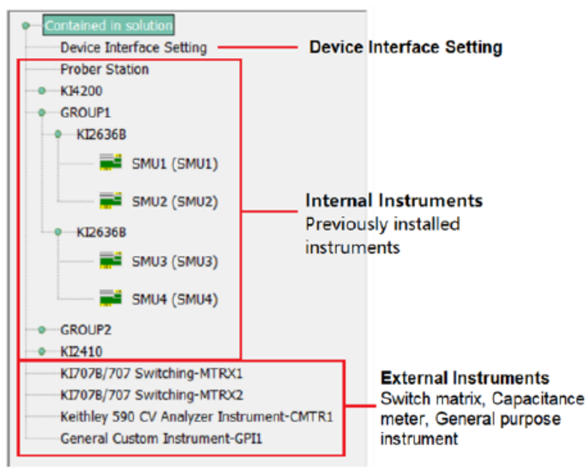
Figure 483: ACS Copy (Online Mode)



Configuration navigator

The configuration navigator displays a tree view that shows each component present in the system configuration. Selecting a component, or node, in the configuration navigator displays the properties associated with the selected component in the work area. The following figure shows a typical system configuration with the configuration navigator partially expanded.

Figure 484: Configuration navigator view of typical system

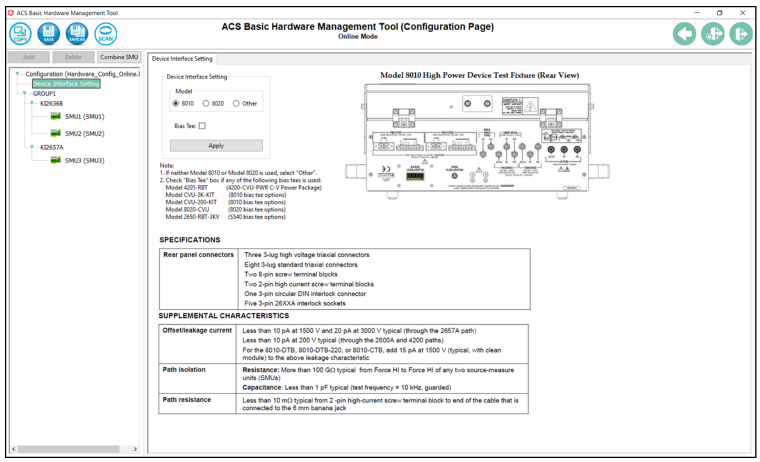


See the Configuration navigator introduction (on page 5-1) for more detail.

Device interface setting

The Hardware Management Tool uses the device interface setting to manage the hardware configuration to optimize the default instrument settings (see the following figure). Select the Device Interface Setting, as shown in the following graphic, to see the device settings.

Figure 485: Device Interface Setting



In the Device Interface Setting GUI, you must choose the Model 8010 High Power Test Fixture, Model 8020 High Power Interface Panel, or Other device with or without a bias tee. See the table below for configurations that also include default settings.

To apply the device interface setting:

1. Select the model in your device interface. If you have a Model 8010 or Model 8020, select accordingly, or select Other.
2. Select the bias tee if you are using any of the following bias tees: Model CVU-3K-Kit, Model CVU-200-KIT, Model 8020-CVU, or Model 4205-RBT.

NOTE

To use the bias tees with the CV instrument in the Model 4200A-SCS and access the high voltage capacitance measurement modules, ACS software must be installed on the Model 4200A (there is no remote support). If you do not need a bias tee and want to use the Model 4210-CVU alone, you should use the CVITM test module. The CVITM test module requires the Model 4200A-SCS KXCI software which can be used remotely if ACS software is installed on the remote computer.

3. Select **Apply**.

NOTE

If you have a bias tee in your configuration, you must select the bias tee option in the device interface setting to access the appropriate CVU connection compensation. For more information, refer to CVU connection compensation (on page 5-189).

The table below details how the device interface setting changes the default instrument settings:

Model	With bias tee option?	Delay factor	Contact check	Measurement speed
8010	No	1	On	Normal
8010	Yes (applies to CVU-3K-KIT, CVU-200-KIT, or 4205-RBT)	2	On	Slow
8020	No	1	Off	Normal
8020	Yes (applies to 8020-CVU bias tee)	2	Off	Slow
Other	No	1	On	Normal
Other	Yes (applies to CVU-3K-KIT, CVU-200-KIT, or 4205-RBT)	2	On	Slow

The delay factor is a multiplier to the Auto Delay and is applied when Auto Delay is enabled. Auto Delay is an instrument-specific delay associated with the SMU measurement range. Auto Delay and Contact Check can be enabled in the Instrument tab settings (on page 7-22).

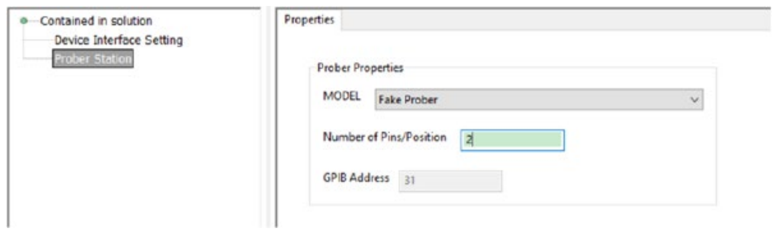
If you make changes in this tab after configuring device interface setting, you will override values presented in the above table. Measurement speed is controlled in the Timing window of ITMs. Refer to Model 42xx SMU ITM (on page 5-93) for more details.

If you make changes to timing after configuring device interface setting, you will override the measurement speed settings in the above table.

Prober station

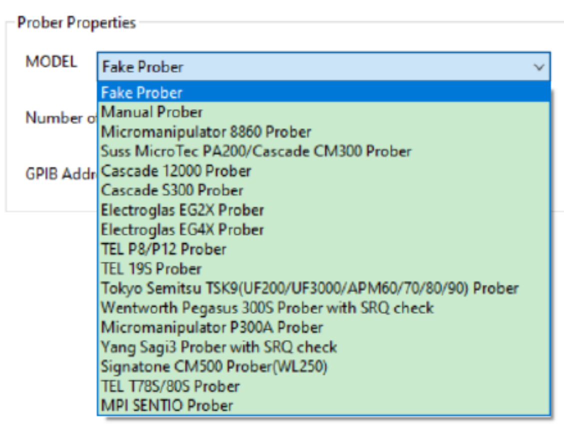
Select prober station in the configuration navigator and the prober properties are available, as shown in the following figure. See the Configuration navigator introduction (on page 5-1) for more detail.

Figure 486: Prober Properties



- Model contains a list where you can choose a prober to accomplish your test, as shown in the following figure.
- Based on the prober you choose, choose the pins or the position of the prober to accomplish your test.
- GPIB address is fixed at 31.

Figure 487: Prober model list



Open ACS and confirm hardware

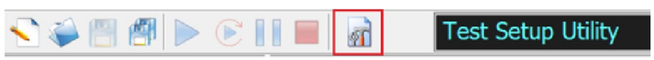
The Hardware Management Tool scans information for the instruments using ACS and modifies configurable properties.

NOTE

Before you use ACS software, you must scan the hardware of your test configuration. If you do not scan the hardware, you see a warning indicating the hardware configuration file does not exist. Refer to Hardware Management Tool (on page 10-4) for more detail.

Select the Configure Hardware icon on the toolbar.

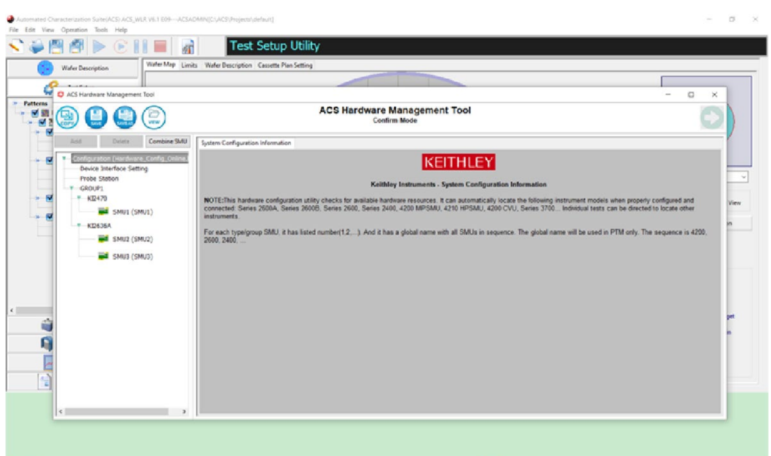
Figure 488: Start hardware configuration



The hardware management dialog opens, shown in the following figure. The interface of the hardware configuration consists of:

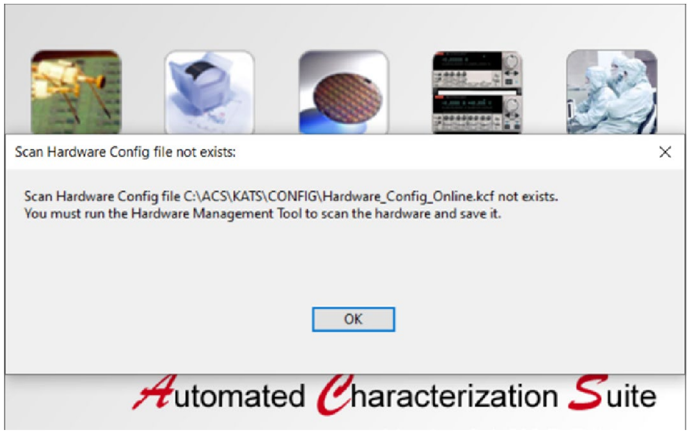
- **The configuration navigator:** The left panel, which displays the instruments and equipment that are included in the ACS system configuration.
- **The work area:** The right panel, where the instrument and equipment properties are displayed.

Figure 489: Hardware Management Tool interface



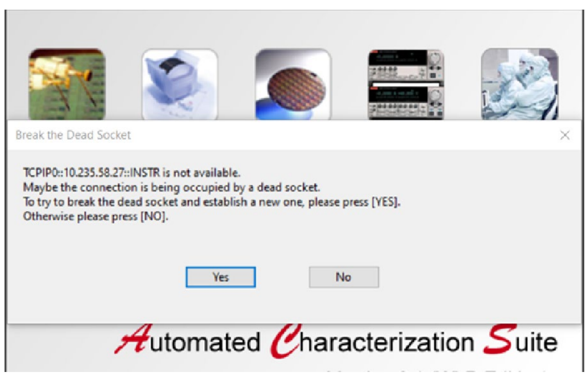
If you encounter an error message when opening ACS, the hardware configuration may have changed. See the following figure for an example. You need to run the Hardware Management Tool (on page 10-4) to establish the hardware configuration.

Figure 490: Hardware Configuration File does not exist



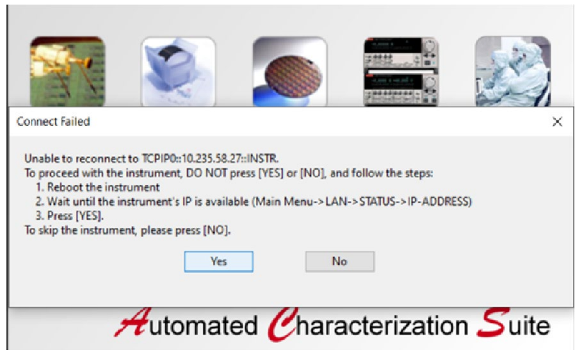
It is possible you may encounter a TCPIP error, as shown in the following figure. You can select **Yes** to continue to confirm your hardware, or select **No** to stop.

Figure 491: Break the dead socket



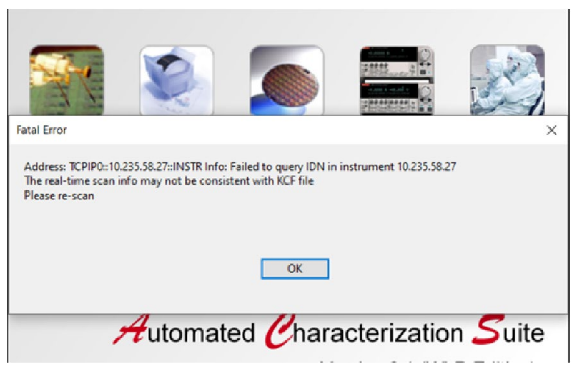
If you select **Yes** and the software cannot connect to the instrument, you are notified, as shown in the following figure.

Figure 492: Connection failed



If the TCPIP fatal error continues, refer to the following figure that indicates you should scan the instruments using the Hardware Management Tool (on page 10-4).

Figure 493: Fatal error continues

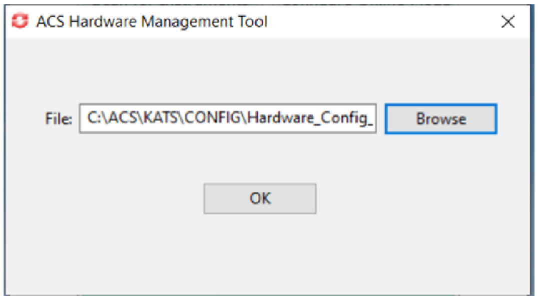


Configuration Page (Confirm Mode)

Open the Hardware Management Tool interface to access the existing configuration file. Choose **View** to select **Confirm Mode**. In confirm mode, you can select a `.kcf` file and view the instruments configuration saved in the file, as shown in the next figure.

- Select **Browse** to select the `.kcf` file.
- Select **OK** to enter the confirm mode page.

Figure 494: Review existing configuration file



The Configuration Page shows the updated instruments configuration.

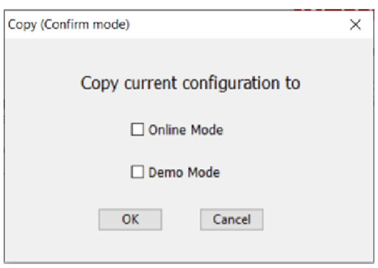
The left panel is the configuration navigator where all the instruments and equipment that are included in the ACS system configuration are displayed. The right panel is the Work Area where all the instruments and equipment properties are displayed.

- Select **Add** to open the external instrument window.
- Select **Delete** to delete an external instrument.

In the external instrument window, you can select and configure external instruments to add to the configuration, including Switch Matrix, general-purpose instrument, and a capacitance meter, (for details refer to Add and delete a PTM-only instrument (on page 10-23)).

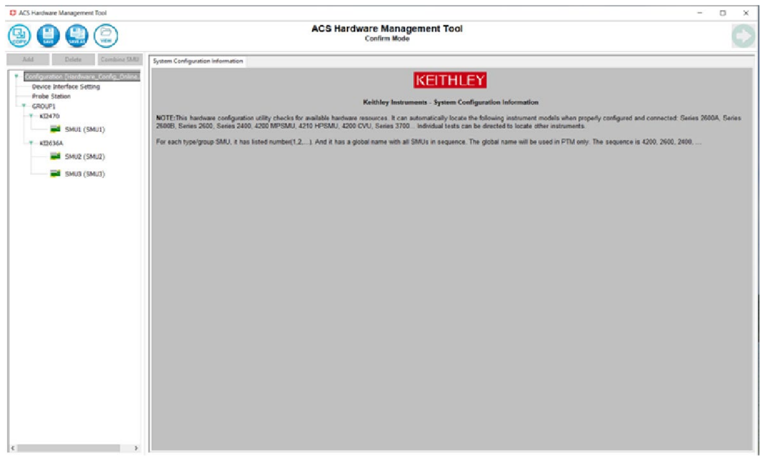
- Select **COPY** to copy the current configuration, either online or demo mode.

Figure 495: Configuration Page (Confirm Mode)



- Select **SAVE** to save the instrument configuration to the demo configuration file.
- Select **SAVE AS** to save the instruments configuration to a selected `.kcf` file.
- Select **VIEW** to review the existing configuration file.

Figure 496: Confirm mode



Configure Demo mode

Demo mode allows you to manage and simulate a virtual hardware system without having actual instruments connected to the system computer.

You can configure the instruments for the demo system in the Hardware Config (Demo) panel by selecting the Parametric Curve Tracer Demo System or the Customized Demo System, as shown in the following figure.

Select the menu to choose an instrument:

- 2600-PCT-1B (200 V / 10 A)
- 2600-PCT-2B (200 V / 50 A)
- 2600-PCT-2B (+50 A , 200 V / 100 A)
- 2600-PCT-3B (3 KV / 10 A)
- 2600-PCT-3B (3 KV / 50 A)
- 2600-PCT-4B (+50 A, 3 KV / 100 A)
- With PCT-CVU: Select PCT-CVU

NOTE

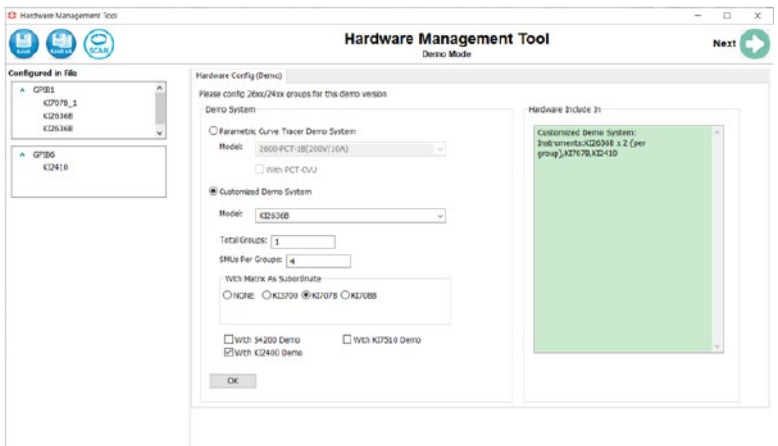
The Series 3400 cannot be simulated in the scan in demo mode.

For the Customized Demo System, you can configure customized instruments for a demo system. Select the model, total group, and SMUs per group. The Configured in File indicates the instruments in the demo system, as shown in the following figure.

Make sure you save the demo configuration file. You can also create a `.kcf` file by selecting **Save As**.

You can add a 4200A, 7510, or 2400 instrument to the demo. The Configured in File indicates the instruments in the demo system, as shown in the following figure.

Figure 497: Hardware Management Tool (Demo Mode)



NOTE

You can also simulate the Series 3700A, Model 707B, 708B instruments, or a 4200A-SCS instrument.

To simulate a scan for the Series 2600B:

1. Select the **Customized Demo System** option and select your 2600B model from the Model list.
2. Input the Total Groups and SMUs per Group.
3. Select **OK**.

The 2600B information is displayed in the configuration navigator with corresponding properties information displayed in the work area.

To simulate a scan of the Series 3700A and Model 707B or 708B instruments:

1. Select the **Customized Demo System** option and select your specific **With Matrix as Subordinate** instrument. For more information about the 3700A, 707B, or 708B used as a subordinate or child, refer to TSP-Link connections (on page 11-9) and Assign a node number (on page 11-13) for more detail.
2. Select **OK**.

Depending on your selection, the 3700A, 707B, or 708B information is displayed in the configuration navigator with corresponding properties information displayed in the work area.

To simulate a scan of the Model 4200A Parametric Analyzer or 4200-SCS:

1. Select the **Customized Demo System** option and select **With S4200 Demo**.
2. Select **OK**.

Configuration Page (Demo Mode)

The Configuration Page shows the updated instruments configuration. It has two parts.

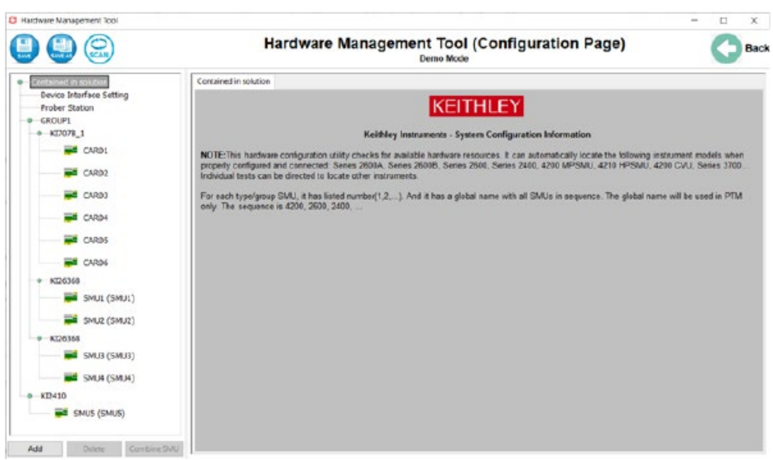
The left panel is the configuration navigator. All the instruments and equipment are included in the ACS system configuration are displayed here. See the Configuration navigator introduction (on page 5-1) for more detail. The right panel is the Work Area. The instruments and equipment properties are viewed and modified here.

- Select **Add** to open the external instrument window.
- Select **Delete** to delete an external instrument.

In the external instrument window, you can select and configure external instruments to add to the configuration, including Switch Matrix, general-purpose instrument, Capacitance Meter.

- Select **SAVE** to save the instrument configuration to the demo configuration file.
- Select **SAVE AS** to save the instruments configuration to a selected `.kcf` file.
- Select **SCAN** to rescan the hardware.
- Select **Back** to go back to Hardware Config (Demo) view.

Figure 498: Configuration Page (Demo Mode)



Choose **View** to select **Confirm Mode**. In Confirm Mode, you can select a `.kcf` file and view the instruments configuration saved in the file. You can also add external instruments into the configuration and save the configuration to a `.kcf` file.

1. Select **File** to browse to the `.kcf` file.
2. Select **OK** to enter the Confirm Mode page.

The configuration page shows the updated instruments configuration.

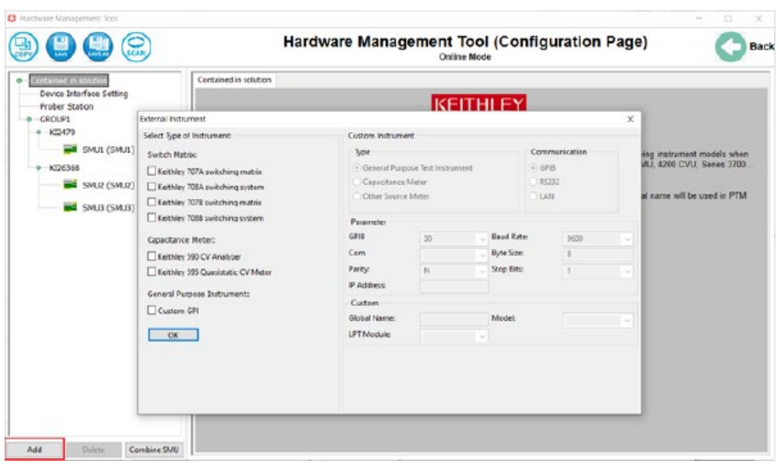
Add and delete a PTM-only instrument

A PTM-only instrument can be manually added to your hardware configuration through the Hardware Management Tool demo mode. Refer to PTM-only instruments (on page 10-2) for more detail.

To add a PTM-only instrument:

1. In the configuration navigator, select **Add** to open the external instrument dialog, as shown in the following figure.

Figure 499: Add or delete external instruments



2. In the PTM-only Instrument dialog, select the type of instrument to add.
3. Configure the parameters for a general-purpose instrument in the Custom section.

To delete a PTM-only instrument:

1. Select the PTM-only instrument.
2. Select **Delete**.
3. Select **Yes** to confirm this action.

The following table lists the remaining types of instruments you can add.

Switch matrix	
Model 707A/707B switching matrix (DDC)	Model 708A/708B switching system (DDC)
Capacitance meter	
Model 590 CV Analyzer	Model 595 Quasistatic CV Meter
General-purpose instrument	
Custom GPI	

To add a switch matrix, capacitance meter, or general-purpose instrument:

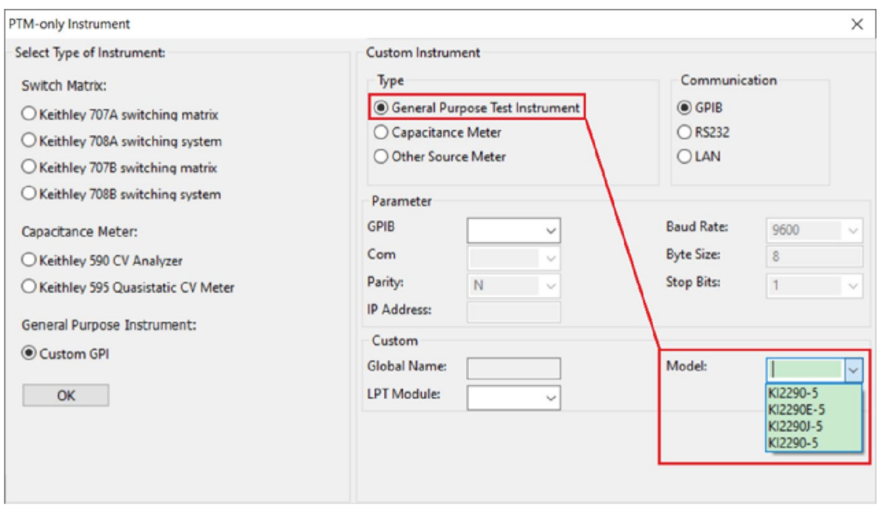
1. Select **Add** in the configuration navigator to open the external instrument dialog, as shown in the following figure.
2. Select the PTM-only instrument that you want to add.

When you choose the custom GPI, you need to set the type of General Purpose Test Instrument and the parameters, as shown in the following figure. For more information about the Keithley Instruments Model 2290 Power Supply, refer to Power supply library.

NOTE

The LPT Model name must be the same as the Python file name saved in the `ptmlpt` folder (`\\ACS\library\pyLibrary\ptmlpt`). ACS supplies these LPT library packages: `ki23xlpt`, `ki24xxlpt`, `ki26xxlpt`, `ki37xxlpt`, `ki42cvulpt`, `ki42xxlpt`, `ki70xlpt`, `kiag4284lpt`, and `ki20xxlpt`.

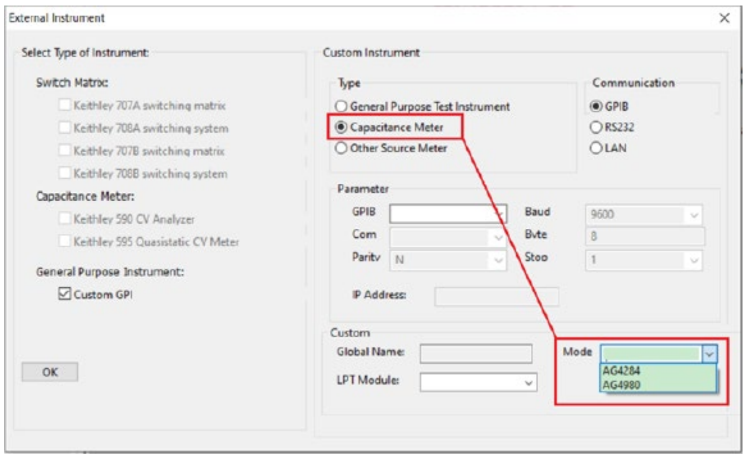
Figure 500: General Purpose Test Instruments



When you choose capacitance meter type, you have two options:

- AG4284 (Keysight)
- AG4980 (Keysight)

Figure 501: Capacitance Meter Instruments



When you choose Other Source Meter, you have the options:

- KI236
- KI237
- KI238

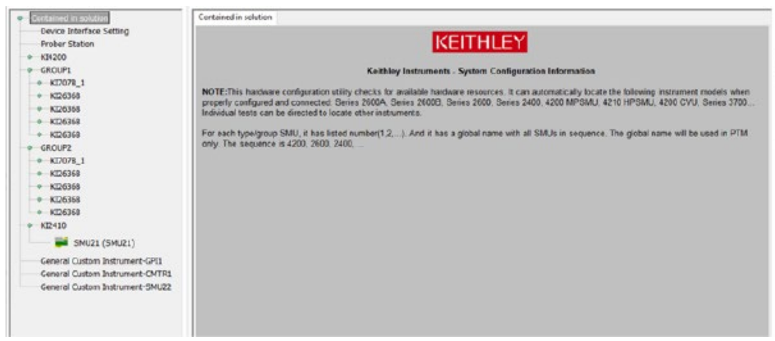
Instrument system configuration

Each instrument that is included in the system configuration appears in the configuration navigator. The following sections describe the properties for each supported instrument.

KI system configuration

When you select the Keithley Instruments System Configuration in the configuration navigator, the work area displays the system configuration.

Figure 502: KI System Configuration



Series 2600B system properties

Single configuration: When the node of a Series 2600B instrument on the configuration navigator is selected, the work area displays an overview of the instrument.

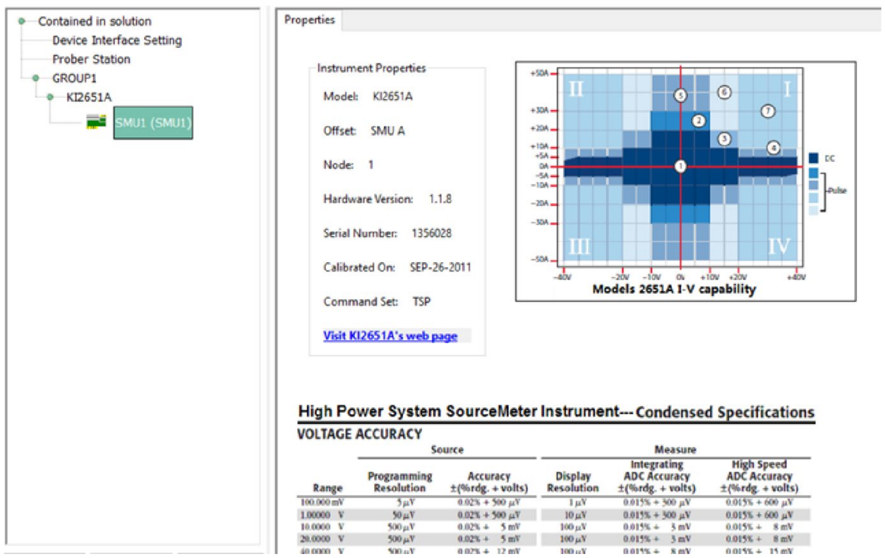
Group configuration: When the node of GROUPx on the configuration navigator is selected, the work area displays the groups' GPIB address and configuration.

SMU configuration: When the node of a Series 2600B SMUx on the configuration navigator is selected, the work area shows the SMU properties, specifications, and power capacity.

Example: Model 2651A

The following figure shows the instrument properties. You see the model number, node number, version of the hardware, serial number, the calibration date, and specifications for the SourceMeter. You can use this information to configure your testing and create a schedule to make sure calibration is completed as needed.

Figure 503: Model 2651A information



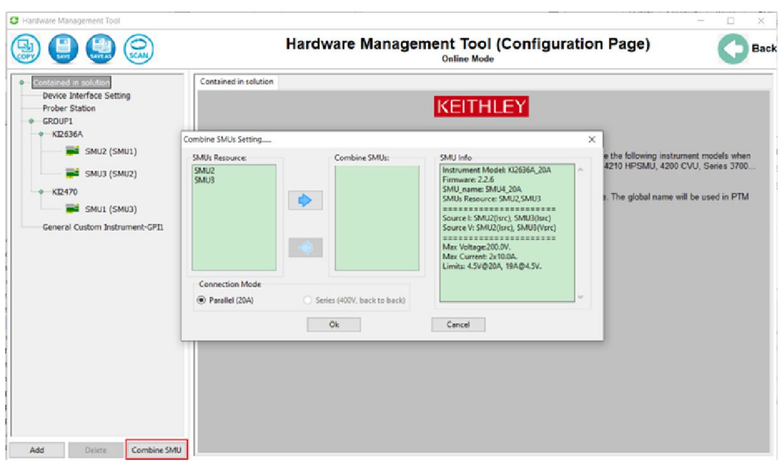
Composite SMU

ACS supports combining two 2651A, 2636A/B, 2612A/B, or 2602A/B SMUs as a single SMU. Two 2651A SMUs can be combined in parallel as a single SMU. This will give you the capability to output 100 A, or use in series for an 80 V output. Combining two 2636A/B or 2612A/B or 2602A/B SMUs in parallel can extend the operating range to 20 A pulse, or use in series to extend the operating range to 400 V pulse. In the configuration navigator, the combined 2651A SMUs are considered a composite SMU. The work area will show the properties for the composite SMU.

Select **Combine SMUs**. A dialog opens where you choose the SMUs to combine (only two 2651A, 2636A/B, 2612A/B, or 2602A/B SMUs can be combined) and choose the connection mode (see the following figure).

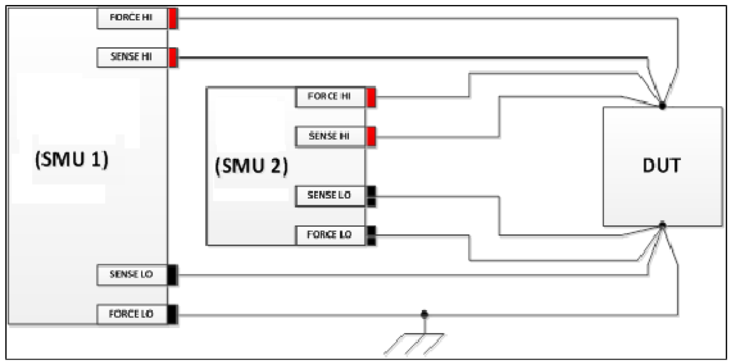
- Parallel (100 A or 20 A): Two SMUs are in parallel for higher current output. The composite SMU for the 2651A is named SMUx_100A (x represents the SMU number). The composite SMU for the 2636A/B, 2612A/B, or 2602A/B is named SMUx_20A.

Figure 504: Combine two SMUs



- Series (80 V or 400 V): Two SMUs in series back-to-back for a higher voltage output. The composite SMU for the 2651A has two terminals, SMUx_80V_HI and SMUx_80V_LO (x represents the SMU number). The composite SMU for the 2636A/B, 2612A/B, or 2602A/B has two terminals, SMUx_400V_HI and SMUx_400V_LO.

Figure 505: Parallel composite SMU connections

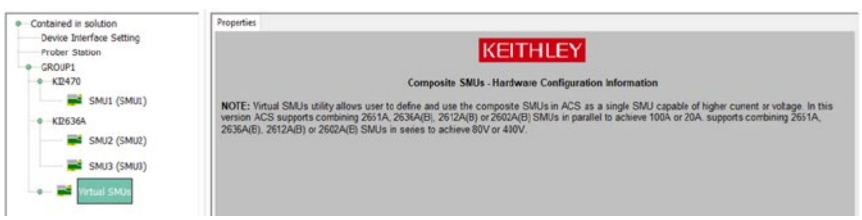


NOTE

The parallel SMU can only be used in an ITM.

In the configuration navigator, the composite SMU is labeled "Virtual SMUs" (see the following figure). The properties of the composite SMU are displayed in the work area.

Figure 506: Virtual SMUs



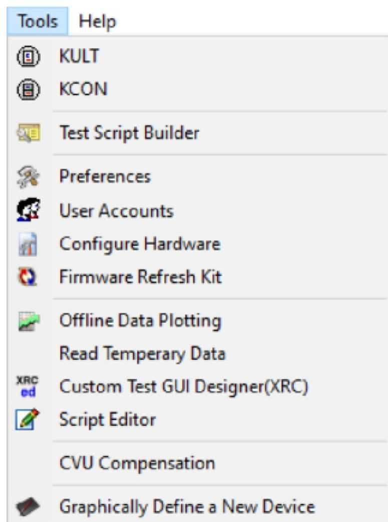
Firmware Refresh Kit option

In the Tools menu, the Firmware Refresh Kit option allows you to update the firmware for Series 2600 and 2600A instruments.

NOTE

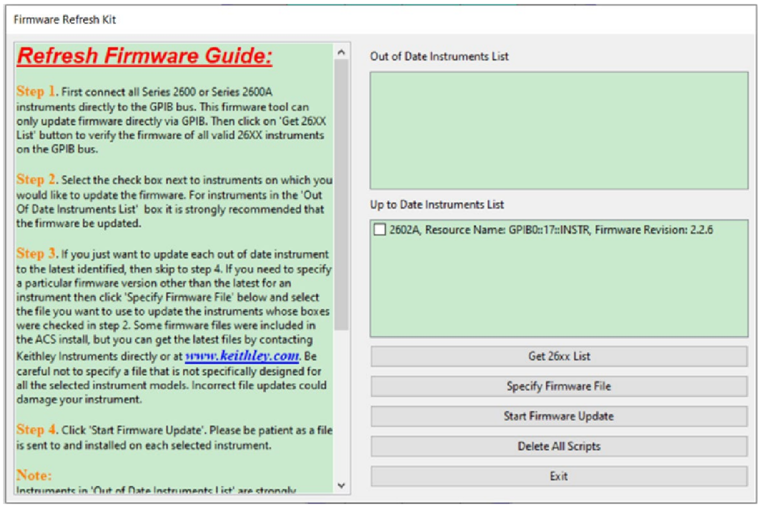
This firmware tool updates instruments connected to GPIB only. Do not to use a firmware file that is not specifically designed for the selected instrument.

Figure 507: Firmware Refresh Kit



The Firmware Refresh Kit dialog is shown in the following figure. You can use Get 26xx List, Specify Firmware File, Start Firmware Update, Delete All Scripts, and Exit to perform the firmware options.

Figure 508: Firmware Refresh Kit dialog



If you want to update or change your firmware on instruments you have, see the following list of options:

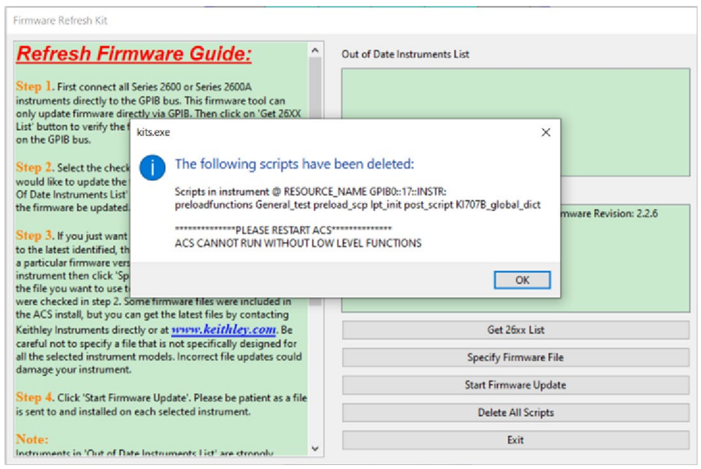
- Connect all Series 2600 and 2600A instruments directly to a GPIB cable. Select the **Get 26xx List** option to verify the firmware for the 2600 and 2600A instruments.
- For instruments in the Out-Of-Date Instruments List area, it is strongly recommended that firmware is updated. The software automatically applies the firmware update with this option.

CAUTION

For the next option, make sure you select the correct firmware file when updating firmware. Incorrect firmware files may damage your instrument. For example, only install 2600 firmware files on 2600 instruments, and 2600A files on 2600A instruments. ACS software includes firmware files, however, you can find the latest firmware files on tek.com/keithley (it is recommended you search for 2600A firmware options or 2600B firmware options, depending on your needs).

- If you want to update an instrument to a specific firmware version, other than the latest for an instrument, select **Specify Firmware File** and select the file you want to use.
- Select **Start Firmware Update**. The firmware update time will vary depending on several factors, such as the number of instruments connected.
- If you want to delete all scripts for the instruments in the out-of-date instruments list and the up-to-date instruments list, select **Delete All Scripts** (see the following figure).

Figure 509: Delete scripts



Select **Exit** to close the Firmware Refresh Kit dialog.

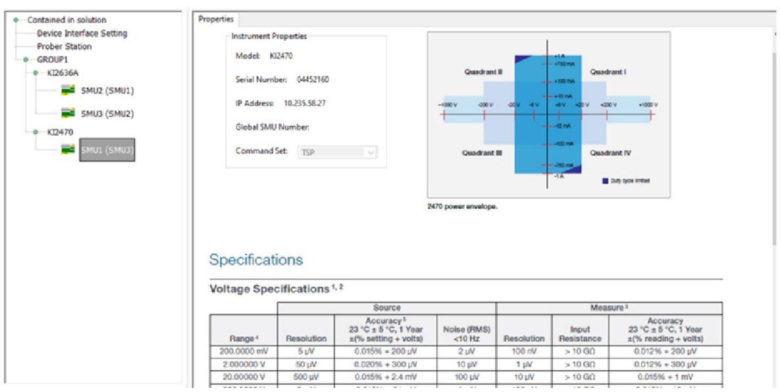
Series 2400 system properties

When the node of a Series 2400 instrument on the configuration navigator is selected, the work area shows the SMU properties, specifications, and power capacity.

The instrument Properties area displays the following information, as shown in the following figure.

- SMU Model
- SMU Serial Number
- GPIB Address
- Instrument name

Figure 510: SMU configuration information



Note that you have the option to delete scripts in a Series 2400 instrument.

To remove scripts:

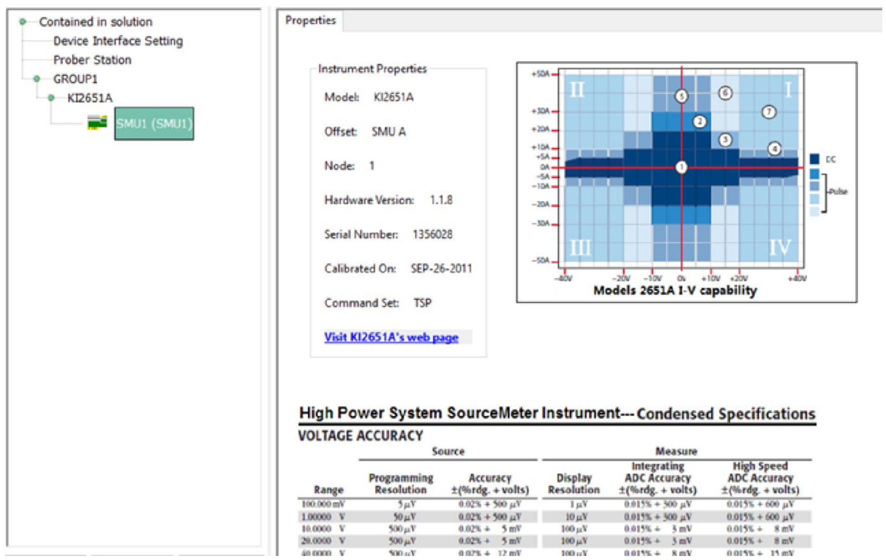
1. Press the menu button on the 2400 instrument.
2. Go to **Scripts > Manage**.
3. Select the scripts to delete.

For more information, refer to the Series 2400 documentation on tek.com/keithley.

Series 3400 system properties

When the node of a Series 3400 PGU instrument on the configuration navigator is selected, the work area displays the Instrument Properties and Matrix Connection.

Figure 511: Model 2651A information



The model of the PGU is displayed in the Instrument Properties text box and it also contains the GPIB address and the serial number of the instrument. The Matrix Connection shows the connection between the channel terminal and the matrix terminal.

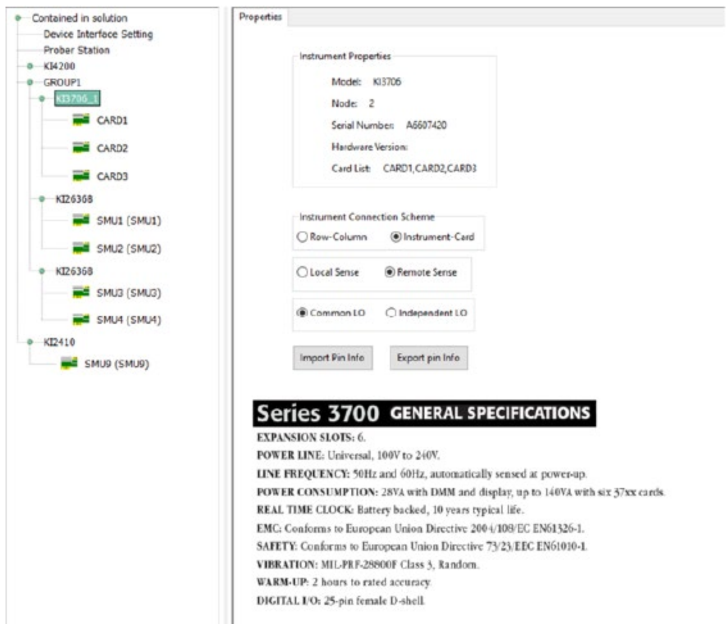
NOTE

When the instrument is connected through a matrix, the Connection column will display a column number. Otherwise, the Connection column will display NC when not connected through a matrix.

Series 3700A system properties

When the node of a Model 3706A on the configuration navigator is selected, the work area shows the Properties, Instrument Connection Scheme, and specifications.

Figure 512: Model 3706A information



A Model 3706 Properties tab contains the following:

- **Instrument Properties area:** Displays the node and the two type of cards.
- **Instrument Connection Scheme area:** Allows you to specify the instrument connection scheme:
  - Row-column or instrument-card.
  - Instrument SENSE terminals (for example, local or remote sense for measurements).

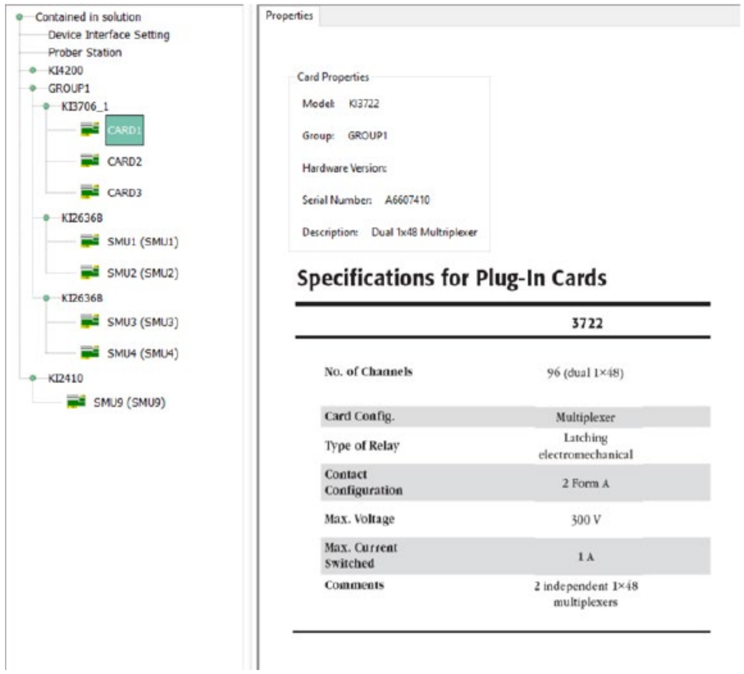
The following Instrument Connection Scheme selections define the scheme for interconnections between the instruments, the switch-matrix rows and columns, and the test system (prober or test fixture).

- Row-Column (instruments to rows and prober/test fixture to columns) or Instrument Card (both instrument and prober/test fixture to columns).
- Local Sense (connections only to instrument FORCE terminals) or Remote Sense (connections both to instrument FORCE and SENSE terminals).
- Series 3700A General Specifications area: displays the information about the specifications.

Series 3700: multiplexer card

The next figure shows two cards are inserted. They are multiplexer card (Card1) and Switch card (Card2). When the node of CARD1 is selected, the work area shows properties and specifications.

Figure 513: Multiplexer card information



As for Card1, the Card Properties area contains information about Model, Group, Hardware Version, Serial Number, and Description. Specifications for cards are also listed.

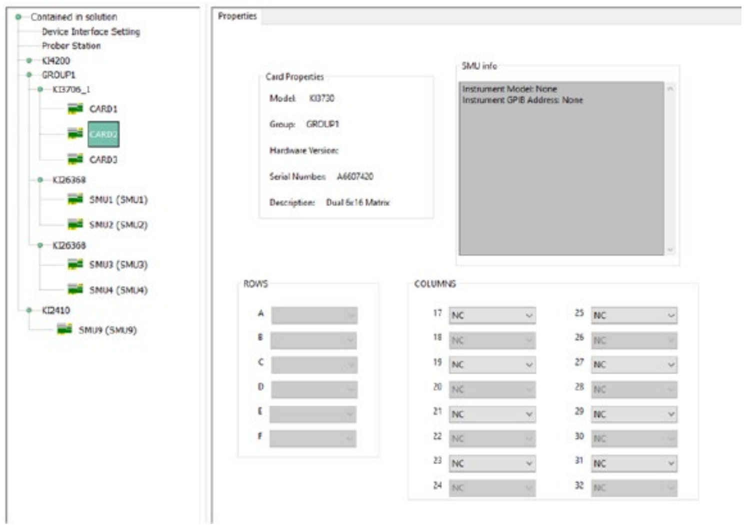
NOTE

If any of the matrices in the system are set to Remote Sense, all the Series 2600B and Series 2400 instruments in the system are set to four-wire (4W) mode when executing a test. Also, the SMU Remote Sense option box in the Preferences settings Instrument tab is disabled. When all the matrices are set to Local Sense, all the Series 2600B and Series 2400 instruments in the system are set to two-wire (2W) mode when executing a test. It is highly recommended that all the matrices in the system are in the same mode. For more information on the Preference settings, refer to Automation (on page 7-1).

Series 3700: matrix card

When the node of CARD2 is selected, the work area shows the Properties and connection options that may be defined for the device.

Figure 514: Matrix card information



The Card Properties area for the matrix card is different from that of the multiplexer card in that there are two additional areas that provide for the definition of the connection method:

- Rows area
- Columns area

In the Rows area, the combo boxes labeled 1 through 6 correspond to the 6 rows of all Model 3706 compatible matrix cards. Use the menus of the combo boxes to connect the rows to the instrument terminals.

NOTE

Prober or test-fixture pins are always connected to matrix-card columns.

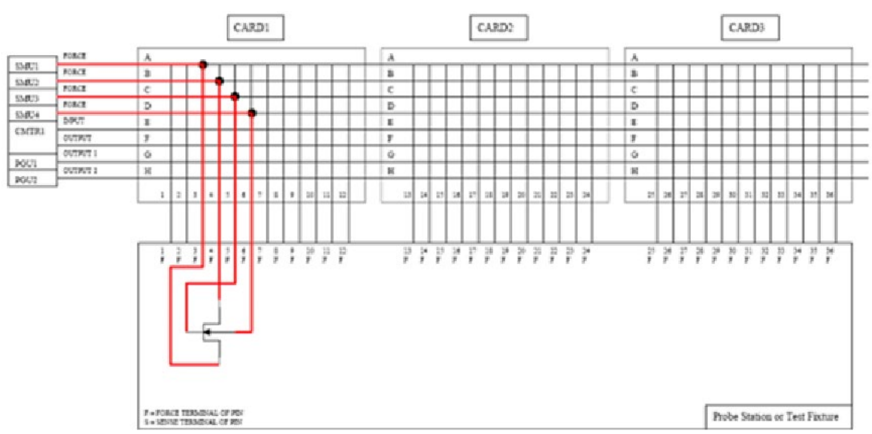
In the Columns area, the combo boxes labeled 1 through 16 (or 17 through 32 for this example) correspond to the 16 columns of all Model 3706A compatible matrix cards. Use the menus of the combo boxes to connect the columns to the instrument terminals and prober or test-fixture pins.

NOTE

You may connect instrument terminals to the matrix card columns only when the Instrument Connection Scheme setting is active.

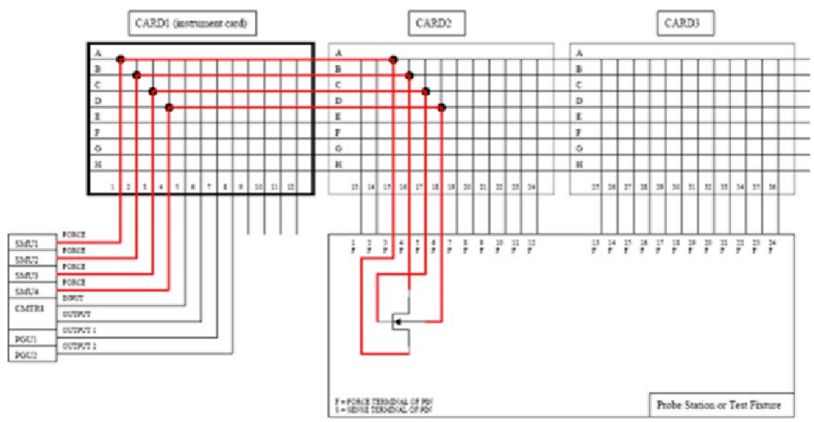
- As for the work area of Card2, you can choose the items in the menu in Rows and Columns area. There are different connections for Card2.
- Row-Column: With the Row-Column scheme, instruments are connected to switch matrix rows, and prober/test fixture pins are connected to switch-matrix columns. This connection scheme is the simplest. Instrument signals can route to prober/test-fixture pins through only one matrix card. However, the Row-Column scheme limits the number of external instruments. If you need to connect numerous external instruments to the prober/test-fixture, use the Instrument Card scheme.

Figure 515: Row-column, Local Sense connection scheme example



- Instrument Card: With the Instrument Card scheme, both instruments and prober/test fixture pins are connected to the columns of the switch matrix. Instrument signals route to the prober/test-fixture pins through two or more matrix cards, as illustrated in the next figure. This connection scheme can support large systems with numerous instruments, by removing the 8-row instrument-connection limitation.

Figure 516: Instrument card, Local Sense connection scheme example

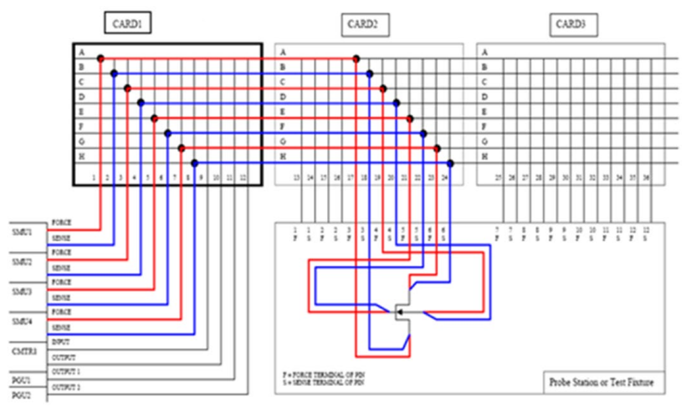


- Local Sense: Use Local Sense when the measurement-pathway resistance is small and the associated voltage errors are negligible. The measurement pathway is comprised of the following conductors, connected in series:
  a. The cables used to connect the instruments to the matrix
  b. The internal matrix-card signal path
  c. The cables used to connect the matrix to the prober or test fixture

Current flowing through the measurement pathway creates a voltage drop (an error voltage) that is directly proportional to the pathway resistance. This error voltage is present in all Local Sense voltage measurements.

- Remote Sense: Use Remote Sense to eliminate the effects of measurement pathway resistance. The next figure illustrates the use of Remote Sense in an instrument card configuration. Note that Remote Sense requires twice as many measurement pathways. The FORCE pathways (in red) are the current-carrying pathways and the SENSE pathways (in blue) are the measurement pathways.

Figure 517: Instrument card, Remote Sense connection scheme example



Models 707A/707B and 708A/708B switch matrices

Models 707A, 707B, 708A, and 708B switch matrices are added to ACS hardware configurations by using the add a PTM-only instrument through a GPIB connection. Models 707B and 708B are automatically scanned and added to the configuration through the GPIB and TSP-Link® as a master or subordinate.

ACS supports multiple matrices, such as Models 707A, 707B, 708A, 708B, and Series 3700A instruments to the hardware configuration. Additionally, ACS supports multiple matrices as a group, or as individual instruments.

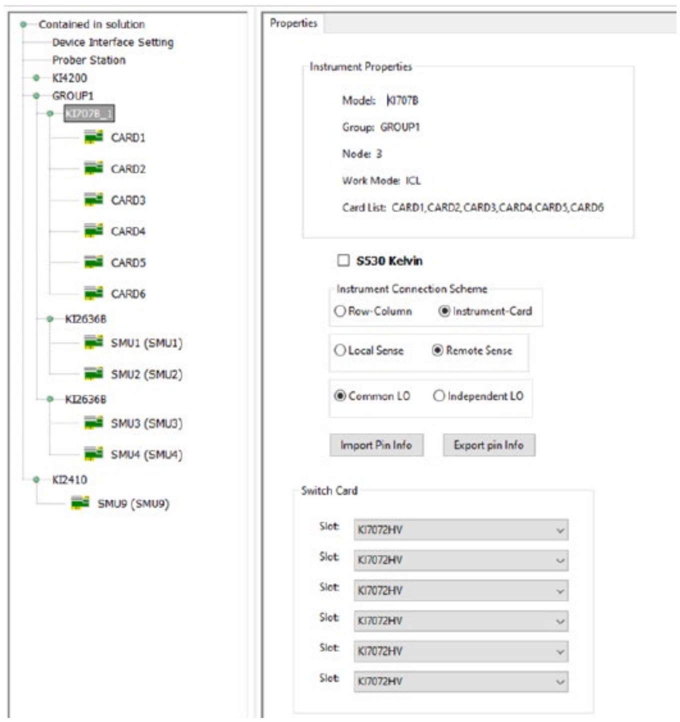
ACS supports both device dependent commands (DDC) mode and Test Script Processor (TSP) mode for Models 707B and 708B:

- DDC mode instruments must be manually added to the hardware configuration
- TSP mode instruments are automatically scanned and added to the hardware configuration

Model 707B and 708B configuration

When the instrument node of a Model 707B or 708B in the configuration navigator is selected, you can view the instrument Properties, Instrument Connection Scheme, and Switch Card configuration in the Properties tab of the matrix.

Figure 518: Model 707B or 708B properties



The Model 707B or 708B Properties tab contains:

- Instrument Properties: Model of instrument; associated group (TSP mode); node (TSP mode); Work Mode (TSP or DDC); Card List (ICL mode), and GPIB Address (DDC mode).
- Instrument Connection Scheme:
    a. Row-Column or Instrument-Card
    b. Instrument SENSE terminals: Local or Remote Sense
    c. Common LO or Independent LO
- Switch Card: select matrix cards for the matrix (Model 707B has six slots; Model 708B has one slot):

    Keithley 7071 Matrix Card

    Keithley 7072 Matrix Card

    Keithley 7072 HV Matrix Card

    Keithley 7072 HVD Matrix Card with Discharges

    Keithley 7136 Low Current MUX Card

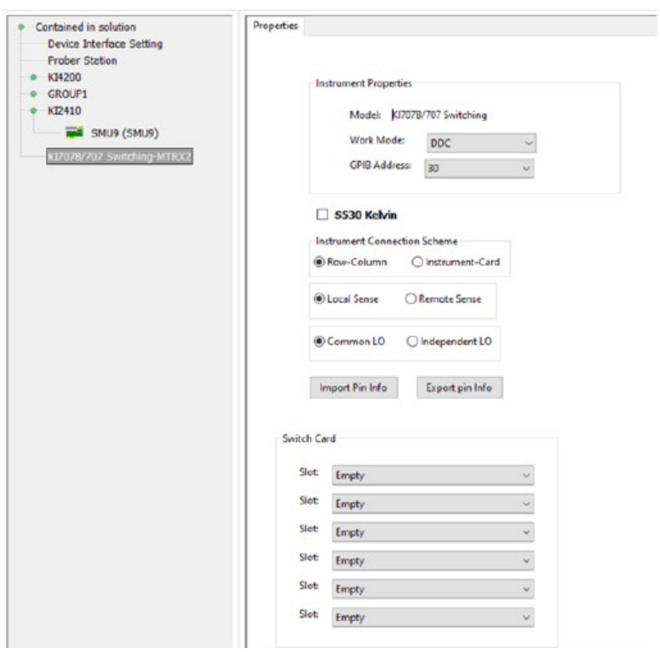
    Keithley 7174 Low Current Matrix Card

    Keithley 7530 Matrix Card

    Keithley 7531 Matrix Card

    Keithley 9174 Semiconductor Matrix Card

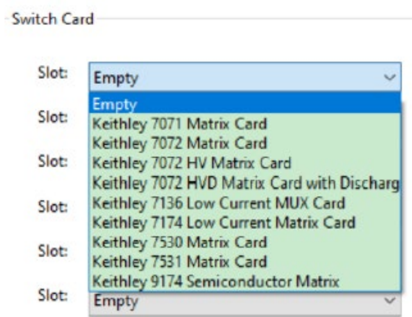
Figure 519: Model 707B or 708B properties (DDC mode)



NOTE

If any of the matrices in the system are set to Remote Sense, all the Series 2600B and Series 2400 instruments in the system are set to four-wire (4-W) mode when executing a test and the SMU Remote Sense option in the Preferences settings Instrument tab is disabled. When all the matrices are set to Local Sense, all the Series 2600B and Series 2400 instruments in the system are set to two-wire (2-W) mode when executing a test. It is highly recommended that all the matrices in the system are in the same mode. For more information on the Preference settings, refer to Automation functions (on page 7-1).

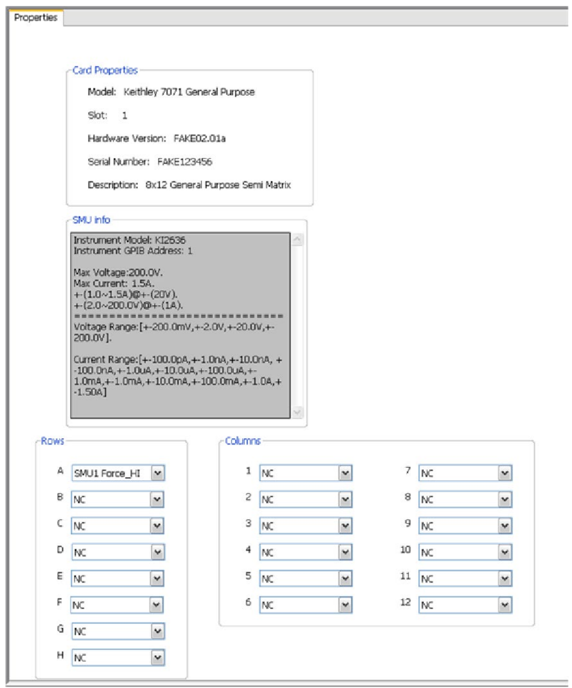
Figure 520: Models 707B and 708B switch cards



Matrix card information

When you select one of the matrix cards on the hardware configuration navigator, the GUI displays the card properties of the matrix card. You can configure the terminals and pins to the rows and columns. See the following figure for more detail.

Figure 521: Model 7071 matrix card properties



The matrix card Properties tab contains:

- **Card Properties:** Instrument model, slot number, hardware version, serial number, and card description.
- **SMU info:** Instrument model, instrument GPIB address, maximum voltage, maximum current, voltage range, and current range.
- **Rows:** A through H correspond to the eight rows of all Model 707B and 708B compatible matrix cards. Use the menus to connect the rows to the instrument terminals.
- **Columns:** 1 through 12 correspond to columns of all Model 707B and 708B compatible matrix cards. Use the menus to connect the columns to the instrument terminals or prober test-fixture pins, or any combination of both terminals and prober pins, depending on your needs.

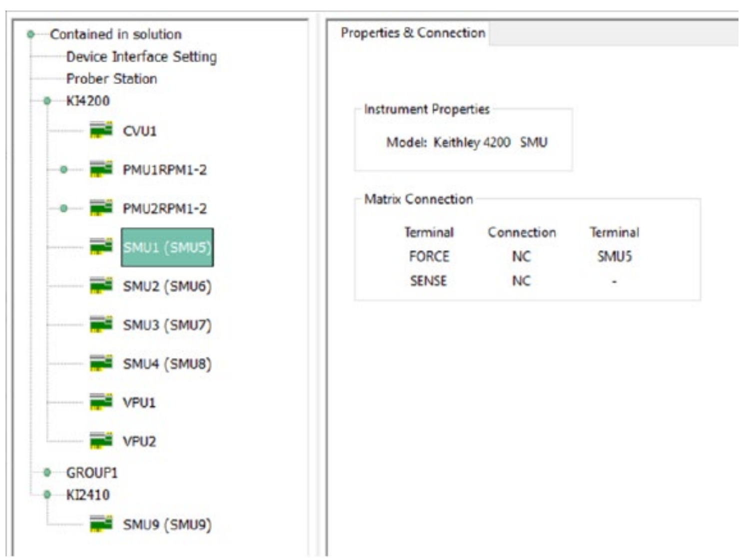
Model 4200A system properties

Selecting Keithley Instruments 4200A-SCS in the configuration navigator allows you to view the instruments in the configuration navigator, as shown in the following figure.

Figure 522: Model 4200 group properties

Selecting the SMU1 instrument in the configuration navigator will display the properties and connections for the instrument, as shown in the following figure.

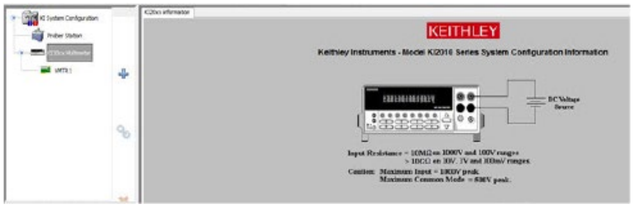
Figure 523: Model 4200 SMU information



Model 2010 system properties

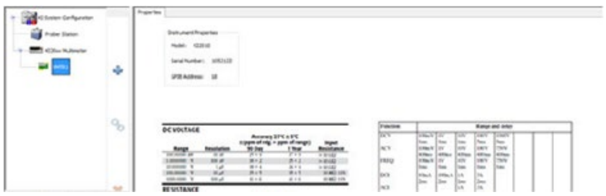
When you select the Model 2010 in the configuration navigator, the work area shows the instrument properties, specifications, and power capacity.

Figure 524: Model KI2010 system configuration



In the next figure, you will see the instrument name displayed, the serial number, and the GPIB address.

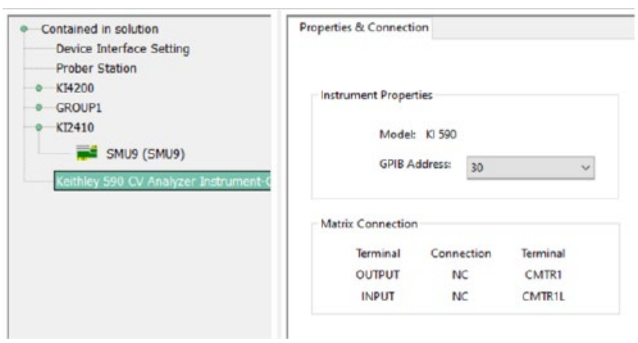
Figure 525: Model KI2010 configuration information



Model 590 C-V analyzer

When you select the Model 590 C-V Analyzer in the configuration navigator, the Properties & Connections tab opens.

Figure 526: Model 590 CV analyzer



The Properties & Connections tab provides access to the Model 590 instrument properties and useful switch-matrix connection information. The Instrument Properties area of this Properties & Connections tab provides access to the following Keithley Instruments Model 590 instrument properties:

- **Model:** Full vendor name, model number, and instrument description.
- **GPIB Address:** Primary GPIB address list. Addresses that are in use are displayed with asterisks (*) next to them. Range is 0 to 30 (GPIB address 31 is reserved as the controller address). If the selected GPIB address conflicts with the GPIB address of another system component, a red exclamation-point symbol (!) is displayed next to the selected address.

NOTE

You can programmatically read the GPIB address and other instrument properties on the system configuration using the LPTLib `getinstattr` function. Proper usage of `getinstattr` allows you to develop user libraries in a configuration independent manner.

Figure 527: GPIB address selection

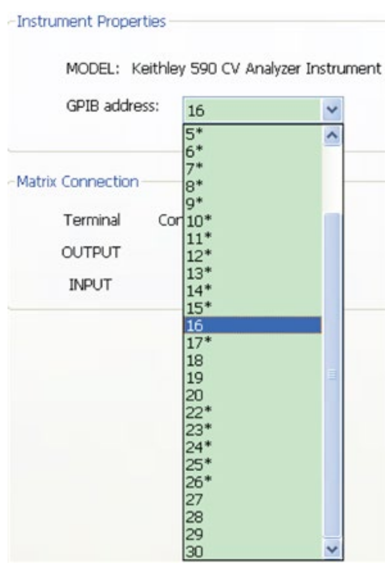


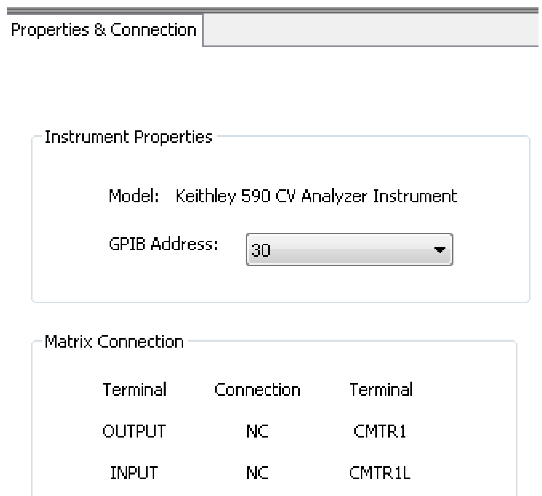
Figure 528: GPIB address properties

When a matrix is included in the system configuration, the matrix connections area of this tab displays the matrix connections that are associated with the Model 590 measurement terminals.

Capacitance meter configuration

When you configure a capacitance meter, the only item that you can change is the GPIB address. You will also see the matrix connection displayed.

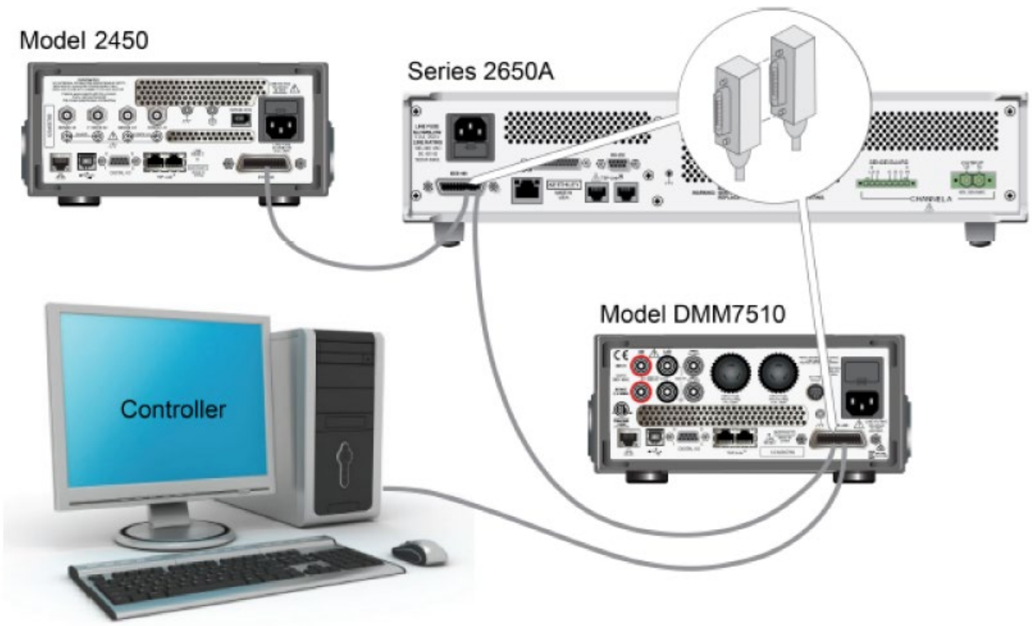
Figure 529: Configuration of capacitance meter



DMM7510 configuration

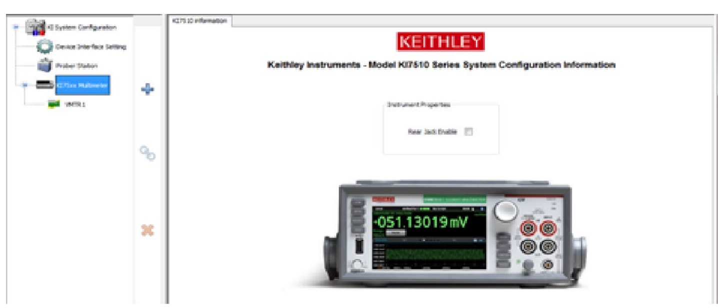
The Model DMM7510 is connected with a GPIB cable. When the Hardware Management Tool scans the instruments, it determines the instrument GPIB address from 0 to 30. It is considered an individual group and is recognized as VMTRx in ACS.

Figure 530: GPIB connection example



Each instrument that is included in the system configuration appears in the configuration navigator. Selecting a configuration navigator node will display the configuration information and instrument properties in the work area. See the following two figures.

Figure 531: Model DMM7510 system configuration



In the next figure, the instrument name, serial number, and GPIB address is displayed.

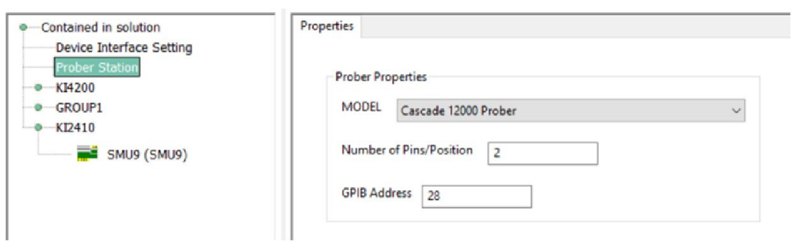
Figure 532: Model DMM7510 properties

The Model DMM7510 has four terminals: INPUT HI, SENSE HI, INPUT LO, and SENSE LO.

Prober station configuration

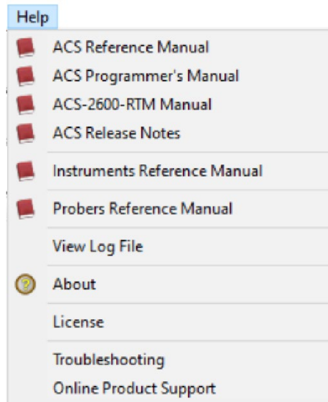
If the system has a configured prober station, it is displayed on the configuration navigator as a node. When this node is selected, the work area displays the properties of the prober station. You can configure the Model and Number of Pins/Position, as shown in the following figure.

Figure 533: Prober configuration interface



When setting up the prober, refer to the *S600 Probers Reference Manual*, in the Help menu, as shown in the following figure.

Figure 534: Probers Reference information



Configuration file locations

A configuration file is created and saved by you when you set up your test parameters. When you start ACS after creating a configuration file, the software will automatically find and read the file.

The following are examples of configuration file names and the path where the files are saved:

- `Hardware_Config_Online.kcf`
- `Hardware_Config_Demo.kcf`
- `\\ACS\KATS\CONFIG\`

Section 11

Hardware examples

In this section:

Test and control connections

This section contains several hardware examples for you to reference depending on your testing needs.

Switch matrix connections

A switch matrix enhances the connectivity of the system by allowing any source measure unit (SMU) or other instrument to be connected to any device under test (DUT) pin.

Switching mainframes

The following table lists the recommended switching mainframes along with a brief description of each. The Keithley Instruments Models 3706A, 707A, and 707B hold six matrix cards, while the Keithley Instruments Models 708A and 708B hold one matrix card.

Recommended switching mainframes

Mainframe	Description
Model 3706A	6-slot system switch with DMM
Model 3706A-NFP	6-slot system switch with DMM, without front panel and keypad
Model 3706A-S	6-slot system switch
Model 3706A-SNFP	6-slot system switch, without front panel and keypad
Model 707A and 707B	6-slot Switching Matrix Mainframe
Model 708A and 708B	Single slot Switching Matrix Mainframe

Integrated software control for the Keithley Instruments Models 707A, 707B, 708A, 708B and Model 3706A Switching Matrix is provided by the ACS software. For more information, you can refer to the Python user library.

For additional information about developing user modules and libraries and for information on the commands, refer to the *ACS Programmer's Manual*, LPT library reference section

Recommended matrix cards

The following table summarizes the recommended Keithley Instruments matrix cards, along with a brief description of each. Note that a key characteristic of these cards is low offset-current to minimize the negative effects of offset currents on low-current measurements.

Recommended switching matrix cards	
Matrix Card	**Description**
Model 3730	6x16 matrix, <100 pA offset current
Model 7072	8x12 matrix, <1 pA offset current
Model 7172	8x12 matrix, <500 fA offset current
Model 7174A	8x12 matrix, 100 fA offset current
Model 9174	8x12 matrix, 100 fA offset current

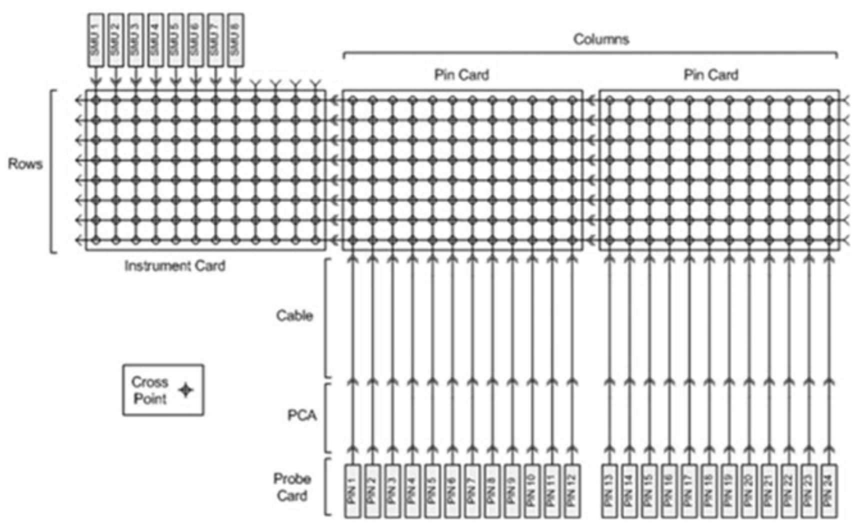
Switch mainframe control

The switch matrix is controlled by the system computer or Model 4200A-SCS using the GPIB (IEEE-488) or LAN interface. Refer to IEEE-488 connections (on page 11-4) for detailed bus connection information.

Typical SMU matrix card connections

The following figure shows a typical SMU matrix card connection used in a typical system. This is a configuration that includes 8 SMUs and 24 Pins. SMUs are connected to the instrument card. There is always one instrument card. Pins are connected to the pin cards. There are 1 to 5 pin cards.

Figure 535: SMU matrix connection of typical system

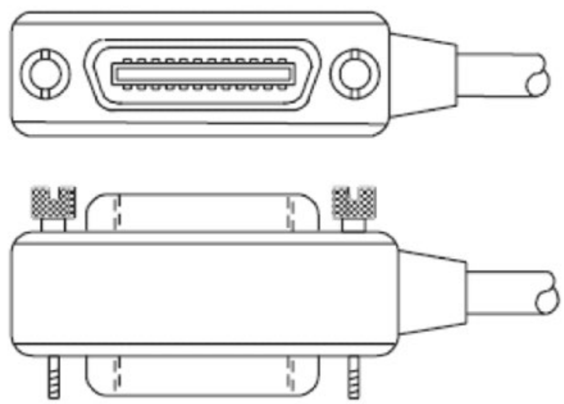


IEEE-488 connections

The built-in IEEE-488 interface allows you to connect to the system computer or Model 4200A-SCS to a variety of GPIB-equipped devices, such as Series 2400 SourceMeter instruments, a capacitance voltage (CV) meter or switching matrix. The Model 4200A-SCS unit can also be configured as a GPIB child and be controlled by an external host computer.

For GPIB communication, connect an IEEE-488 cable between the computer GPIB port and instrument GPIB port. The standard IEEE-488 connectors are shown in the following figure.

Figure 536: IEEE 488 connector



Recommended cables

To avoid electrical interference, use only shielded IEEE-488 connecting cables such as the Keithley Instruments Models 7007-1 and 7007-2.

Configure IEEE-488 controller operation

The Model 4200A-SCS can be configured to operate either as a GPIB controller or GPIB child.

Other test instruments with GPIB connectors can be configured as a GPIB child. The system computer with ACS installed and the Model 4200A-SCS acts as a GPIB controller when ACS is running. When operating as a controller, the Model 4200A-SCS reserves primary address 0, making that address unavailable to GPIB child devices such as GPIB switch matrices, CV meters, and automatic probe stations.

Drivers for these and other instruments, typically integrated into semiconductor test systems, are included with ACS.

Configure IEEE-488 child operation and control

The Model 4200A-SCS acts as a GPIB child when the Keithley External Control Interface (KXCI) software is running. When KXCI is running, the Model 4200A-SCS can be controlled by an external computer.

For more information about KXCI, refer to the 4200A-SCS documentation.

You can also visit tek.com/keithley to search for updated information by model number.

Other instruments that are considered a child in the hardware configuration can be controlled by the system computer or the Model 4200A-SCS using GPIB commands.

LAN connections

The system computer or Model 4200A-SCS can connect to devices that support TCP/IP and comply with IEEE Std 802.3 (ethernet), such as Series 2600B SourceMeter, Series 2400 SourceMeter, or switching matrix. These instruments can be connected to a system computer or Model 4200A-SCS through ethernet or directly to a LAN cable (as shown in the following figure). ACS can access the instruments with an IP address.

The IP address of these instruments need to be on the same subnet with the system computer or Model 4200A-SCS. For more information about how to set the IP address of the instruments, refer to the manual for the instrument.

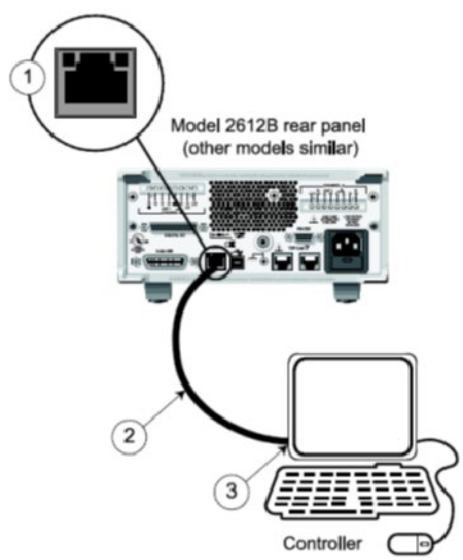
NOTE

The drivers for the Model 4200A-SCS LAN interface are pre-installed on the system. Contact your system administrator before attempting to enable the LAN interface. For the best results, use only CAT 5 UTP cables equipped with RJ-45 connectors to connect the Model 4200A-SCS to your LAN.

Use the following table to identify the items in the following figure.

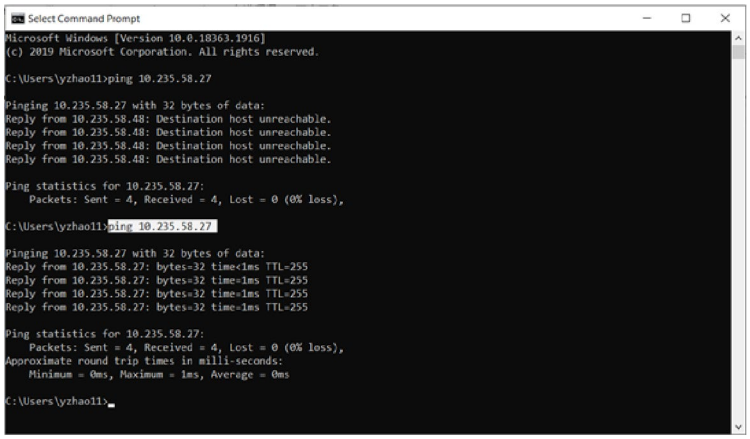
1	Series 2600B ethernet port (LAN)
2	Straight-through LAN cable or crossover LAN cable (Model 1746946XX)
3	Ethernet port (on the host computer)

Figure 537: ACS LAN connection



You can query the TCP/IP address of the connected instruments to verify the LAN connections, as shown in the following figure.

Figure 538: Query LAN connection

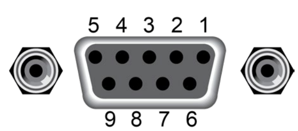


RS-232 connections

The RS-232 port allows you to interface with the system computer or Model 4200A-SCS with a variety of serial devices, such as a serial printer or plotter. It can also be used to control semi-automatic probe stations and other serial equipment.

The following figure shows a RS-232 (COM1) connector, which is a standard DB-9 (9-pin) male connector. A 9-pin-to-25-pin adapter may be used.

Figure 539: RS-232 interface connector



The following table provides the pin numbers and pin descriptions for the RS-232 connector.

Pinouts for the RS-232 connector	
Pin	**Description**
1	No connection
2	TXD, transmit data
3	RXD, receive data
4	No connection
5	GND, signal ground
6	No connection
7	RTS, ready to send
8	CTS, clear to send
9	No connection

Recommended serial cables

To avoid electrical interference, use only properly-shielded serial cables. Shielded 25-pin cables may be used with shielded 25-pin-to-9-pin adapters where needed.

TSP-Link connections

The TSP-Link™ is an expansion interface that allows the instruments to communicate with each other.

Parent and child instruments

In a TSP-Link system, one of the nodes (instruments) is the parent and the others are child nodes.

The parent can control the other nodes (child nodes) in the system. When any node transitions from local operation to remote, it becomes the parent of the system; all other nodes also transition to remote operation, and become the child nodes. When any node transitions from remote operation to local, all other nodes also transition to local operation, and the parent/child relationship between nodes is dissolved.

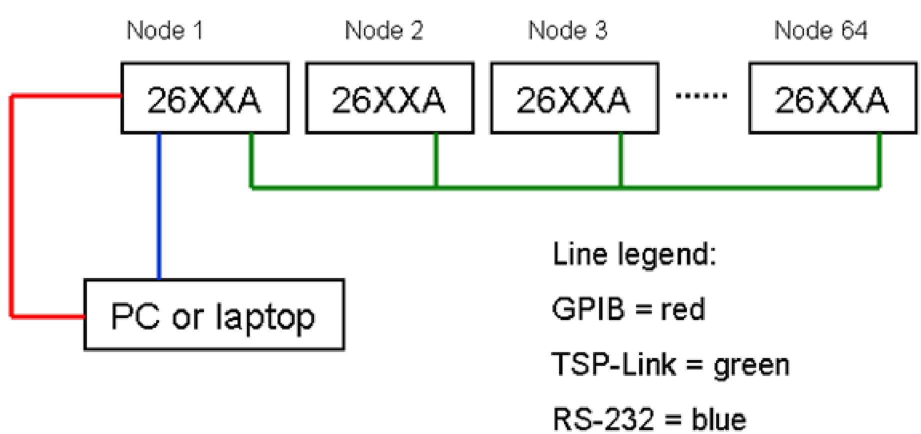
A child is a node that is controlled by the parent. The GPIB and RS-232 command interfaces of the child nodes are disabled.

Connections for an expanded system are shown in the following figure. As shown, one unit is optionally connected to the system computer using the GPIB or RS-232 interface and all the units in the system are daisy-chained together using LAN crossover cables. Each TSP-enabled instrument has two TSP-Link connectors. You must use a LAN crossover cable for each TSP-Link connector that is type category 5e or higher. The maximum cable length between each node is 3 meters.

NOTE

A computer is not needed for standalone systems.

Figure 540: TSP-Link connections



4200A-SCS LAN connections

The Model 4200A-SCS can be connected to an ethernet LAN so that tests and data can be easily accessed and archived. From a networking perspective, the Model 4200A-SCS operates the same as any other personal computer running Microsoft Windows. The following information outlines the LAN connections and recommended cables.

NOTE

The drivers for the Model 4200A-SCS LAN interface are preinstalled on the system. Contact your system administrator before attempting to enable the LAN interface.

For the best results, use only CAT 5 UTP cables equipped with RJ-45 connectors to connect the Model 4200A-SCS to your LAN.

Basic configuration

ACS software installations are supported on the following hardware configurations:

- On a laptop or desktop computer for Series 2600B System instruments.
- On a laptop or desktop computer for Series 2600B System instruments and Series 2400 instruments.
- On a laptop or desktop computer for Series 2600B System instruments and Series 3700A instruments.
- On a laptop or desktop computer for Series 2400 instruments and Model 707A, 707B, 708A, or 708B instruments.
- On a Model 4200A-SCS.
- On a laptop or desktop computer for Model 4200A-SCS.
- On a Model 4200A-SCS for Series 2600B instruments.

NOTE

Different test modules should be used for each configuration.

ACS on a laptop or desktop computer

Configuration 1, 2, and 3

NOTE

Configuration 2 includes Series 2600B System instruments and Series 2400 instruments. Configuration 3 includes Series 2600B System instruments and Series 3700A instruments.

ITMs and STMs: Only control Series 2600B instruments (and Series 2400 instruments, when part of the configuration).

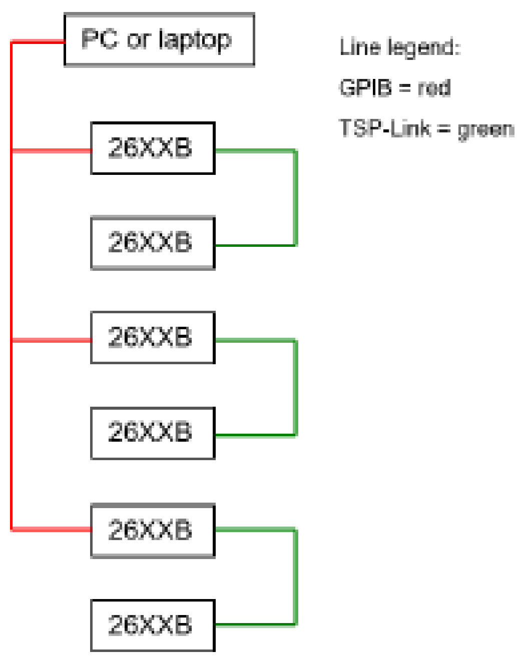
CTMs: Only control certain external instruments, such as switch matrices, pulse generators, and CV meters.

PTMs: Control instruments and process data.

To use ACS on laptop or desktop computer:

1. Install a CEC GPIB card (KUSB-488A or KPCI-488) on a system computer with the latest driver.
2. Connect the master instrument to the system computer or laptop through the GPIB.
3. Connect the child instruments to the master one-at-a-time using TSP-Link™ (see the following figure).
4. Make sure each group of Series 2600B instruments has a master. The Series 2600B that links to the controller (system computer or Model 4200A-SCS) with the GPIB cable is the master.
5. Connect each Series 2600B in the group to the TSP-Link. Refer the *Series 2600B System SourceMeter® User's Manual* for more details.

Figure 541: ACS on a computer for Series 2600B instruments



Assign a GPIB address

Each group of Series 2600B instruments should have a unique GPIB address (refer to the *Series 2600B System SourceMeter® User's Manual* documentation for more details).

The master's GPIB address is the group's GPIB address. It is not necessary to change all Series 2600B GPIB addresses to the same number.

NOTE

Address 21 is reserved for the system controller (GPIB card).

Assign a node number

Each Series 2600B instrument in a group must have a unique node number. Node numbers should be assigned sequentially, since the SMU numbers are assigned automatically when the system is started and according to the node number sequence. If the node numbers were not assigned sequentially, the SMU numbers will not be consistent with the setup.

To assign node numbers:

1. Assign the master to Node 1.
2. Assign the Series 2600B that is linked with the master by TSP-Link to Node 2.
3. Sequentially assign all other Series 2600B node numbers.

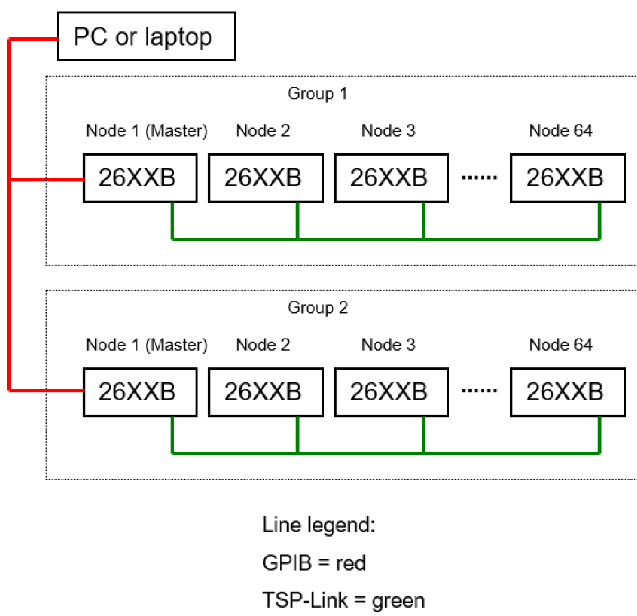
Check the node, SMU, and timer numbers

Follow these steps to check the node, SMU, and timer numbers on the Series 2600B front panels. For a Series 2600B master without Linear Parametric Test (LPT) functions:

- When performing a scan of the ACS hardware configuration, the node numbers, SMU numbers, and timer numbers are displayed on each of the Series 2600B front panels.
- For a Series 2600B master with LPT functions installed, each time the master is turned on, the information is displayed automatically.
- Send the string `showsmu(10)` to the GPIB address of a Series 2600B master by any GPIB communication software (for example, the `TRTEST.exe` of CEC 488). The information is displayed for 10 seconds (note this method needs LPT functions).

The following figure shows a typical hardware configuration.

Figure 542: Typical hardware configuration



NOTE

There are two TSP-Link connectors on the back panel of each Series 2600B instrument. Also, make sure that you use a LAN crossover cable for each TSP-Link connector that is type category 5e or higher, and length three meters maximum between nodes.

Connect to a computer

Each Series 2600B instrument master and other GPIB devices should be connected to the computer by a GPIB cable. There are no special limits for the topology and the sequence. Follow these steps to connect with a computer:

If only one GPIB card is installed:

- All GPIB devices (such as Series 2600B instruments, the prober, and the Model 3401 Pulse Pattern Generator) should all connect to this GPIB card.

If more than one GPIB card is installed:

- All Model 4200A-SCS related instruments and probers should be connected with a GPIB card 0.
- All Series 2600B groups must be connected with the GPIB card that holds the highest card number. You can check the cards' assigned number using the `TEST488.EXE` routine, which is in CEC488 driver folder.

Connect to the device under test (DUT)

ACS supports the Model 4200A-SCS and Series 2600B instruments. For more information on how to connect a DUT for these instruments, refer to the Keithley Instruments website for the specific customer documentation.

Series 2400 instruments and Model 707A/707B/708A/708B instruments controlled with ACS

Configuration 4

ITMs and STMs: Only used to control Series 2400 instruments.

CTMs: Only used to control certain external instruments, such as switch matrices, pulse generators, and CV meters.

PTMs: Only used to control Series 2400 instruments and Model 707A/707B/708A/708B instruments, and to control certain external instruments, such as pulse generators and CV meters.

ACS on the Model 4200A-SCS

Configuration 5

ITMs: Not used.

STMs: Only used for importing KTM files.

CTMs: Used to control the Model 4200A-SCS internal hardware, such as the SMU, PGU, or PMU. CTMs can also be used to control external instruments such as switch matrices, pulse generators, and CV meters.

NOTE

This configuration includes all instruments and equipment that are supported by KITE (see the following figure).

Connect the master instrument to the 4200A-SCS using the GPIB interface.

Figure 543: ACS software on Model 4200-SCS

ACS on a laptop or desktop computer for a Model 4200A-SCS

Configuration 6

ITMs: Only used to control the Model 4200A-SCS using an ethernet cable and KXCI interface.

PTMs: Used for CV testing.

ACS on the Model 4200A-SCS with Series 2600B instruments

Configuration 7

ITMs: Not used.

STMs: Only used to control Series 2600B instruments and for importing KTM files.

CTMs: Only used to control the Model 4200A-SCS internal hardware, such as the SMU, PGU, PMU, and SCP. Also, they can be used to control certain external instruments, such as switch matrices, pulse generators, and CV meters.

NOTE

ACS supports multigroup Series 2600B units. Each group can include up to 64 Series 2600B System SourceMeter source-measure units (SMU) with a maximum of 128 SMUs.

1. Connect the master instrument to the 4200A-SCS using the GPIB interface.
2. Connect the child instruments one-at-a-time to the master with a TSP-Link.
3. Make sure each group of Series 2600B instruments has a master. The Series 2600B that links to the controller (system computer or Model 4200A-SCS) with the GPIB cable is the master.
4. Connect each Series 2600B in the group to the TSP-Link (refer the *Series 2600B System SourceMeter® User's Manual* documentation for more details).

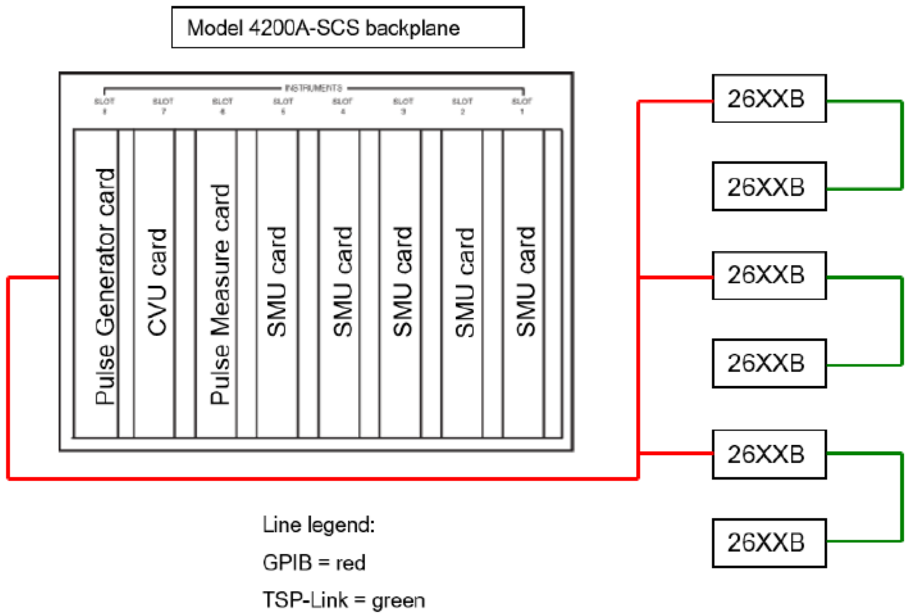
For more information about TSP-Link, refer to the instrument documentation.

The following figure shows an example configuration setup with the Model 4200A-SCS connected with a GPIB cable to the Series 2600B instruments.

NOTE

The example of the Model 4200A-SCS backplane may not be typical for your configuration. The cards that are used in the eight slots of the 4200A-SCS backplane can be any combination.

Figure 544: ACS on a Model 4200-SCS with Series 2600A instruments



Keithley Instruments
Corporate Headquarters • 28775 Aurora Road • Cleveland, Ohio 44139 • 440-248-0400 • 1-800-833-9200 • tek.com/keithley

KEITHLEY
A Tektronix Company

07/2020