www.keithley.com

Automated Characterization Suite (ACS) Basic Edition

Reference Manual

ACSBASIC-901-01 Rev. E / August 2012

ACSBASIC-901-01

KEITHLEY

A GREATER MEASURE OF CONFIDENCE

Automated Characterization Suite (ACS) Basic Edition

Reference Manual

Keithley's standard terms and conditions of sale in effect at the time of acceptance of buyer's order by Keithley shall apply to all purchase of goods and performance of services from Keithley, to the exclusion of any additional or different terms and conditions, including any terms or conditions which buyer may purport to apply under any buyer's request for quotation, purchase order or similar document, or which buyer may offer in response to these terms. A copy of Keithley's current terms can be accessed at http://www.keithley.com/company/termsandconditions (these "Terms"). To obtain a printed copy of these Terms, please contact your local sales office or send an email to orders@keithley.com. Buyer's assent to these Terms, and only these Terms, shall be conclusively presumed from buyer's acceptance of delivery of the products and/or services provided by Keithley.

Document number: ACSBASIC-901-01 Rev. E / August 2012

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 qualified 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 Instruments products are designed for use with electrical signals that are rated Measurement Category I and Measurement Category II, as described in the International Electrotechnical Commission (IEC) Standard IEC 60664. Most measurement, control, and data I/O signals are Measurement Category I and must not be directly connected to mains voltage or to voltage sources with high transient overvoltages. Measurement Category II connections require protection for high transient overvoltages often associated with local AC mains connections. Assume all measurement, control, and data I/O connections are for connection to Category I sources unless otherwise marked or described in the user documentation.

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.

The instrument and accessories must be used in accordance with its specifications and operating instructions, or the safety of the equipment may be impaired.

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

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 safety earth 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 safety earth ground using the wire recommended in the user documentation.

The ⚠ symbol on an instrument means caution, risk of danger. The user should 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 caution, 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 Hg 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 dangers 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.

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 Instruments. Standard fuses with applicable national safety approvals may be used if the rating and type are the same. 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 Instruments to maintain accuracy and functionality of the product). If you are unsure about the applicability of a replacement component, call a Keithley Instruments office for information.

To clean an instrument, use a damp cloth 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.

Table of Contents

Section 1

ACS Basic introduction

In this section:

ACS Basic software overview

If you have any questions after reviewing this information, please contact your local Keithley Instruments representative or call one of our applications engineers at 1-888-KEITHLEY (1-888-534-8453) within the U.S. and Canada. You can also visit the Keithley Instruments website at www.keithley.com for updated worldwide contact information.

NOTE

When the Series 2600B System SourceMeter® instruments are referenced, it also includes the Series 2600A System SourceMeter instruments, since these two series of instruments are fully interchangeable. However, the following instruments are not supported in ACS Basic: Model 2604B, Model 2614B, and Model 2634B.

ACS Basic software can control external instruments through GPIB or LXI communication, such as the Keithley Instruments Series 2600B System SourceMeter® instruments. It can also control hardware on the Keithley Instruments Model 4200-SCS Semiconductor Characterization System. It also provides a combined test-execution engine that supports Model 4200-SCS and Series 2600B group testing.

ACS Basic supports all of Keithley's source and measure instrument products, including Series 2600B, Series 2400, and Models 2651A and 2657A SourceMeter instruments, Model 237 SMU, and Model 4200-SCS SMUs.

ACS Basic can perform multisite parallel I-V testing using a Series 2600B as the primary instrument. You can have as many as 16 groups of instruments on a single GPIB card, for a total of 16 parallel test sites (contact Keithley Instruments if you need more than 16 parallel sites). Each group can support up to 32 Series 2600B instruments, or 64 channels.

ACS Basic has three modes:

1. **Single-test mode**: allows you to perform a single test on a single device. You can select a test from one of three libraries based on the device and instruments available. You can also create your own test or modify available tests.
2. **Multi-test mode**: allows you to perform multiple tests on a single device (target application), or multiple tests on multiple devices. It is easier to use external instruments (DMM, switch, etc.) in multi-mode.
3. **Trace mode**: allows you to perform interactive operations with a SourceMeter and is only used with the Series 2600B, Model 2651A, and Series 2400 instruments. You can adjust the maximum sweep value and maximum power to the device. You can easily transition between DC and pulse forcing modes.

ACS Basic provides three types of test modules:

1. Graphical interactive test module (ITM)
2. Script test module (STM)
3. Python language test module (PTM)

An ITM allows you to define a test interactively using a graphical user interface (GUI). An STM supports Test Script Processor (TSP™) files. A TSP file is written to customize tests for Series 2600B instrument(s) or a Model 3706A System Switch/Multimeter instrument. A PTM can control any GPIB instrument. Also, a script editor is included in ACS Basic for editing STM and PTM modules.

NOTE

When the Model 3706A and Series 3700A System Switch/Multimeter instruments are referenced, it also includes the Model 3706 and Series 3700 System Switch/Multimeter instruments, since these instruments are fully interchangeable.

Reference manual content

This manual contains information about the ACS Basic Edition software only. It includes information about how to install, configure, and operate the ACS Basic software, as well as full details of software features such as building, loading, and executing test projects.

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

- Models 707/707A/707B and 708/708A/708B Switching Matrix
- Series 2400 SourceMeter®
- Series 2600 System SourceMeter
- Series 2600B System SourceMeter
- Model 2651A High Current System SourceMeter
- Model 2657A High Power System SourceMeter
- Series 3700A System Switch/Multimeter
- Model 4200-SCS Semiconductor Characterization Suite

You can also refer to the supplied documentation that is located on the Keithley Instruments product information CD-ROM that is shipped with the software (CD Part Number: ACSBASIC-950-01). This reference manual is also in the software Help drop-down menu. Additionally, you can visit the Keithley Instruments website at www.keithley.com to search for updated information by model number.

Note that you should have received an ACS Basic Quick Start Guide with your order that you can use to quickly learn about ACS Basic, which tells you how to connect, install, and test a single instrument. Additionally, there is an ACS Basic Libraries Reference manual that contains detailed information about the Series 2600B LPT library and the device library. This manual is on the product information CD-ROM mentioned previously and is part of the Help that is included in the ACS Basic software.

Section 2

ACS Basic software introduction

In this section:

Pre-installation software requirements

ACS Basic software is distributed on a CD-ROM and can be installed on any computer that meets the minimum configurations as described below. Additionally, ACS Basic software is supported on a computer/laptop for the Series 2600B System SourceMeter® units, the Model 4200-SCS, and other external GPIB instruments.

For detailed configuration information, refer to the Hardware Configuration topic.

NOTE

It is recommended that you back up your previous version of ACS or ACS Basic before installing a new version of the software.

When you install ACS Basic software, the following system requirements apply:

Minimum configuration for a computer:

Operating System: Microsoft® Windows® XP Professional Service Pack 3 (SP3), Windows 7 x 86 (32 bit), or Windows 7 x 64 (64 bit)

- CPU: Pentium 4, 2.4GHz+ or equivalent
- System Memory (RAM): 1GB
- Hard Disk: 600MB free space
- Graphics Card: 16MB of VRAM
- Screen: 1024 x 768, 32-bit True Color

Recommended configuration for a computer:

Operating System: Microsoft Windows XP Professional Service Pack 3 (SP3), Windows 7 x 86 (32 bit), or Windows 7 x 64 (64 bit)

CPU: Intel® dual core class, AMD Athlon® class or higher

- System Memory (RAM): 2GB
- Hard Disk: 1GB free space
- Graphics Card: 64MB of VRAM
- Screen: 1280 x 1024, 32-bit True Color

NOTE

ACS Basic software can use GPIB cables to communicate with instruments. Before using ACS Basic, you need to make sure that the GPIB driver is functioning properly in your operating system (Windows XP Professional Service Pack 3 or Windows 7).

Software installation

NOTE

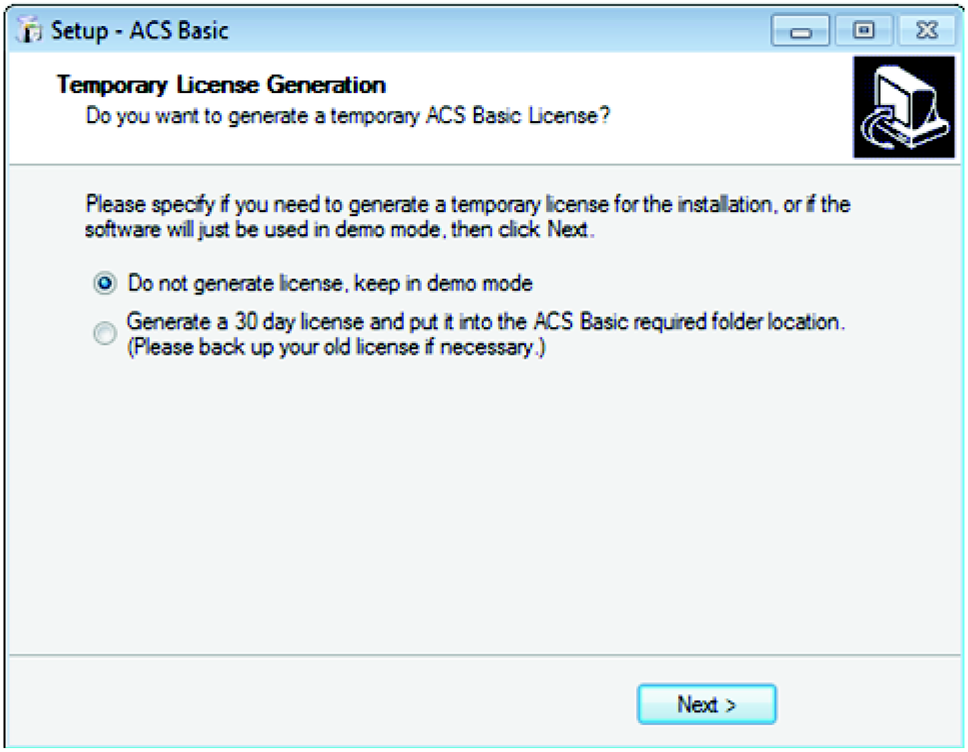
Before installing ACS Basic, make sure you have read and have completed all of the pre-installation requirements (see previous information).

ACS Basic installation instructions:

1. Restart your computer.
2. Log in as an administrator.
3. Insert the ACS Basic setup disk into the CD-ROM.
4. The CD should open automatically. If it does not open, click **My Computer**, and open the CD.
5. Find the file named **index.html** and double-click.
6. When the CD opens you can click on the software to start the installation.
7. Once the ACS Basic Setup Wizard dialog box opens, follow the instructions to install the software on your system.

When the installation process is complete, the Temporary License Generation dialog box opens (see next figure)(note that if you already have a USB license key, select "Do not generate license, keep in demo mode" in order to continue installation).

Figure 1: Temporary license generation dialog box



Setup - ACS Basic

Temporary License Generation

Do you want to generate a temporary ACS Basic License?

Please specify if you need to generate a temporary license for the installation, or if the software will just be used in demo mode, then click Next.

● Do not generate license, keep in demo mode

○ Generate a 30 day license and put it into the ACS Basic required folder location. (Please back up your old license if necessary.)

NOTE

After you have installed ACS Basic software, and before you begin to set up any projects or run any tests, it is recommended that you review the 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 can be found by clicking on your Windows Start icon > All Programs > ACS Basic > Reference > Release Notes.

Section 3

ACS Basic software overview

In this section:

Before starting ACS Basic

1. Make sure that all of the instruments are connected with a GPIB/LAN cable. If more than one Series 2600B SourceMeter® instrument is used in a group, make sure that the TSP-Link™ connection has been made correctly. For more information on setting up the hardware, refer to the Hardware Configuration topic for specific connection and instrument setup instructions.
2. Make sure that all the instruments are turned on and self-testing is completed on all of the instruments.
3. Assign GPIB addresses and node numbers (see Configure the GPIB interface for more information).

Start ACS Basic

1. Click the ACS Basic icon (either a desktop icon or a quick launch icon chosen at installation). You can also find in the Start menu.

⚠ CAUTION

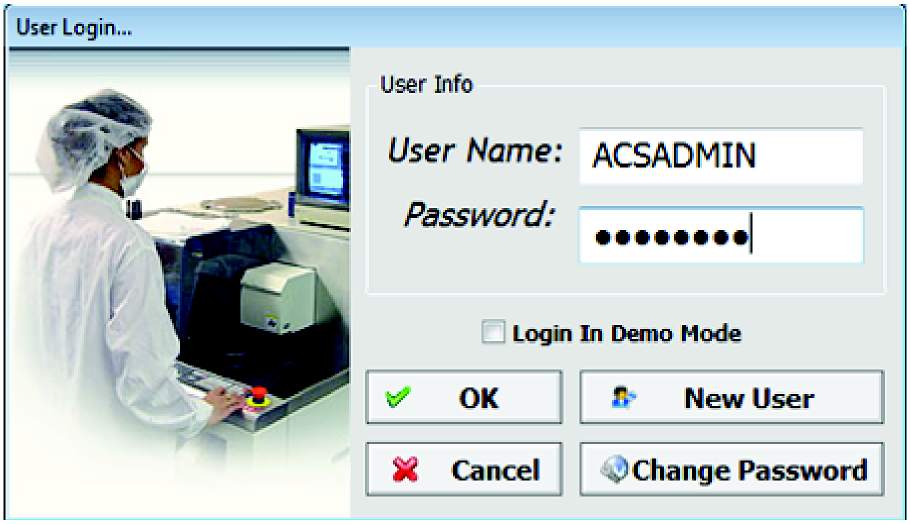
FATAL ERROR POSSIBLE. To avoid fatal errors to instruments, never start ACS Basic software until all the instruments have completed self-testing. For those who are using Series 2600 and 2600B instruments with TSP-Link: Always start the "subordinate" instruments first, and then the "master." Since the master scans through all the linked subordinate instruments, it must be started after all the subordinate machines are turned on, or errors may occur when you start ACS Basic

NOTE

You must obtain an ACS Basic software license key in order to start the software in professional mode. If you have not obtained a USB license key, you can select the Login In Demo Mode function. This will allow you to start ACS Basic without a software license.

2. Enter the password for ACS Basic in the User Login dialog box (the default User Name is ACSADMIN, and there is not a default Password; see next figure).

Figure 2: User Login window



User Name: ACSADMIN

Password: ●●●●●●●●

NOTE

As shown in the previous figure, you can register new accounts using the New User function (see Creating new users). To change your password, use the Change Password function.

ACS Basic USB license key

ACS Basic requires a user license in order to work in either professional or evaluation mode. Upon installation, the software can generate a temporary license file that is valid for 30 days.

NOTE

In previous versions of ACS Basic, the licensekey.txt file stored the licensing information. This license key was tied to the computer ID of a specific computer on which ACS Basic was installed. Once you have a permanent License File, you can install it by either:
Copying the file into the C:\ACS_BASIC\KATS folder, or while in demo mode, clicking the menu Help > Update License File and importing the license file. You can also click the Help menu, then Request a license to learn how to get a license file. The licensekey.txt file will provide you with the ability to remote login to the system to launch and run ACS Basic.

By default, in the absence of a license key, the software will launch in demo mode. Additionally, if you do not have any external devices or instruments connected to the computer, a license key will not be required. However, you will only be able to use ACS Basic in demo mode.

The ACS Basic SafeNet Sentinel USB license key is similar to a standard USB flash drive (see next figure). The USB license key is included with the ACS Basic software package. The USB license key allows you to operate ACS Basic in the professional mode or evaluation mode, and can be moved from one computer to another (multiple installations) without buying additional licenses.

NOTE

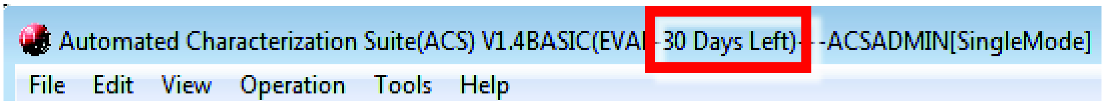
The evaluation license is a temporary key that you can use for 30 days.

Figure 3: USB SafeNet Sentinel license key

Before you start running the ACS Basic software, insert your USB license key into the computer's USB interface. The computer and ACS Basic software will automatically detect the USB key, then ACS Basic will start in the licensed mode (for example, Professional License or Evaluation License).

If you have the temporary key, ACS Basic will start in the evaluation mode, which has the same functionality as professional mode. The status of the temporary license key and the amount of days that you have available is displayed while running ACS Basic in the title bar (see next figure).

Figure 4: Evaluation mode with temporary license file

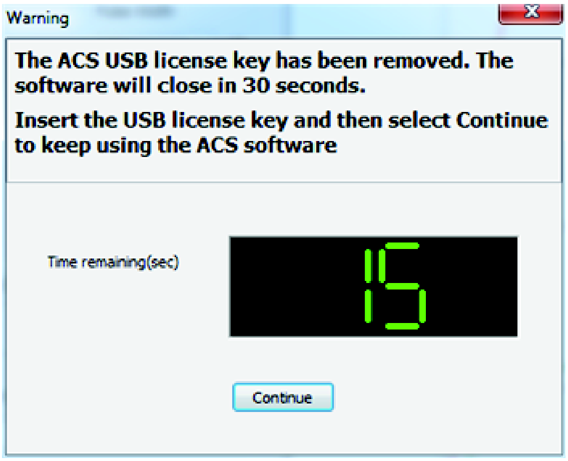


Automated Characterization Suite(ACS) V1.4BASIC(EVAL-30 Days Left)--ACSADMIN[SingleMode]

⚠ CAUTION

DO NOT remove the USB key from the computer while you are using the ACS Basic software in the professional or evaluation mode. If the key is removed, you will receive a warning message that indicates you have 30 seconds to insert the USB key back in the computer (see next figure). After the key is reinserted in the computer, you must click Continue in order to keep ACS Basic functioning. If the key is not inserted in the computer before the time has elapsed, the software will close and you will lose all data from your current session.

Figure 5: USB Warning message



Warning

The ACS USB license key has been removed. The software will close in 30 seconds.

Insert the USB license key and then select Continue to keep using the ACS software

Time remaining(sec)

15

Update a temporary USB key

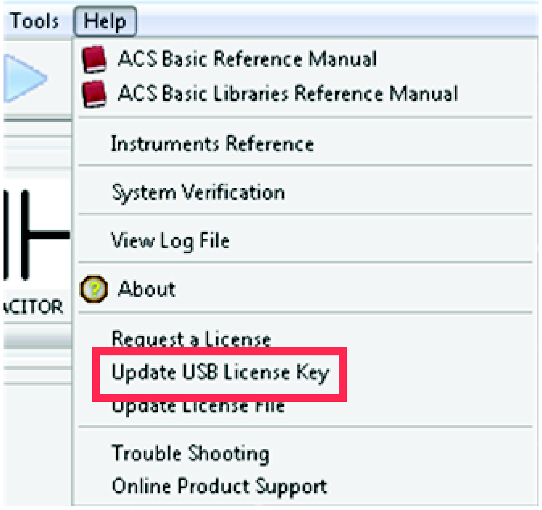
If your temporary USB license key has expired, you can request an update from Keithley Instruments.

To avoid an interruption in your operations and continue using the ACS Basic software in evaluation mode, perform the following steps to obtain an updated license key file.

Update a USB key:

1. In the ACS Basic Help menu, click **Update USB License Key** (see next figure).

Figure 6: Update License Key



2. After you select Update USB License Key, the Secure Update Utility dialog box opens. Click the **Generate Request Code** function to generate a file and save it.
3. E-mail the generated file to the following address: acs.license@keithley.com
4. After you send the generated file to Keithley Instruments, an updated file will be generated and returned to you by e-mail.
5. The newly generated file is named ACS_Basic update.upw, or something similar.
6. After you receive the updated file for your USB key, save it to a location on the computer.
7. In the ACS Basic Help menu, click **Update USB License Key**.
8. In the Secure Update Utility dialog box, click the folder icon to browse and locate the file you received. Select the file and click **Open** to upload it.
9. In the Secure Update Utility dialog box, click the **Apply Code** function.

NOTE

Before the temporary USB license key expires, you can purchase a permanent license key for ACS Basic to maintain full functionality.

Operating modes

ACS Basic starts in one of three different operating modes, depending on your license status and hardware setup (refer to the ACS Basic license key topic for more information). The three possible modes are:

- Demo
- Evaluation
- Professional

Demo mode

Demo mode is suitable for developing test projects and for reviewing results from previous tests.

When instruments are connected to the computer and ACS Basic starts in demonstration mode, then the USB license key is not correctly installed or has expired (see the Update a temporary USB key topic for more information) or has not been purchased.

NOTE

In Demo mode, ACS Basic cannot communicate with test or measurement hardware.

Evaluation mode

This mode has all of the functionality of ACS professional and lasts for 30 days. It requires a USB license key that contains a temporary license. See the Update a temporary USB key topic to find out how to renew your key.

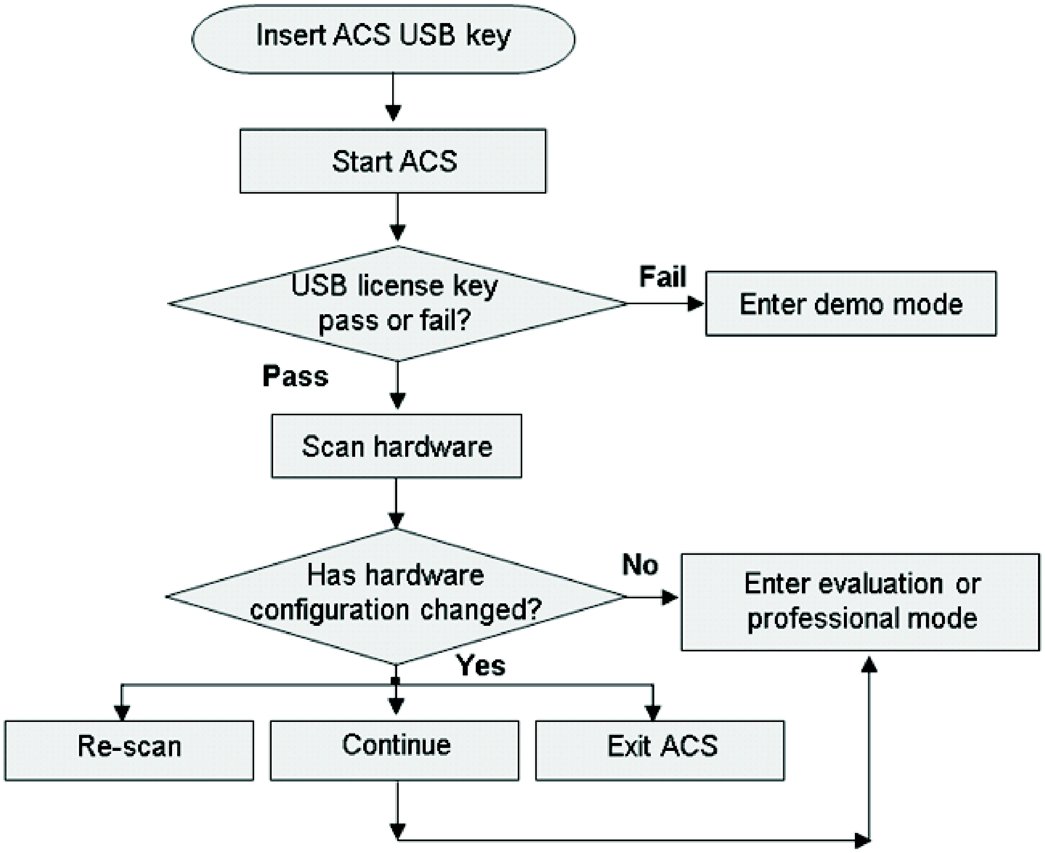
Professional mode

When ACS Basic starts in professional mode, it is an indication that the software has a valid USB license key.

ACS Basic may indicate that the default project requires hardware that is not available when starting in professional mode. You will notice this indication when ACS Basic is started for the first time with a new hardware configuration.

The following flowchart (see next figure) gives an example of the software's internal process flow during the startup sequence. The software modes are based on the USB license key and hardware configuration.

Figure 7: License key software modes


Insert ACS USB key
Start ACS
USB license key pass or fail?
Fail
Enter demo mode
Pass
Scan hardware
Has hardware configuration changed?
No
Enter evaluation or professional mode
Yes
Re-scan
Continue
Exit ACS

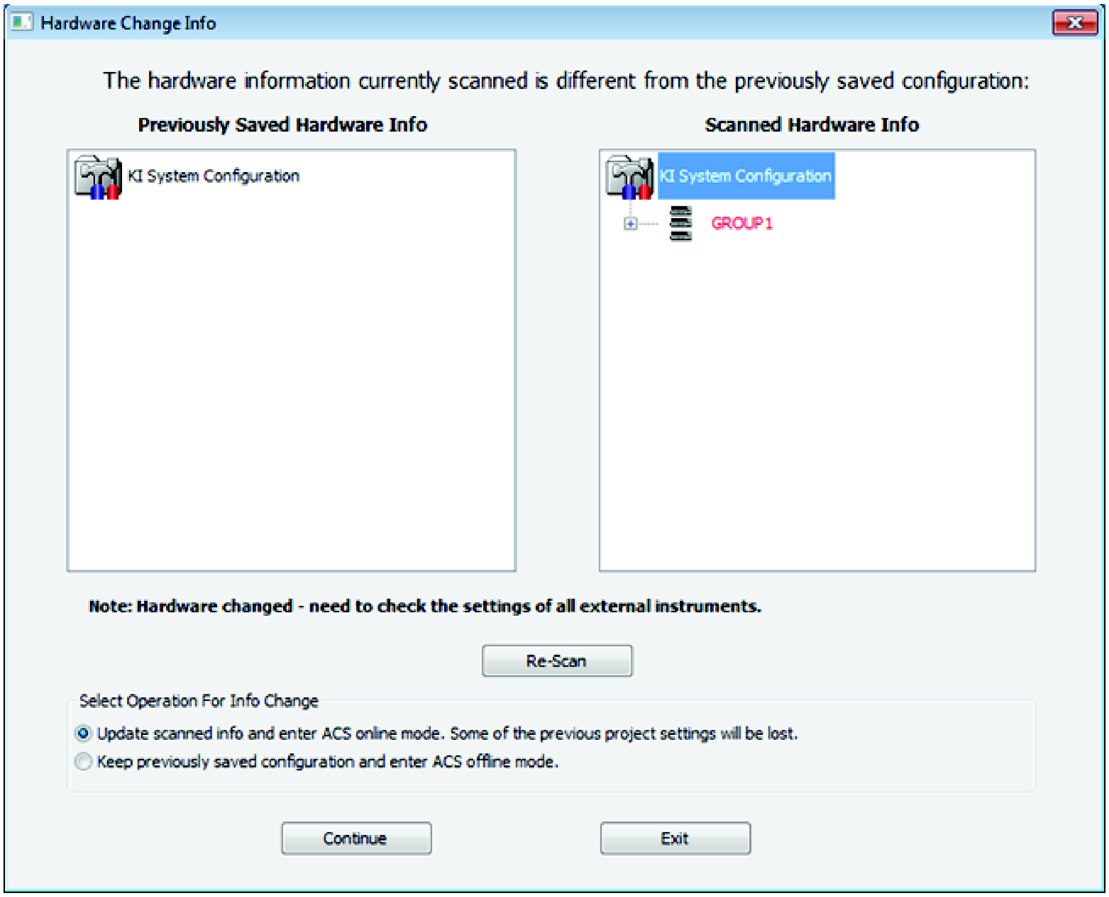
Once the scan has completed, the software determines the state of the configured hardware. If nothing has changed, ACS Basic will automatically go to professional (online) mode or evaluation (online) mode depending on your license key. However, if ACS Basic detects a different hardware configuration, compared to the currently saved configuration (for example, the physical hardware configuration does not match the configuration on file), a verification dialog box opens. You can select Re-scan, Continue, or Exit (see next figure).

1. Re-scan: will re-scan all connected instruments.
2. Continue: will accept the newly scanned hardware as the current configuration and continue in online mode, if the "Keep previously saved configuration and enter ACS online mode" is selected.
3. Exit: will exit ACS Basic without updating the hardware configuration file.

NOTE

If you have not obtained an ACS Basic software USB license key, then ACS Basic will start in Demo mode, regardless of any changes to the hardware configuration.

Figure 8: Hardware change info


Hardware Change Info

The hardware information currently scanned is different from the previously saved configuration:

Note: Hardware changed - need to check the settings of all external instruments.

Select Operation For Info Change

● Update scanned info and enter ACS online mode. Some of the previous project settings will be lost.
○ Keep previously saved configuration and enter ACS offline mode.

NOTE

When the ACS Basic software is started in professional or evaluation mode, the startup time will depend on how many instruments are in the hardware configuration group. For example, if there are several instruments, the startup time will be longer.

Account management

User accounts

ACS Basic software provides three types of user accounts and each has different rights and security features.

1. **Administrator** (or engineer administrator): This account has unrestricted rights to create, edit, and run test projects, including test parameters within ACS Basic. The administrator can also edit or delete operator accounts. The default administrator account user name is ACSADMIN (case sensitive).

NOTE

There can be other engineer user accounts, however, there is only one engineer administrator.

2. **Engineer**: This account has unrestricted rights to create, edit, and run test projects, including test parameters within ACS Basic. However, an engineer account cannot edit or delete operator accounts.
3. **Operator**: This account can open and run available test projects that were created by either an Engineer or Administrator.

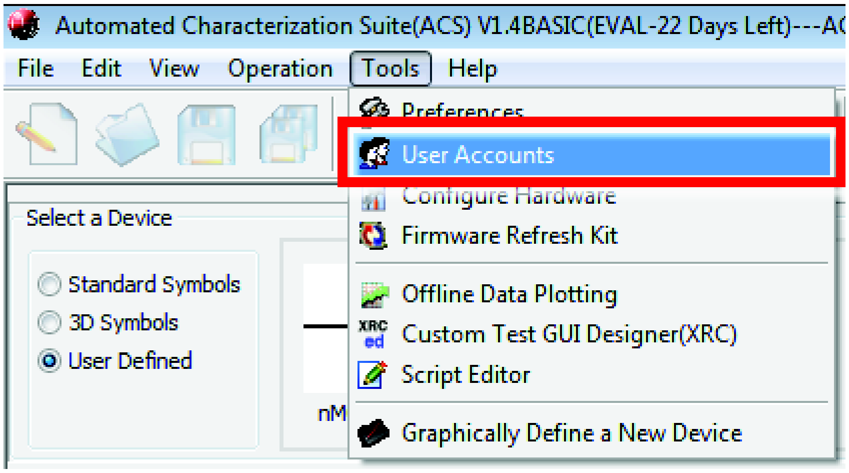
Creating new users

NOTE

Only the administrator account (ACSADMIN) has the necessary rights to add a new Operator or Engineer account.

If you are currently logged in to ACS Basic, you can click on Tools and select User Accounts from the drop-down menu (see next figure).

Figure 9: User Accounts location

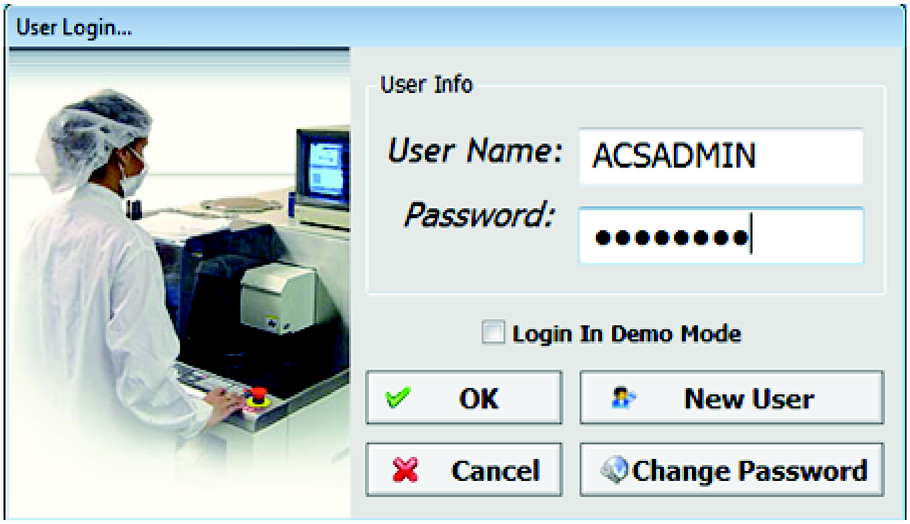


NOTE

Only the administrator account (ACSADMIN) will see the complete list of user accounts. Engineer accounts will only see their own account. Operators do not have access to this feature.

If you are not logged in to ACS Basic, you can click on the **New User** function in the User Login GUI (see next figure).

Figure 10: User Login window



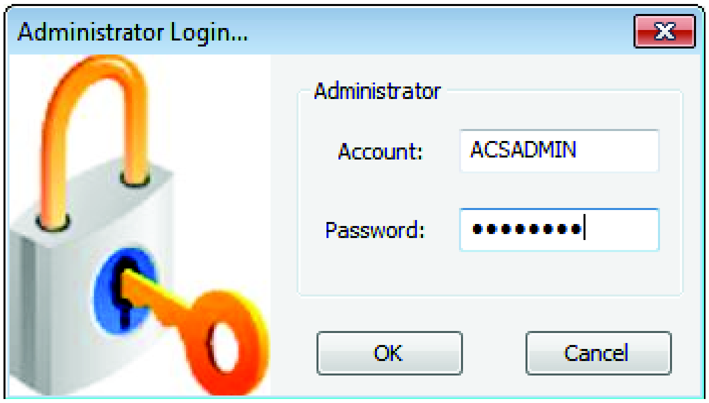
User Name: ACSADMIN

Password: ●●●●●●●●●

□ Login In Demo Mode

- The Administrator Login dialog box opens (see next figure).
- Enter the password for the Administrator (the default Administrator account name is ACSADMIN).

Figure 11: Administrator login dialog box

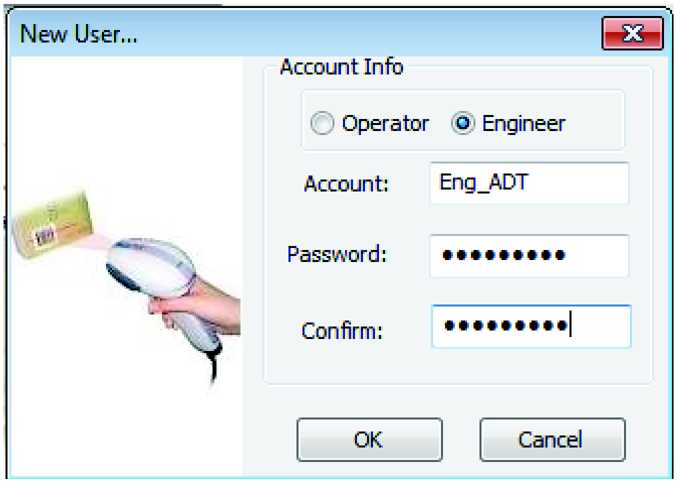


Account: ACSADMIN

Password: ●●●●●●●●●

- Click **OK**, and a New User dialog box opens (see next figure).
- Select the type of account you would like to create: **Operator** or **Engineer**
  a. An Operator account allows you to load and run test projects.
  b. An Engineer account allows you access to all of the ACS Basic functions.
- Enter your **Account** name and **Password**.
- Enter your password again in the **Confirm** field.
- Click **OK**, and the new user account is complete.

Figure 12: New User dialog box



New User...

Account Info

○ Operator ● Engineer

Account: Eng_ADT

Password: •••••••••

Confirm: ••••••••••

Initialize multiple user accounts

This information is for those who need to initialize multiple user accounts in a Windows® environment.

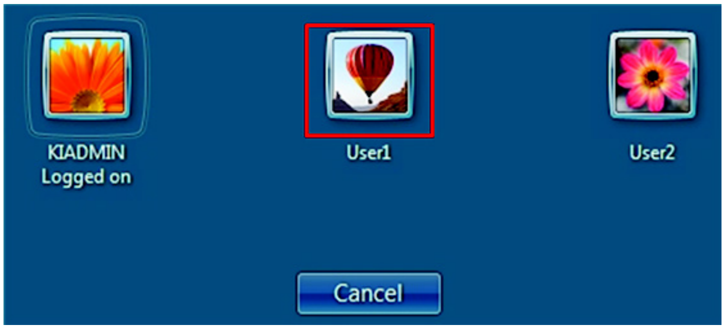
⚠ CAUTION

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

NOTE

This information is for those who have more than one user account when logging in to a Windows environment (see next figure). If you have more than one user account on your computer, make sure that you are logged into the user account where ACS Basic was installed.

Figure 13: Windows User Accounts

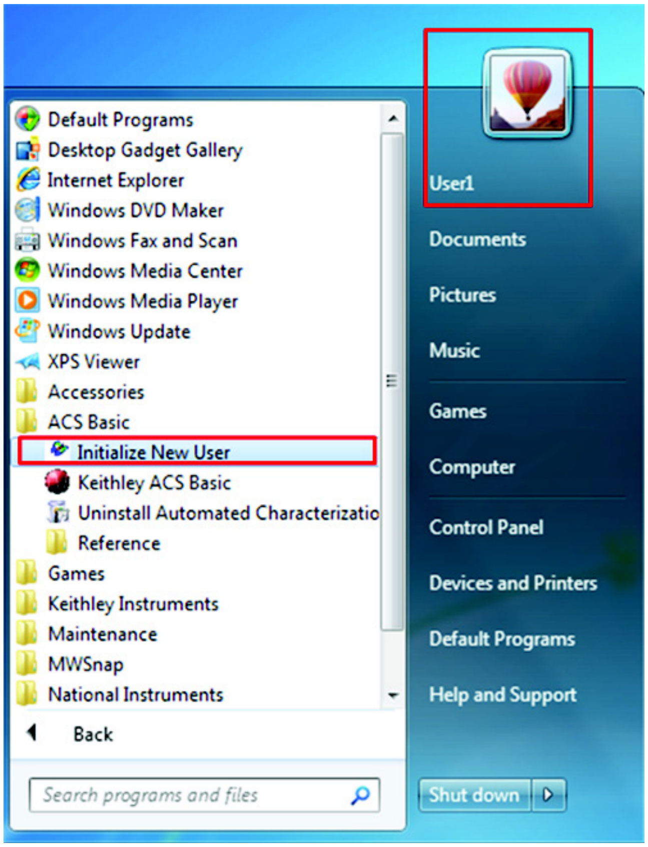


NOTE

If the administrator (for example, Admin1) has already installed ACS Basic software, none of the other users (User1 or User2) need to install the software. However, you do need to initialize the user accounts (User1 and User2) so that they have access to ACS Basic software.

After logging in to a user account (for example, User1), click on your Windows Start function and find ACS Basic in All Programs. Click the Initialize User function (see next figure).

Figure 14: Windows Start Function



After you click the Initialize User function, you must restart your computer.

Once your computer reboots, you can log in to Windows with your initialized account and run ACS Basic software.

Change password

NOTE

Only the administrator account (ACSADMIN) has the necessary rights to change a password for all of the user accounts. An engineer account can change its own password, but not any other accounts. An operator does not have access to this function.

While in the User Accounts window, select the account that will have the password changed.

Click the **Edit** button to open the Change password dialog box. Enter the current password of the account, enter a new password, and then confirm in the appropriate fields.

Click **OK** to complete the process.

Delete user account

NOTE

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

While in the User Accounts window, select the account you want to delete and click the **Delete** button. To complete the process click **Yes** when asked if you want to delete the account.

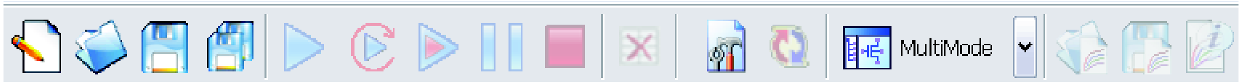
ACS Basic GUI

When you first open ACS Basic, the default test mode that you will see is Single-test mode. ACS Basic has three test modes available:

1. Single-test mode
2. Multi-test mode
3. Trace mode

Operation toolbar

The Operation toolbar is located at the top of the ACS Basic GUI and contains a set of functions that allow you to perform the following tasks (see next figure), New project, Open project, Save project, Save All, Run, Repeat, Append, Pause, Stop, Clear Append, Hardware Configuration, Scan Hardware, and choose a Mode option (this list of tasks corresponds with the icons in the Operation toolbar from left to right).



Project file management icons

New, Open, Save, and **Save All** icons (used in Multi-test mode)(See the following test mode topics for more information: **Single-test mode** (on page 5-4), **Multi-test mode** (on page 5-10), **Trace mode** (on page 5-15).

These functions allow you to create and manage project files in Multi-test mode:

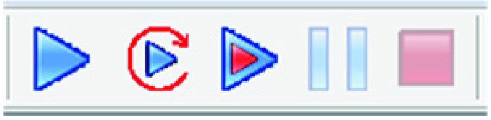
- **New**: creates a new project
- **Open**: opens an existing project
- **Save**: saves all project settings of the configuration navigator
- **Save All**: saves all project settings and test result data

Test control icons

Run, Repeat, Append, Pause, and **Stop** icons (see next figure).

- **Run**: initiates the test procedure
- **Repeat**: continues test execution until Stop is selected
- **Append**: initiates a test procedure and appends data in a new Data tab
- **Pause**: allows you to temporarily stop the test flow (the test will pause right after the current operation is completed; the next test will not execute in multi-test mode.)
- **Stop**: immediately ends the test

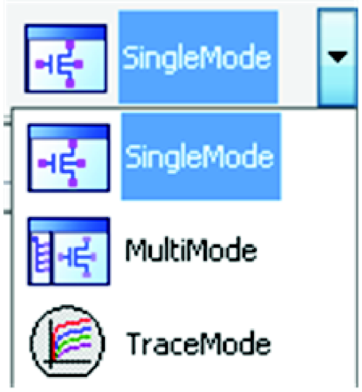
Figure 15: Run, Repeat, Append, Pause, and Stop functions



Multi-mode, Single-mode, and Trace mode

These functions are used to switch between different test modes. You have to click the down-arrow to access the test modes (see next figure).

Figure 16: Multiple-mode, Single-mode, and Trace-mode



- **Single-mode**: allows you to perform a single test on a single device.
- **Multi-mode**: allows you to perform multiple tests on a single device (target application), or multiple tests on multiple devices. It is easier to use external instruments (DMM, switch, etc.) in multi-mode.
- **Trace mode**: allows you to perform interactive operations with a SourceMeter. You can adjust the maximum sweep value and maximum power into the device. You can also easily transition between DC and pulse forcing modes.

⚠ CAUTION

If you switch between the Single-test mode and the Multi-test mode, your test settings and data may be lost. Therefore, it is recommended that you save your test data before you switch modes.

Single-test mode icons

Click the Hardware configuration icon to access the configure hardware utility GUI.

The Scan hardware icon is enabled when the hardware configuration GUI is open. This function scans the hardware and updates the test hardware information

The following functions are active only in the Single-test mode:

- Return to Library view: opens a test module in the device library.
- Export Test: saves a test module to the device library.
- Device View: opens a GUI that contains the device setup and connection information.
- Test View: opens a GUI that contains the test configuration information.

NOTE

The Device view and Test view functions do not appear simultaneously. For example, when in Device view, the Test view function opens; when in Test View the Device View function opens.

ACS Basic GUI icons

Test setup icons

Return to Library View returns to the GUI to add a test module from the library.

Export Test exports the current test module to the library.

Add new Device/Test Module allows you to add a device or an ITM, STM, or PTM test module.

Launch Script Editor allows you to access the editing tool to edit the library.

Project management icons

Move Test Module Up moves the selected test to the next spot in the Configuration Navigator.

Move Test Module Down moves the selected test to the next spot in the Configuration Navigator.

Rename allows you to rename a pattern, subsite, device, or test module.

Delete allows you to delete a selected pattern, subsite, device, or test module.

Copy allows you to make a copy of a pattern, subsite, device, or test module.

Cut allows you to cut a pattern, subsite, device, or test module.

Paste allows you to paste a pattern, subsite, device, or test module in the Configuration Navigator. It only appears after Copy or Cut has been selected.

NOTE

If you copy a test module (for example, an ITM), and past it under another device where the attributes and settings do not match the original device, you will lose the test module's information. Please ensure that you are copying and pasting between devices that contain the same attributes and settings.

Log message window

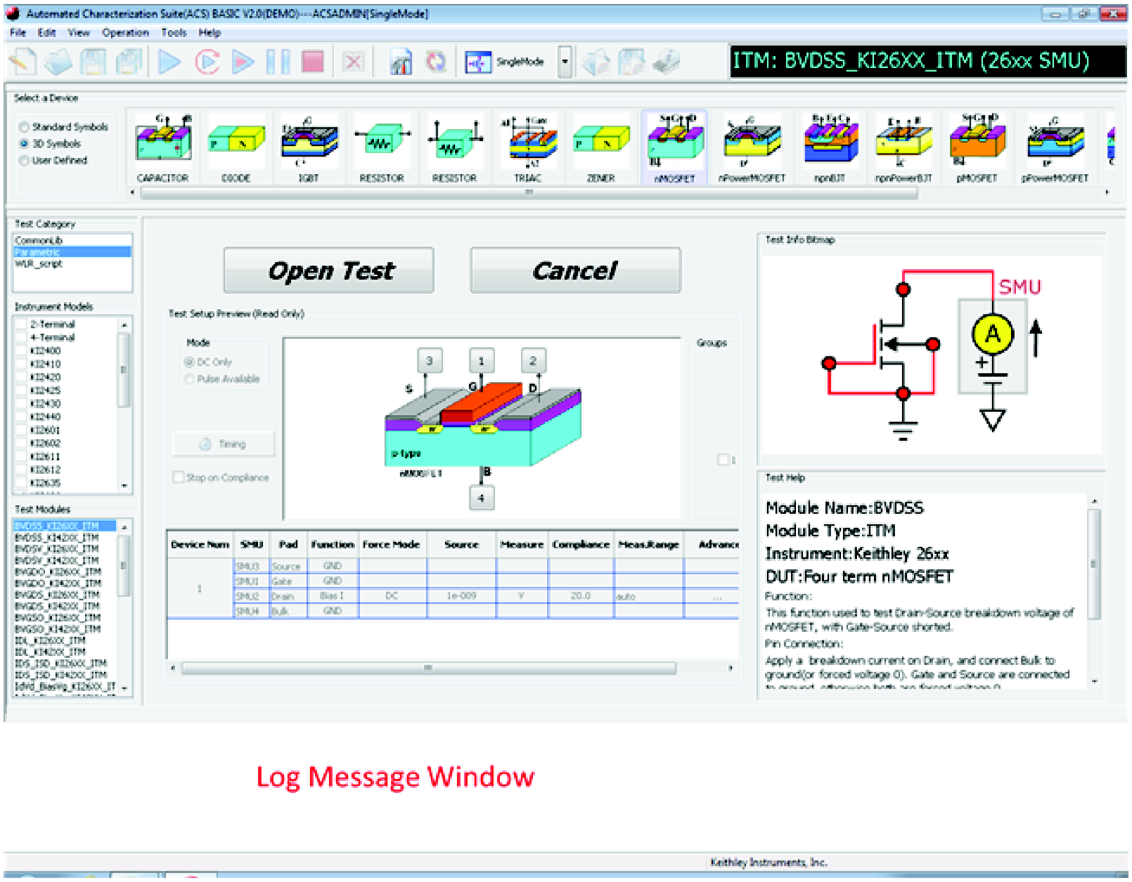
When you run a python test module (PTM) or script test module (STM), ACS Basic will log the commands, errors, warning messages, and some test results in the log message window.

NOTE

The log window is not used for interactive test modules (ITMs). That is, the ITM test will not display in the log message window (see next figure).

The log message window is located at the bottom of the ACS Basic GUI (see next figure). The log message displays in the log window in real time.

Figure 17: Log message window



Automated Characterization Suite(ACS) BASIC V2.0(DEMO)---ACSADMIN[SingleMode]

ITM: BVDSS_KI26XX_ITM (26xx SMU)

Device Num	SMU	Pad	Function	Force Mode	Source	Measure	Compliance	Meas.Range	Advance
1	SMU3	Source	GND						
	SMU1	Gate	GND						
	SMU2	Drain	Bias I	DC	1e-009	V	20.0	auto	...
	SMU4	Bulk	GND						

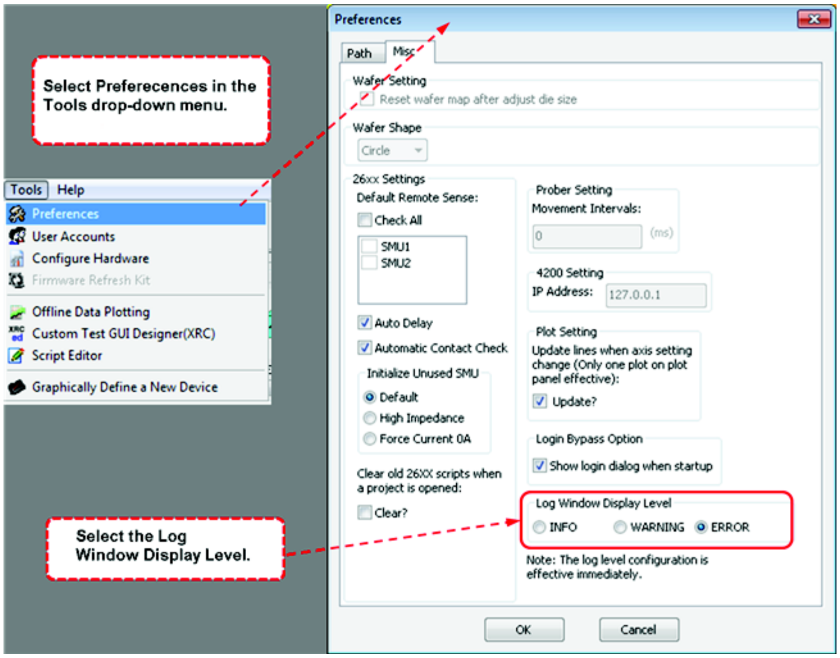
Module Name:BVDSS
Module Type:ITM
Instrument:Keithley 26xx
DUT:Four term nMOSFET
Function:
This function used to test Drain-Source breakdown voltage of nMOSFET, with Gate-Source shorted.
Pin Connection:
Apply a breakdown current on Drain, and connect Bulk to ground(or forced voltage 0). Gate and Source are connected

Log Message Window

In the View menu, you can choose to show or hide the log window. By default, it is hidden.

You can also select the log level to display in the log message window: **Info**, **Warning**, or **Error** in Tools > Preferences settings (see next figure).

Figure 18: Preference setting for Log Level



Select Prefereences in the Tools drop-down menu.

Select the Log Window Display Level.

Preferences

Wafer Setting

☑ Reset wafer map after adjust die size

Wafer Shape: Circle

26xx Settings

Default Remote Sense:

□ Check All

□ SMU1

□ SMU2

☑ Auto Delay

☑ Automatic Contact Check

Initialize Unused SMU

● Default

○ High Impedance

○ Force Current 0A

Clear old 26XX scripts when a project is opened:

□ Clear?

Prober Setting

Movement Intervals: 0 (ms)

4200 Setting

IP Address: 127.0.0.1

Plot Setting

Update lines when axis setting change (Only one plot on plot panel effective):

☑ Update?

Login Bypass Option

☑ Show login dialog when startup

Log Window Display Level

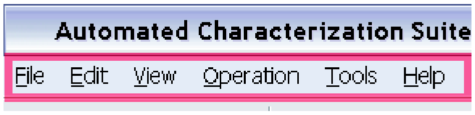
○ INFO ○ WARNING ● ERROR

Note: The log level configuration is effective immediately.

Menu toolbar elements

The menu toolbar is located at the top of the ACS Basic GUI (see next figure).

Figure 19: Menu toolbar location



File menu

The **New**, **Open**, **Save**, **Save as Template**, and **Save All** functions are the same as previously described. Refer to the Project file management functions topic. These functions are only enabled in multi-mode. Only the EXIT button can be used in single-mode.

Edit menu

The **Copy**, **Cut**, and **Delete** functions are the same as previously described. Refer to the Project file management functions topic for more information. Also, the Edit menu is disabled in Single-mode.

View menu

Only the **Show Log Window** item can be accessed in ACS Basic

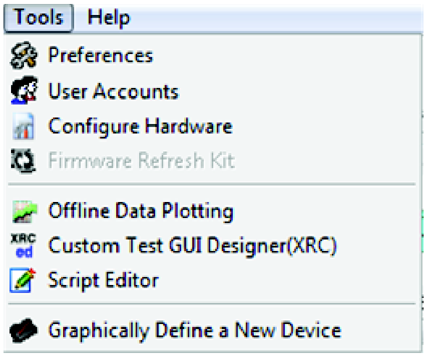
Operation menu

The **Run**, **Repeat**, **Append**, **Pause**, **Stop**, and **Clear Append** functions are the same as previously described. Refer to the Test control functions topic, if you need more information. Notice that Repeat, Pause, and Stop are disabled when you are in single-mode.

Tools menu

There are several functions available in the Tools menu (see next figure).

Figure 20: Tools menu



Preferences: set advanced preferences. Refer to the Advanced Setting Preferences for details.

User Accounts: view the User Admin dialog box for user accounts information (see User accounts security for details).

Configure Hardware: access the hardware configuration utility. Refer to the Hardware Configuration topic for details.

Firmware Refresh Kit: refresh the Series 2600 or Series 2600B firmware. This utility verifies or performs updates on the firmware on any Series 2600 and 2600B instruments connected to the ACS Basic. Any updates will be made using firmware files stored locally on your computer or network.

NOTE

The latest firmware files are available for download on www.keithley.com.

Offline Data Plotting: access the Data Plot Tool. Refer to the Data plot section for details.

Custom Test GUI Designer (XRC): access the GUI builder. Refer to Create a TSP GUI file for details.

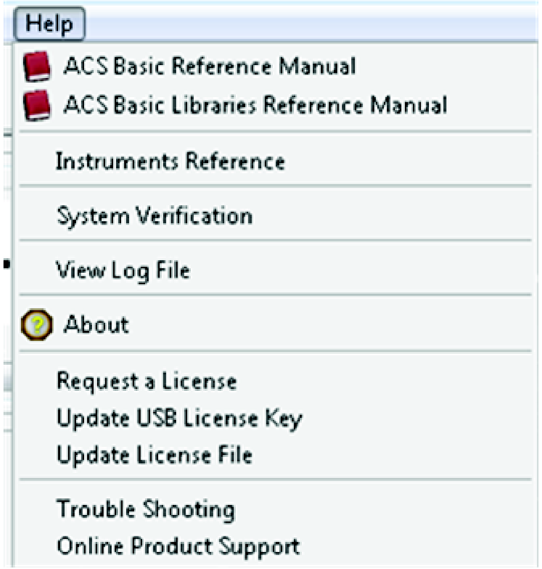
Script Editor: access the Script Editor Tool. Refer to the Script Editor tool for details.

Graphically Define a New Device: access the device editor tool. For more information on the Device Editor refer to the Graphically define a new device for details.

Help menu

You have many items available in the Help menu for ACS Basic (see next figure).

Figure 21: Help menu



ACS Basic Reference Manual: provides detailed comprehensive information about the software features

ACS Basic Libraries Reference Manual: contains LPT library programming commands and device library commands

Instrument Reference: click to find a list of manuals that you can use with ACS Basic supported instruments, including Series 2400, 2600B, 3700A, Model 4200-SCS, and Models 707A/707B and 708A/708B .

NOTE

All of the products that Keithley Instruments manufactures are located on our website. You can search for a specific product and find updated manuals anytime by clicking this link: www.keithley.com.

System Verification: click to open the verification project in multiple-test mode.

View Log File: click to open the saved log file.

About: click to learn more About ACS Basic, such as the build date and type, plus other general information.

Request a License: click to receive a dialog box that gives detailed information on how to request a license.

Update USB License Key: click to open the update license Key dialog. Refer to the Update a temporary USB key for instructions.

Update License File: if you request a license file, you will receive instructions on how to update your new license file; additionally, when you click this option, it will take you to the area where your license key is located.

Trouble Shooting: shows system information, such as hardware and software information, including information about your computer and your version of ACS Basic. You can save this information if you need to request a license, or send it to a Keithley Instruments applications engineer for troubleshooting.

Online Product support: click to open Keithley Instruments' online product support website: www.keithley.com/support. Here you will be able to find all the information that you want on Keithley's products.

Section 4

ACS Basic hardware configuration

In this section:

Introduction

The Automated Characterization Suite (ACS) Basic hardware configuration utility is used to manage the hardware configuration of the following Keithley instruments:

- Series 2400 SourceMeter® Instruments
- Series 2600 SourceMeter Instruments
- Series 2600B SourceMeter Instruments
- Model 707A/707B and 708A/708B Switching Matrices
- Series 2400 SourceMeter Instruments
- Series 3700A System Switch/Multimeter
- External GPIB instruments and all external system components supported by the ACS Basic software tools (refer to Connect to external instruments topic for more details).

Using the hardware configuration utility, these instruments can be easily added, configured, and removed from the system configuration.

NOTE

If you make changes to the hardware (for example, remove one instrument, change the 707B work mode, change node information, etc.) while ACS Basic is running, you must also do the following tasks:
1. Click the Scan Hardware function in the toolbar to re-scan hardware.
2. Click the Save function to update the changes.
3. Close and restart ACS Basic for the changes to take effect.
4. Check the test setup to ensure the new hardware configuration is correct.

If the above tasks are not completed, the results from your tests may not be accurate.

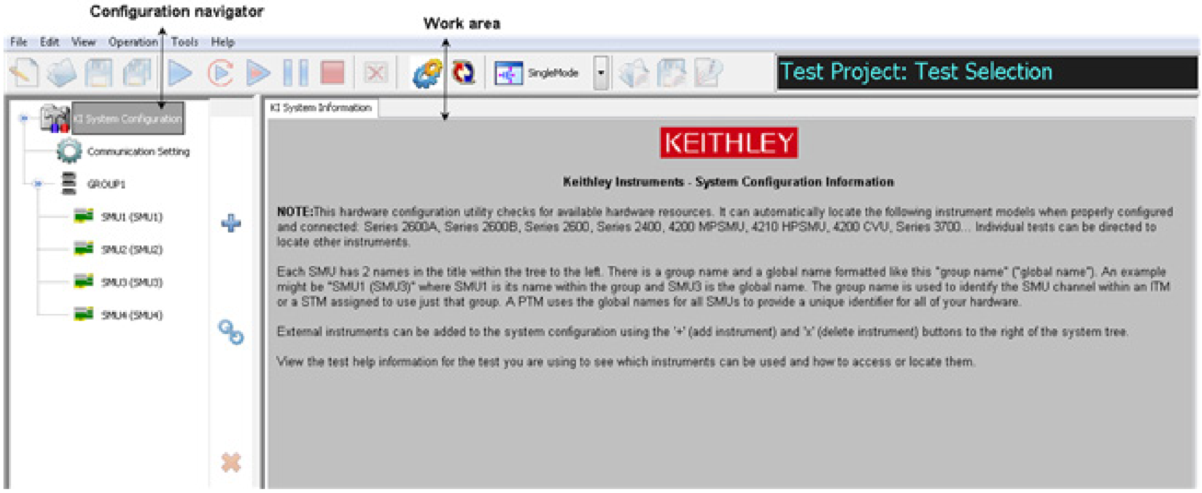
Configure hardware function

The Configure Hardware function is used to get information while scanning the hardware instruments.

To start a scan, click the Configure Hardware option from the Tools menu or click the configure hardware icon on the toolbar.

The interface of the hardware configuration utility has two parts: the left panel is the **configuration navigator**; the right panel is the **work area** (see next figure).

Figure 22: Hardware configuration interface



Keithley Instruments - System Configuration Information

NOTE: This hardware configuration utility checks for available hardware resources. It can automatically locate the following instrument models when properly configured and connected: Series 2600A, Series 2600B, Series 2600, Series 2400, 4200 MPSMU, 4210 HPSMU, 4200 CVU, Series 3700... Individual tests can be directed to locate other instruments.

Each SMU has 2 names in the title within the tree to the left. There is a group name and a global name formatted like this "group name" ("global name"). An example might be "SMU1 (SMU3)" where SMU1 is its name within the group and SMU3 is the global name. The group name is used to identify the SMU channel within an ITM or a STM assigned to use just that group. A PTM uses the global names for all SMUs to provide a unique identifier for all of your hardware.

External instruments can be added to the system configuration using the '+' (add instrument) and 'x' (delete instrument) buttons to the right of the system tree.

View the test help information for the test you are using to see which instruments can be used and how to access or locate them.

The configuration navigator provides a tree view of all the instruments and equipment present in the ACS Basic hardware configuration. The tree branches can be expanded and minimized by clicking on the arrows to the left of each branch.

The work area shows each instrument in the system configuration and its associated properties. Click an instrument in the configuration navigator and the associated properties display in the work area.

Supported communication interfaces

ACS Basic software supports the following communication interfaces:

- GPIB (general purpose interface bus)
- Ethernet (also referred to as a LAN)

ACS Basic 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 together. Refer to the Connect to multiple instruments topic for more information.

The following table indicates how instruments communicate with ACS Basic (see next table).

Instrument	**Platform**	**Communication interface**
2400	Desktop/laptop computer	GPIB
2600	Desktop/laptop computer or 4200	GPIB, TSP subordinate
2600B	Desktop/laptop computer or 4200	GPIB, Ethernet, TSP subordinate
4200	4200 only	Internal
3700A	Desktop/laptop computer or 4200	GPIB, Ethernet, TSP subordinate
Other external instruments	Desktop/laptop computer or 4200	GPIB

Connect to a single instrument

NOTE

Information about connecting to the Model 4200-SCS is not in this section. Refer to the topic Connect to Model 4200-SCS for more information

When using ACS Basic to control hardware, you must select and configure a communications interface. If the hardware is configured with a communications interface, when ACS Basic starts it will automatically scan and find the hardware.

Select an interface

Make sure you choose an interface that is consistent with your instrument's connection: GPIB or Ethernet. Some instruments allow multiple communication interfaces to be enabled, however you can only communicate with one interface at a time using ACS Basic. For more information about communication interfaces, refer to Supported communication interfaces.

NOTE

See the instrument's manual for more information on the GPIB and Ethernet communications interfaces.

Configure the interface

GPIB interface

ACS Basic automatically scans the GPIB instruments during start.

Supported interfaces

ACS Basic software supports these GPIB interfaces:

- KUSB-488
- KUSB-488A
- KUSB-488B
- KPCI-488
- KPCI-488A
- KPCI-488LP
- KPCI-488LPA
- NI USB-488
- NI PCI-488

GPIB card installation

It is recommended that you install the GPIB card and its associated driver prior to installing ACS Basic.

NOTE

For more information on the installation of a GPIB card driver, refer to the GPIB manufacturer's interface documentation for installation instructions.

GPIB Driver Compatibility

General

Almost all GPIB interface drivers can be installed before or after ACS Basic has been installed. The order of installation will not affect the functionality of the software. NI and CEC GPIB interface drivers are exceptions to this rule, see below.

Keithley Instruments

Uninstalling or reinstalling KITE or Keithley GPIB interface drivers (on a system that has ACS Basic already installed) may cause ACS Basic to function incorrectly due to .dll errors. To resolve this issue please reinstall ACS Basic every time KITE or Keithley GPIB interface drivers are uninstalled or reinstalled.

National Instruments

Uninstalling or reinstalling NI GPIB interface drivers or NI VISA drivers (on a system that has ACS Basic already installed) may cause ACS Basic to function incorrectly due to .dll errors. To resolve this issue please reinstall ACS Basic every time NI GPIB interface drivers or NI VISA drivers are uninstalled or reinstalled.

CEC

ACS Basic supports CEC488 GPIB drivers 8.2 or above. Please upgrade older drivers to ensure compatibility. Uninstalling or reinstalling CEC GPIB interface drivers (on a system that has ACS Basic already installed) may cause ACS Basic to function incorrectly due to .dll errors. To resolve this issue please reinstall ACS Basic every time CEC GPIB interface drivers are uninstalled or reinstalled.

Check the GPIB communication

Prior to starting ACS Basic, it is recommended that you verify that the GPIB card and driver are installed correctly and are properly functioning in your operating system. For the Keithley/CEC GPIB interfaces, you can use the following diagnostic utilities: "KI-488 Configuration Utility" and the "KI-488 Diagnostic Tool."

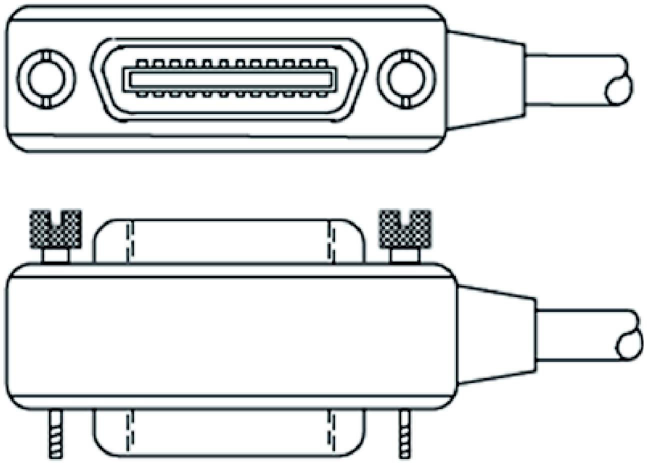
NOTE

For more information on verification instructions on the GPIB communication interface card, refer to the GPIB manufacturer's documentation.

Connect instruments with GPIB

To connect the instruments to the GPIB bus, use an IEEE-488 GPIB cable (see next figure).

Figure 23: Standard IEEE-488 connectors



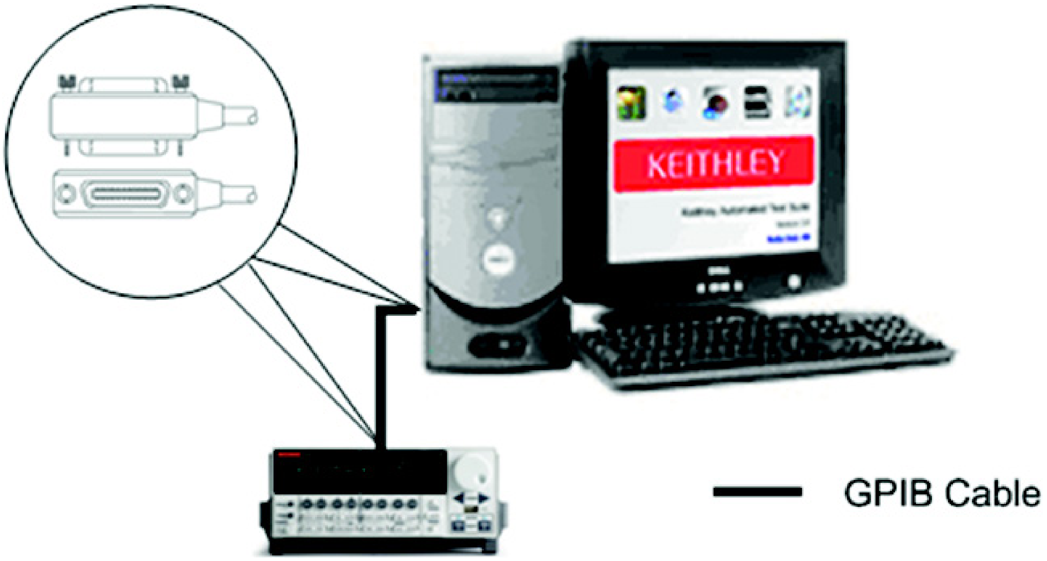
Establish the instrument's GPIB address and make sure that the instruments are configured for a GPIB connection and has a valid, unique address. For example, the Series 2600B ships from the factory with a GPIB primary address of 26. If the GPIB interface is enabled, it momentarily displays the primary address on power-up. You can set the address to a value from 0 to 30, but do not assign the same address to another instrument or to a controller that is on the same GPIB bus.

NOTE

Controller GPIB addresses are usually 0 or 21.

The next figure shows a single instrument connected to a computer over GPIB.

Figure 24: Single instrument GPIB configuration



Ethernet interface

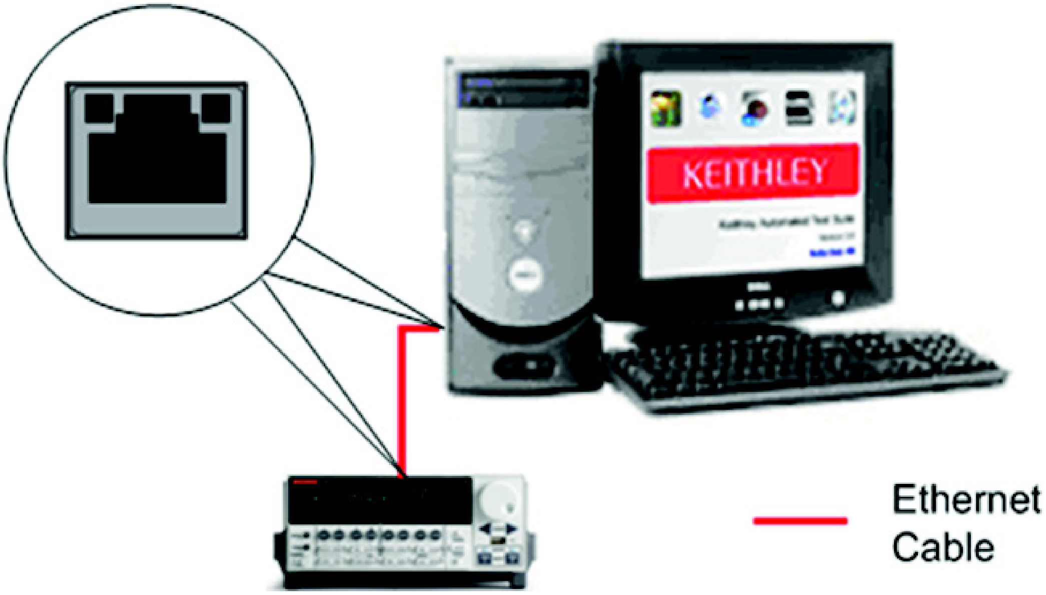
ACS Basic must be manually configured to scan for instruments that are connected to an Ethernet cable.

Ethernet cable connection

The Series 2600B instruments include two cables (part number: CA-180-3A). Use one cable for TSP-Link and use the other cable for the Ethernet connection (for a direct instrument to computer connection).

- Insert the Ethernet cable into the Ethernet port located on the back of the instrument.
- Insert the cable into the Ethernet port located on the back of the host computer.

Figure 25: Ethernet connection



- Assign an IP address to the instrument.

NOTE

For more information on how to configure an Ethernet connection, refer to the instrument's manual.

Manually configure Ethernet hardware

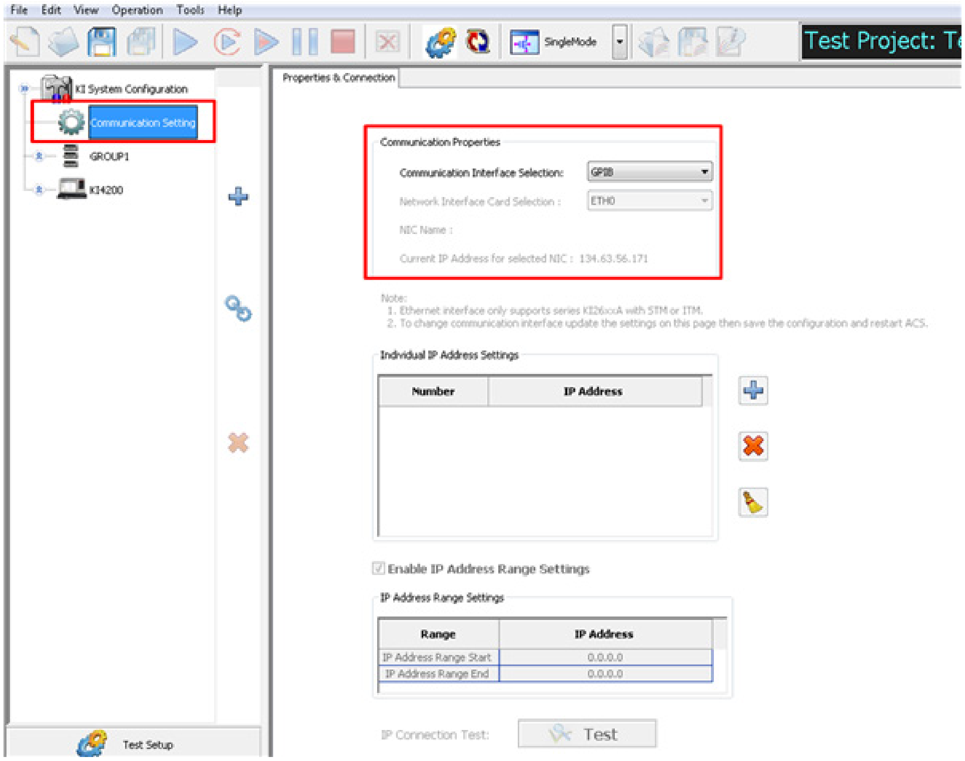
To manually configure ACS Basic to scan for Ethernet instruments, first start ACS Basic and change the settings within the hardware configuration utility.

1. Open the hardware configuration utility
2. Click the Configure Hardware option in the drop-down list in the Tools menu or click the configure hardware icon on the toolbar.

Each instrument that uses an Ethernet connection (corporate or private) requires a unique IP address. In the hardware configuration interface, you can add the IP address of these instruments.

1. Click on the Communication Setting to display the Properties and Connection GUI (see next figure).
2. Set the communication interface selection to Ethernet (see next figure).

Figure 26: Communication interface of hardware configuration



Communication Properties

Communication Interface Selection: GPIB

Network Interface Card Selection : ETH0

NIC Name :

Current IP Address for selected NIC : 134.63.56.171

Note:
1. Ethernet interface only supports series KI26xxA with STM or ITM.
2. To change communication interface update the settings on this page then save the configuration and restart ACS.

Individual IP Address Settings

Number	IP Address

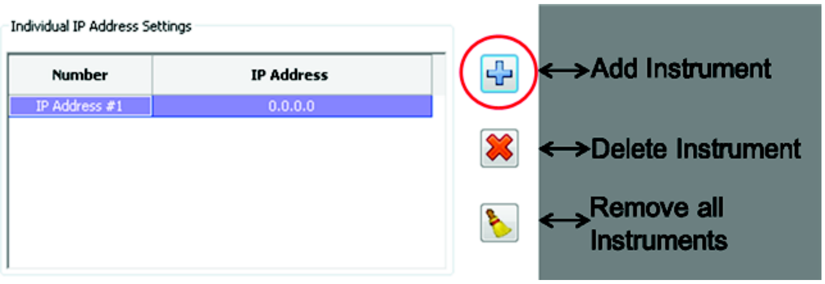
☑ Enable IP Address Range Settings

IP Address Range Settings

Range	IP Address
IP Address Range Start	0.0.0.0
IP Address Range End	0.0.0.0

3. Add an instrument using the add button (see next figure).
4. Click the IP address to enter a unique address for the new instrument.

Figure 27: Add IP address for new instruments


Individual IP Address Settings

Number	IP Address
IP Address #1	0.0.0.0

If you want to delete an instrument, click the Delete instrument icon ✖. If you want to remove all of the instruments, click the Remove all instruments icon (see previous figure).

When the Enable IP Address Range Settings is selected (located below the IP Address setting), you can set the range of addresses scanned. ACS Basic scans all of the instruments within the specified IP address range during startup.

Use the IP Connection Test function to ensure that Ethernet communication is established.

NOTE

For trouble shooting an Ethernet connection issue with an instrument, refer to the instrument's specific manual documentation.

1. Save the hardware configuration.
2. Close and restart ACS Basic for changes to take effect.

Connect to multiple instruments

ACS Basic supports the following instruments in a daisy chain configuration through a GPIB cable, in one group using TSP-Link, or both:

- Multiple 2400 SMUs
- Multiple TSP instruments (includes Series 2600 and 2600B instruments and Model 3706A)
- Multiple external GPIB instruments

Connect multiple 2400 SMUs

Series 2400 instruments can only be connected with a GPIB cable (see next figures):

1. Set a unique GPIB address for each instrument.
2. Connect the GPIB cables to each instrument, daisy chaining the connectors.
3. Start ACS Basic.
4. Open the hardware configuration GUI, and set the instrument properties: Rear Jack and Beeper.

NOTE

For details on how to configure the GPIB address of the Series 2400 SourceMeter instruments, refer to the Series 2400 SourceMeter manual documentation.

Figure 28: Multiple Model 2400 SMU connections

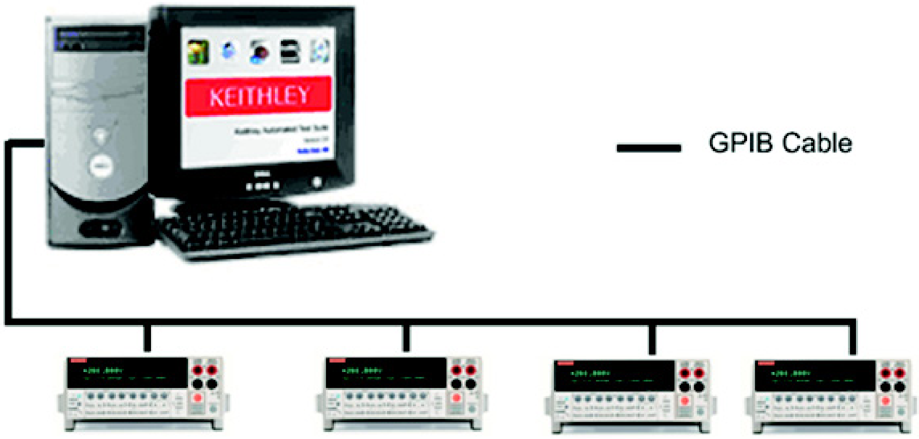
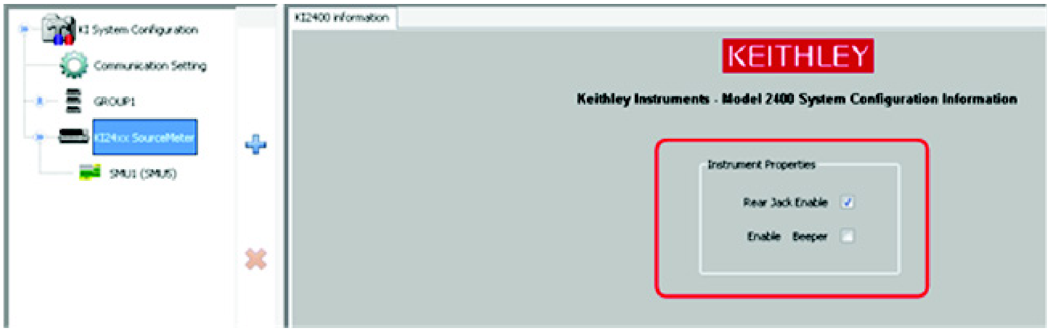


Figure 29: Model 2400 configuration GUI



Keithley Instruments - Model 2400 System Configuration Information

Instrument Properties

Rear Jack Enable ☑

Enable Beeper ☐

Multiple TSP instruments

Multiple TSP instruments configured in one group or several groups are supported by ACS Basic.

Using TSP-Link

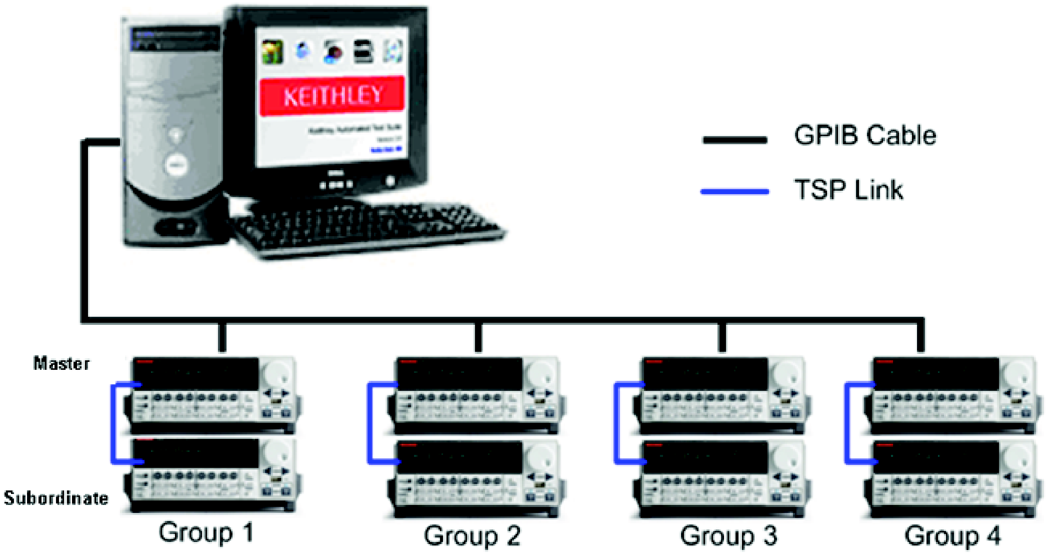
You can use TSP-Link to connect subordinate instruments to the master, while the master instrument is connected to the computer through a GPIB or Ethernet cable. ACS Basic treats all instruments connected with TSP-Link to a master as a single group (refer to Check the hardware configuration). To setup a group of instruments with TSP-Link do the following tasks:

- Connect one instrument to the GPIB or Ethernet interface.
- Connect other TSP-enabled instruments using TSP-Link, daisy chaining the TSP-Link cables between the instruments.
- Assign a unique node number to each instrument.
  - If using GPIB, the master instrument should be node 1; if using Ethernet, the instrument that is connected directly to the PC should be node 1.
  - Start ACS Basic and it will scan the hardware.

GPIB multiple interfaces

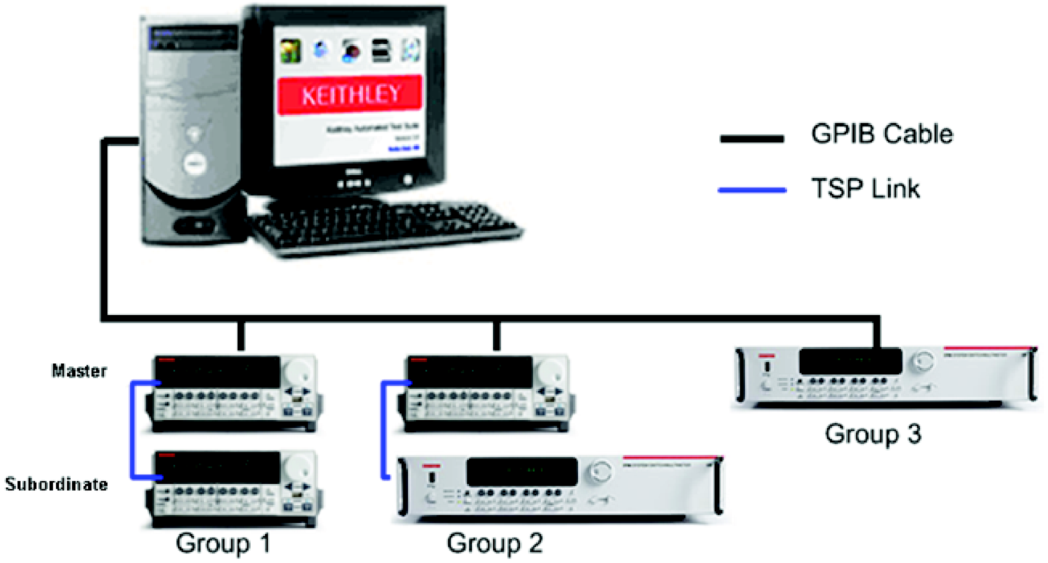
A group is connected through a GPIB cable (see next figure) from the controller to a Series 2600/2600B instrument which serves as the master node. The other instruments connected with TSP-Link are subordinates of the master.

Figure 30: Multiple Model 2600 SMU connections



ACS Basic supports TSP groups containing different instruments, as shown the next figure which depicts the Series 2600/2600B and Series 3700A instruments in multiple groups.

Figure 31: Multiple TSP instrument connections



Ethernet multiple interfaces

If you need to use multiple TSP-Link instrument groups connected by Ethernet, you can use an Ethernet hub or a switch. Refer to the next two figures for Ethernet configuration connections.

- Connect the computer to the Ethernet hub or switch.
- Establish a unique IP address for the group master instrument.
- Connect all the instruments to the Ethernet hub or switch using Ethernet cables.
- Start ACS Basic and use the hardware configuration utility to add each IP address to the hardware configuration (refer to Ethernet cable connection for more information).

NOTE

For details on an Ethernet connection between a computer and master instrument, and the TSP-Link connection between Series 2600B instruments, refer to the Series 2600B System SourceMeter manuals.

Figure 32: Ethernet configuration #1

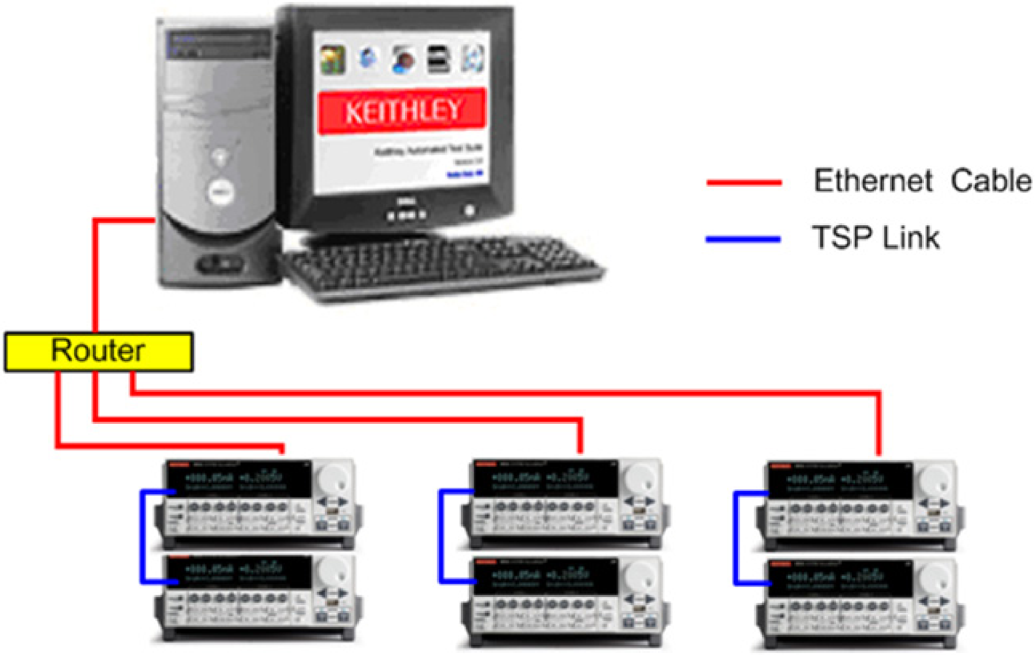
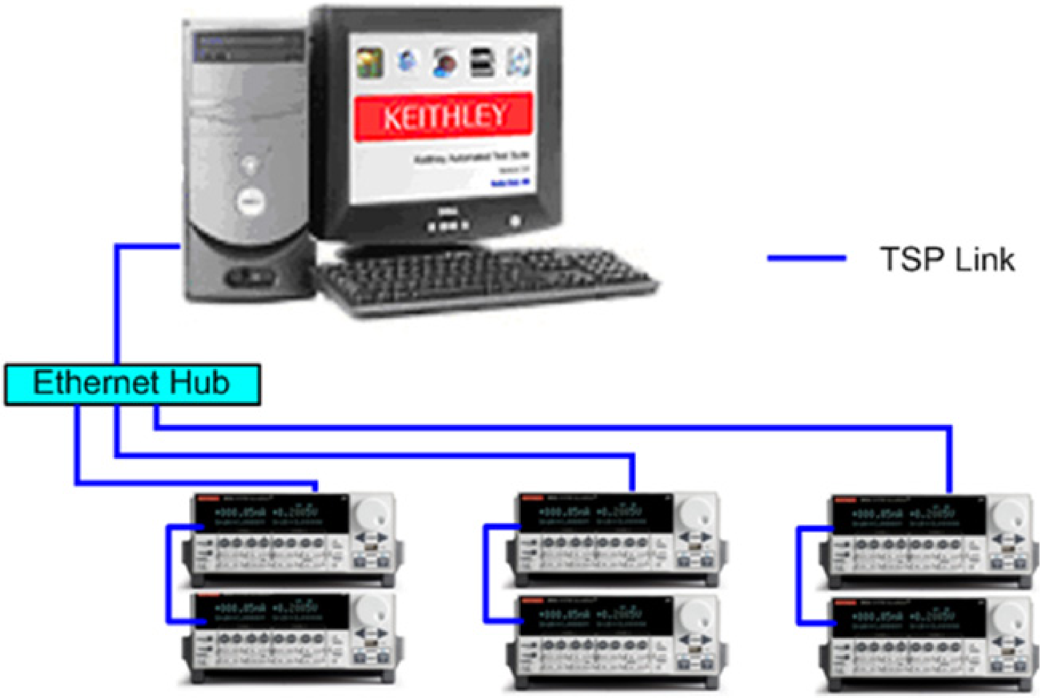
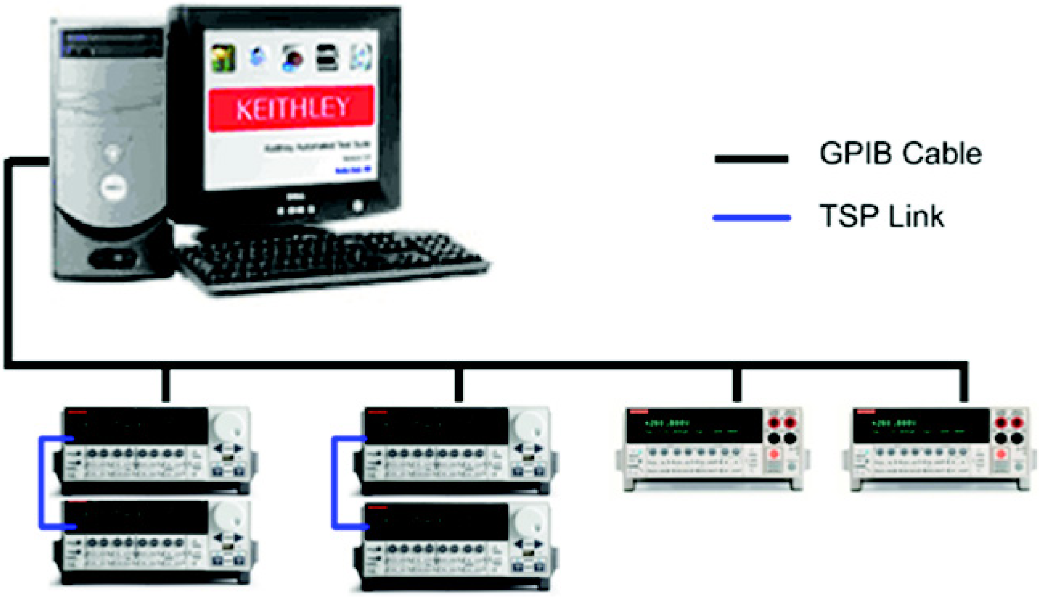


Figure 33: Ethernet configuration #2



The next figure shows ACS Basic software installed on a computer and connected by GPIB cables to Series 2600B and Series 2400 instruments.

Figure 34: Instrument connections configuration



Connect to Model 4200-SCS SMUs

Instruments within Model 4200-SCS

ACS Basic will automatically scan the Model 4200-SCS hardware if ACS Basic is installed on the Model 4200 and as long as the mode of communication is configured for Ethernet (see next figure). Also, ACS Basic can control all of the hardware that is supported by the Model 4200-SCS.

Model 4200-SCS and KCON

Connect the 4200-SCS to the ACS Basic computer through a network connection. Record the IP address of the Model 4200 for reference.

NOTE

Before configuring a Model 4200 ITM, check to ensure that the Ethernet link is working. If you are running ACS Basic on a local Model 4200 using the default 4200 IP address, your hardware configuration is complete.

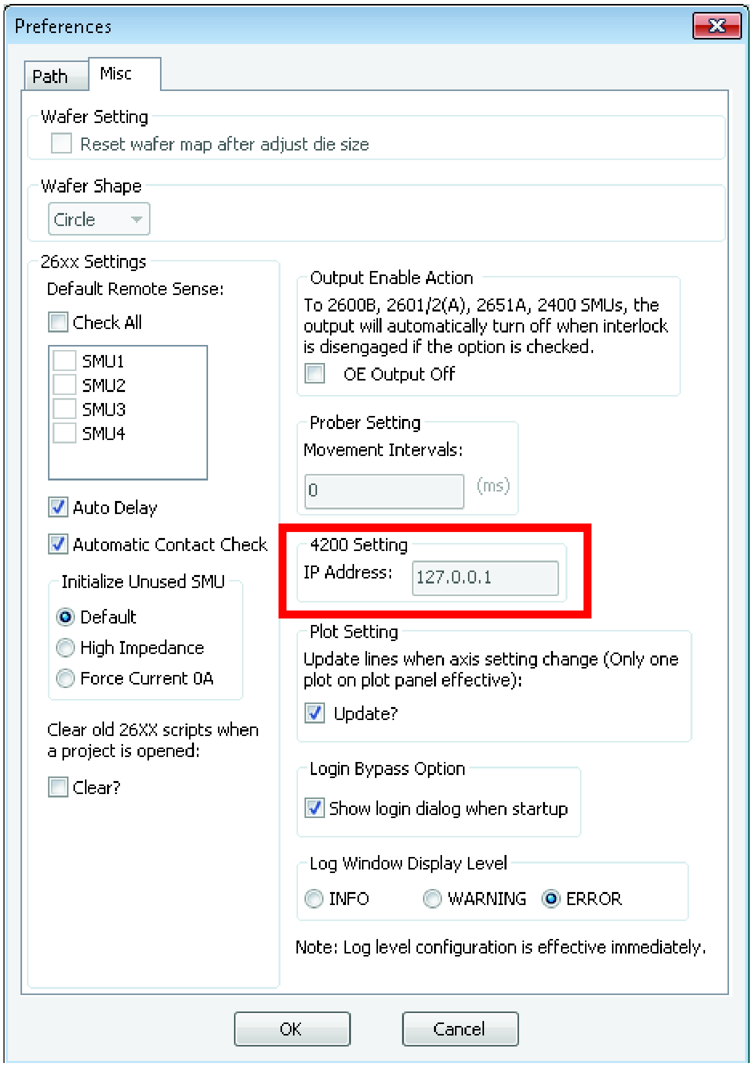
Set up the Model 4200-SCS in the ACS Basic preference menu:

On the toolbar, click on **Tools**, and from the drop-down list select **Preferences** ; the Preferences dialog box opens.

a. Enter the Model 4200 IP address in the IP Address edit box (see next figure).

b. Click **OK**.

Figure 35: Set Model 4200-SCS IP address



Preferences

Wafer Setting

□ Reset wafer map after adjust die size

Wafer Shape

Circle

26xx Settings

Default Remote Sense:

□ Check All

□ SMU1
□ SMU2
□ SMU3
□ SMU4

☑ Auto Delay

☑ Automatic Contact Check

Initialize Unused SMU

● Default
○ High Impedance
○ Force Current 0A

Clear old 26XX scripts when a project is opened:

□ Clear?

Output Enable Action

To 2600B, 2601/2(A), 2651A, 2400 SMUs, the output will automatically turn off when interlock is disengaged if the option is checked.

□ OE Output Off

Prober Setting

Movement Intervals:

0 (ms)

4200 Setting

IP Address: 127.0.0.1

Plot Setting

Update lines when axis setting change (Only one plot on plot panel effective):

☑ Update?

Login Bypass Option

☑ Show login dialog when startup

Log Window Display Level

○ INFO ○ WARNING ● ERROR

Note: Log level configuration is effective immediately.

Close and restart ACS Basic.

Check the hardware configuration:

Select the **Configure Hardware** option from the Tools drop-down list.

a. If the Model 4200 is not displayed in the navigation tree, click the **Scan hardware** icon on the toolbar. The system queries the instruments and then opens the updated configuration information in the configuration navigator.

b. Click the **Save** icon .

c. Close and restart ACS Basic after the new configuration is saved.

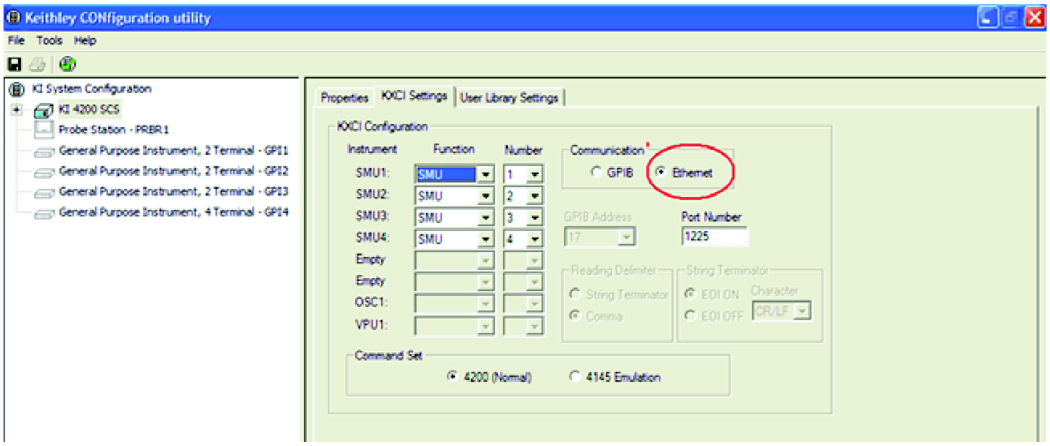
Set the Model 4200-SCS communication mode using Keithley Configuration Utility (KCON) software:

- Open the KCON application on the Model 4200-SCS. The mode of communication with the Model 4200 should be set to Ethernet. You can verify the communication mode by opening KCON and open the KXCI Settings tab (see next figure). Also, the communication mode is visible in the KXCI message area.

NOTE

The Model 4200 can only be used in Ethernet mode with ACS Basic.

Figure 36: Set Model 4200-SCS communication mode



KI System Configuration
- KI 4200 SCS
- Probe Station - PRBR1
- General Purpose Instrument, 2 Terminal - GPI1
- General Purpose Instrument, 2 Terminal - GPI2
- General Purpose Instrument, 2 Terminal - GPI3
- General Purpose Instrument, 4 Terminal - GPI4

KXCI Configuration

Instrument	Function	Number
SMU1:	SMU	1
SMU2:	SMU	2
SMU3:	SMU	3
SMU4:	SMU	4
Empty		
Empty		
OSC1:		
VPU1:		

Communication: ○ GPIB ● Ethernet

GPIB Address: 17

Port Number: 1225

Reading Delimiter: ○ String Terminator ● Comma

String Terminator: ● EOI ON ○ EOI OFF Character: CR/LF

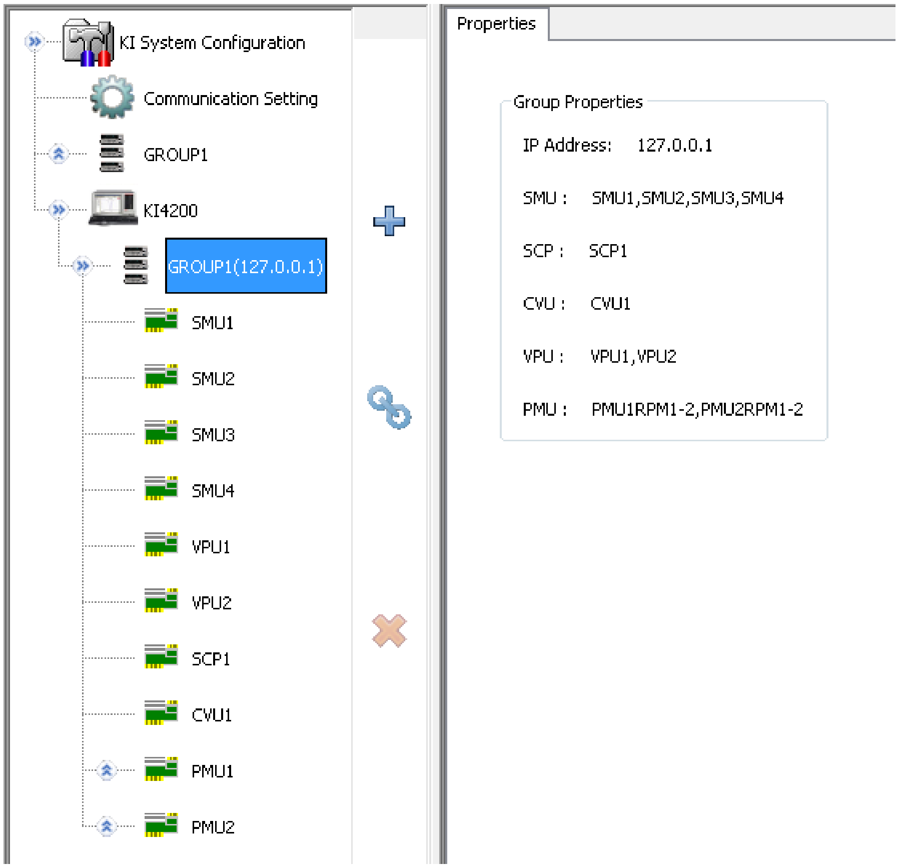
Command Set: ● 4200 (Normal) ○ 4145 Emulation

- Start ACS Basic and it scans the hardware using KCON.
- Check the scanned information in the ACS Basic hardware configuration navigator.

Model 4200 System properties

Click on the KI 4200-SCS icon in the configuration navigator and the properties are displayed in the work area (see next figure).

Figure 37: Model 4200 group properties



Group Properties

IP Address: 127.0.0.1

SMU : SMU1,SMU2,SMU3,SMU4

SCP : SCP1

CVU : CVU1

VPU : VPU1,VPU2

PMU : PMU1RPM1-2,PMU2RPM1-2

Connect to external instruments

ACS Basic can communicate with other external instruments using the hardware configuration utility. However, all external instruments can only be controlled over GPIB.

NOTE

External instruments are not automatically scanned when ACS Basic is started and, they cannot be scanned if you use the scan hardware function. This means that all external instruments must be configured manually on the Hardware Utility Configuration GUI.

The supported external instrument categories are:

- Switch Matrix
- Capacitance Meter
- General Purpose Test Instrument

All supported external instrumentation and equipment are run by Python Test Modules (PTMs)(see Configure a PTM for more information).

The next table lists the supported external instruments.

Category	Instrument
Switch matrix[1]	Model 707/707A (or 707B in DDC mode) Switching Matrix
	Model 708/708A (or 708B in DDC mode) Switching Matrix
Capacitance Meter[2]	Model 4284/4980 CMTR
General purpose test instrument[3]	Other source meters (such as Model 237)
	Any IEEE-488 controlled instrument or equipment

[1] ACS Basic supports the Keithley Instruments Model 707/707A/707B and 708/708A/708B Switch Matrices. The Model 708/708A/708B accepts a single matrix card. The Model 707/707A/707B accepts up to six matrix cards. Also, models 707B and 708B have two working modes: DDC (same as 707A/708A) and ICL. In ICL mode, these instruments can be connected using Ethernet or TSP-Link.

[2] Up to four supported capacitance meters can be added to the system configuration.

[3] ACS Basic supports up to 16 general purpose test instruments (GPIs). Two-terminal and four-terminal types can be in the system configuration simultaneously, but the total number of GPIs cannot exceed 16.

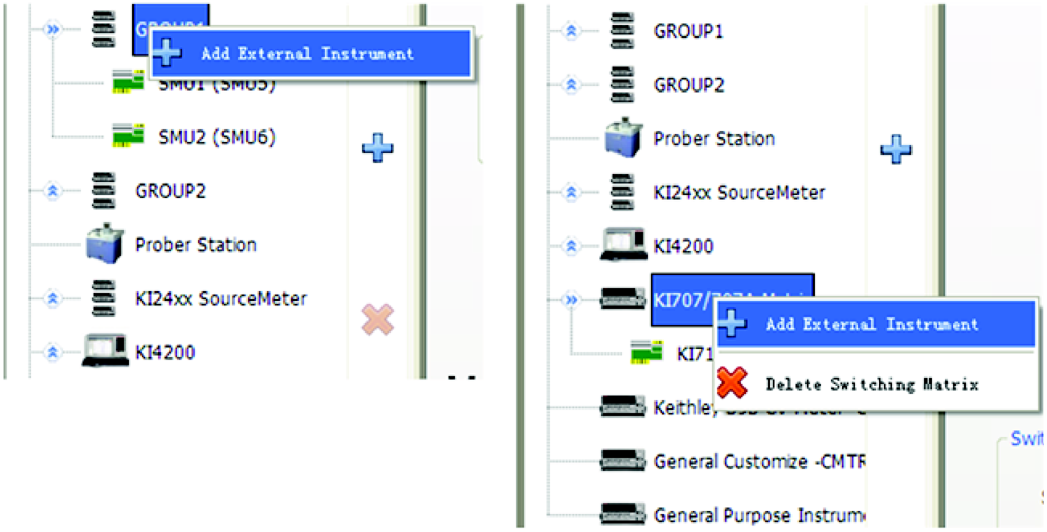
NOTE

ACS Basic provides Python LPT libraries for each supported external instrument. You can use these libraries to create your own test modules. ACS Basic also provides a number of standard Python user libraries to control external equipment that is typically used in semiconductor characterization applications. Standard user-module libraries are provided for equipment that can be used with ACS Basic (refer to Test Setup for more information).

Add an external instrument

1. Click the add instrument icon ✚, or right-click the instrument node in the configuration navigator, and click the pop-up item (see next figure).

Figure 38: Add an external instrument



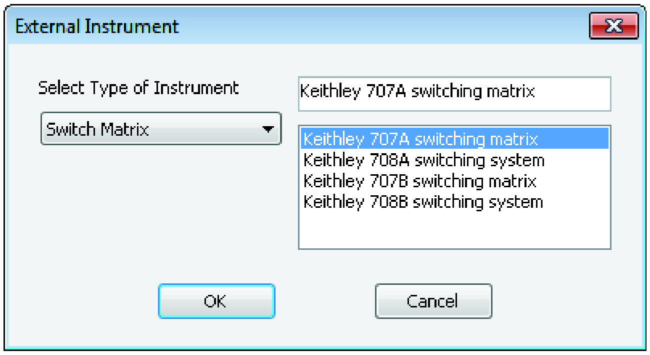
2. After clicking the Add External Instrument option, you will see the External instrument dialog box. In this dialog box you can choose the type of instrument to add (see next figure).
3. Select the instrument that you have in your hardware configuration and click OK to accept or click Cancel to exit the dialog without making any changes.

Add a switching matrix

From the Select Type of Instrument drop-down list select Switch Matrix and a list of supported switch matrix instruments appears in the list box on the right (see next figure). The following switch matrix instruments are supported by ACS Basic:

- Model 707A/707B switching matrix
- Model 708A/708B switching system

Figure 39: Select switch Matrix



Select Type of Instrument: Switch Matrix

Keithley 707A switching matrix

Keithley 707A switching matrix
Keithley 708A switching system
Keithley 707B switching matrix
Keithley 708B switching system

Add a capacitance meter

From the Select Type of Instrument drop-down list select Capacitance Meter and a list of supported capacitance meters appears in the list box on the right. The following capacitance meters are supported by ACS Basic:

- Keithley 590 CV Analyzer
- Keithley 595 Quasistatic CV Meter

Add a general purpose instrument

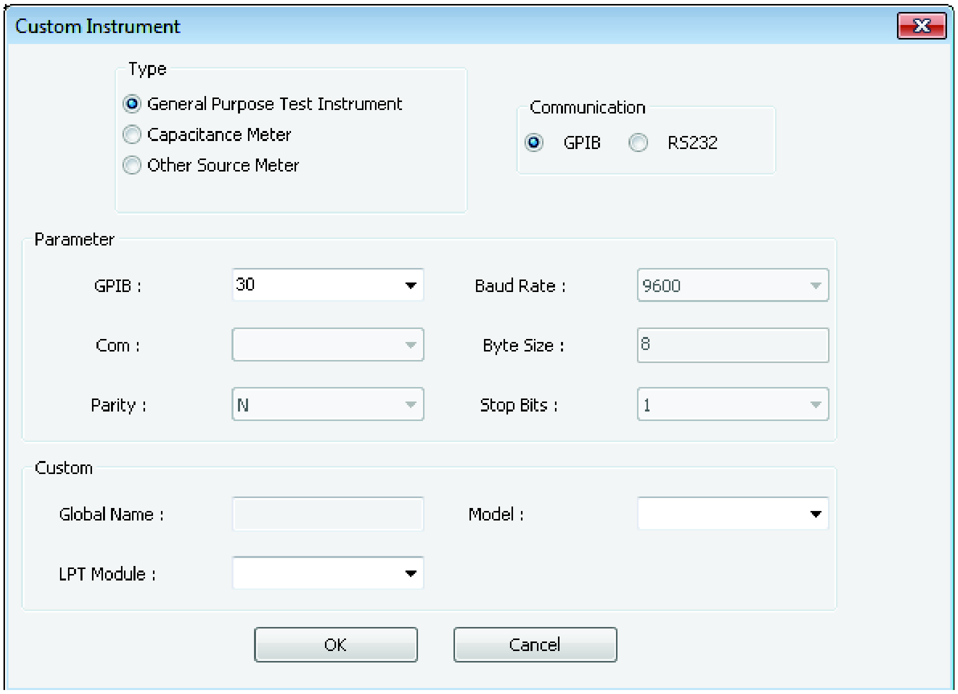
From the Select Type of Instrument drop-down list select General Purpose Instrument (GPI) and a list of supported general purpose instruments appears in the list box on the right. The following general purpose instruments are supported by ACS Basic:

- Generic 2 Terminal GPI
- Generic 4 Terminal GPI
- Custom GPI

Custom GPI

If you selected Custom GPI, the custom instrument window opens where you must input some basic information (see next figure):

Figure 40: Custom Instrument dialog box



Custom Instrument

Type

● General Purpose Test Instrument
○ Capacitance Meter
○ Other Source Meter

Communication

● GPIB ○ RS232

Parameter

GPIB : 30
Baud Rate : 9600
Com :
Byte Size : 8
Parity : N
Stop Bits : 1

Custom

Global Name :
Model :
LPT Module :

- Select the type of custom instrument:
  - General Purpose Test Instrument: some custom instruments, including those instruments from other companies.
  - Capacitance Meter: supports AG4284 or AG4980 capacitance meter.
  - Other Source Meter: supports Model 236/237/238 SMU.
- Select the communication:
  - GPIB
  - RS232
- Define parameters associated with the chosen communication.If you select General Purpose Test Instrument, then enter the model name. If you select Capacitance Meter or Other Source Meter, then use the drop-down list in the Model field to select the instrument Model.
- Select the LPT Module. The LPT Module refers to the LPT libraries that are installed in ACS Basic. For the General Purpose Test Instrument, you will not have to add an LPT module.

NOTE

There are LPT libraries installed when you install ACS Basic. After installation, the Python LPT libraries are saved in the folder: (C:\ACS_Basic\library\pyLibrary\ptmlpt). ACS Basic supplies these LPT library packages: ki23xlpt, ki24xxlpt, ki26xxlpt, ki34xxlpt, ki37xxlpt, ki42cvulpt, ki42xxlpt, ki70xlpt, ki70xblpt, kiag4284lpt. For the external instrument, such as Model 236, 237, HP4284, you can select the corresponding LPT.

Delete an external instrument

Delete an external instrument to remove the instrument from the hardware configuration.

NOTE

Only an external instrument can be deleted (using the delete icon ❌) from the hardware configuration navigator.

- Right-click on the external instrument that you want to remove in the configuration navigator, then click the delete external instrument function.
- Alternatively, you can delete an external instrument by selecting it and then click the delete icon ❌ in the software.

Save the hardware configuration

The Save icon 💾 on the toolbar of Hardware Configure GUI is enabled when you modify any configurable properties. Click the **Save** icon 💾 to save your changes.

NOTE

You must close and restart ACS Basic after you make a change to the hardware configuration or the new configuration will not be functional.

View and setup instrument properties

Series 2600B system properties

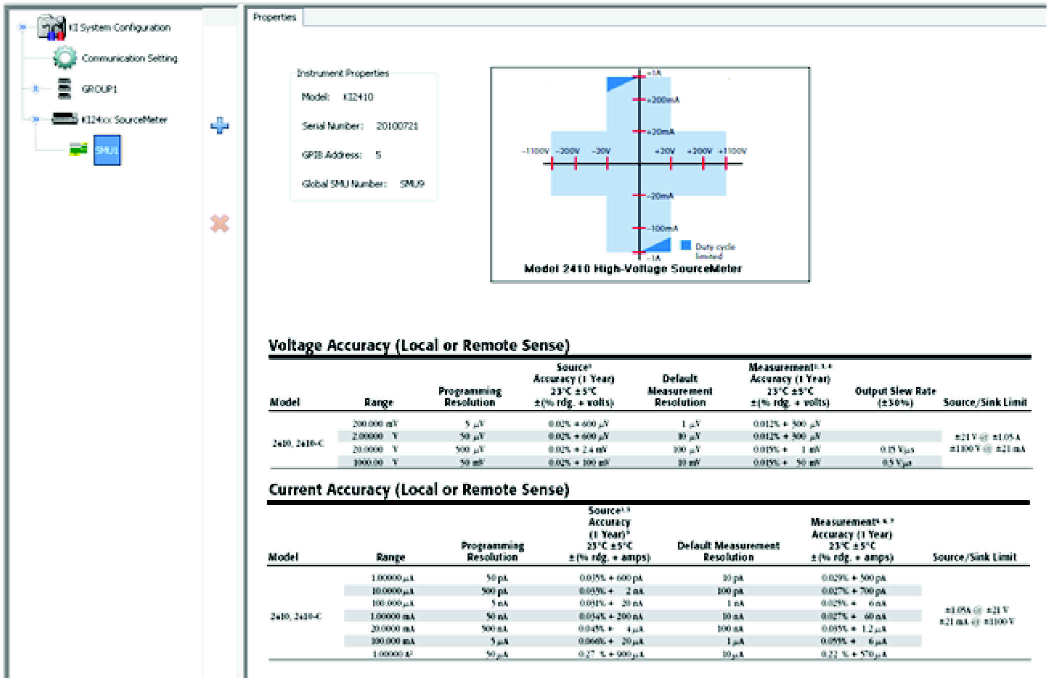
Single configuration: When the node of a Series 2600B instrument on the configuration navigator is selected, the work area displays an overview of the instrument(s).

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 (see next figure).

Example: Model 2651A

Figure 41: Model 2651A information



Instrument Properties

Model: KI2410

Serial Number: 20100721

GPIB Address: 5

Global SMU Number: SMU9

Voltage Accuracy (Local or Remote Sense)

Model	Range	Programming Resolution	Source[1] Accuracy (1 Year) 23°C ±5°C ±(% rdg. + volts)	Default Measurement Resolution	Measurement[2, 3, 4] Accuracy (1 Year) 23°C ±5°C ±(% rdg. + volts)	Output Slew Rate (±30%)	Source/Sink Limit
2410, 2410-C	200.000 mV	5 µV	0.02% + 600 µV	1 µV	0.012% + 300 µV		±21 V @ ±1.05 A ±1100 V @ ±21 mA
	2.00000 V	50 µV	0.02% + 600 µV	10 µV	0.012% + 300 µV		
	20.0000 V	500 µV	0.02% + 2.4 mV	100 µV	0.015% + 1 mV	0.15 V/µs	
	1000.00 V	50 mV	0.02% + 100 mV	10 mV	0.015% + 50 mV	0.5 V/µs	

Current Accuracy (Local or Remote Sense)

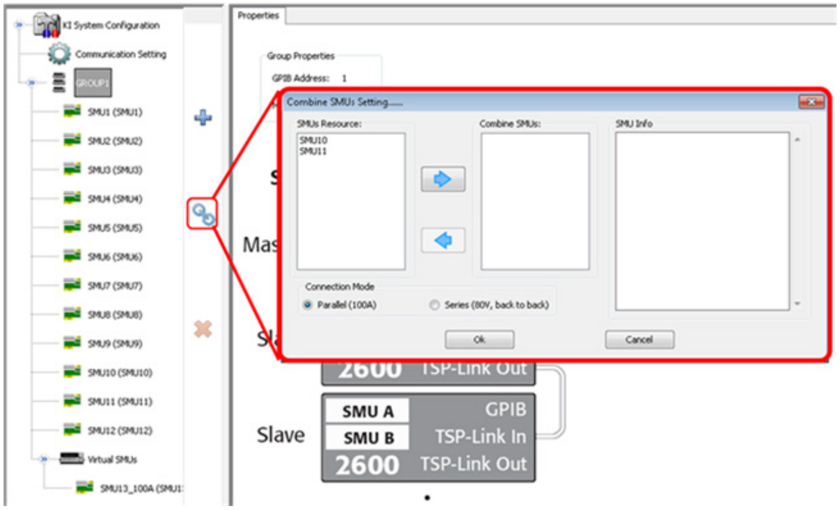
Model	Range	Programming Resolution	Source[1, 5] Accuracy (1 Year)[6] 23°C ±5°C ±(% rdg. + amps)	Default Measurement Resolution	Measurement[4, 7] Accuracy (1 Year) 23°C ±5°C ±(% rdg. + amps)	Source/Sink Limit
2410, 2410-C	1.00000 µA	50 pA	0.035% + 600 pA	10 pA	0.029% + 300 pA	±1.05A @ ±21 V ±21 mA @ ±1100 V
	10.0000 µA	500 pA	0.033% + 2 nA	100 pA	0.027% + 700 pA	
	100.000 µA	5 nA	0.031% + 20 nA	1 nA	0.025% + 6 nA	
	1.00000 mA	50 nA	0.034% + 200 nA	10 nA	0.027% + 60 nA	
	20.0000 mA	500 nA	0.045% + 4 µA	100 nA	0.035% + 1.2 µA	
	100.000 mA	5 µA	0.066% + 20 µA	1 µA	0.055% + 6 µA	
	1.00000 A	50 µA	0.27 % + 900 µA	10 µA	0.22 % + 570 µA	

The Model of the SMU is displayed in the first line. The SMU offset information displays in the second line: SMU A corresponds to offset=0; SMU B corresponds to offset=1. The next three lines display the SMU node number, its hardware version, and serial number. The last two lines display the calibration date. The first is the last time the SMU was calibrated, while the second is the scheduled date when the SMU should be calibrated.

Composite SMU

ACS Basic enables you to combine two 2651A SMUs in parallel as a single SMU. This will give you the capability to output 100 amps of current, or use in series for an 80 volt output. In the configuration navigator, the combined 2651A SMUs are considered a composite SMU. The work area will show the properties for the composite SMU.

Figure 42: Combine two 2651A SMUs



Group Properties
GPIB Address: 1

Combine SMUs Setting.....

SMUs Resource: SMU10 SMU11

Combine SMUs:

SMU Info

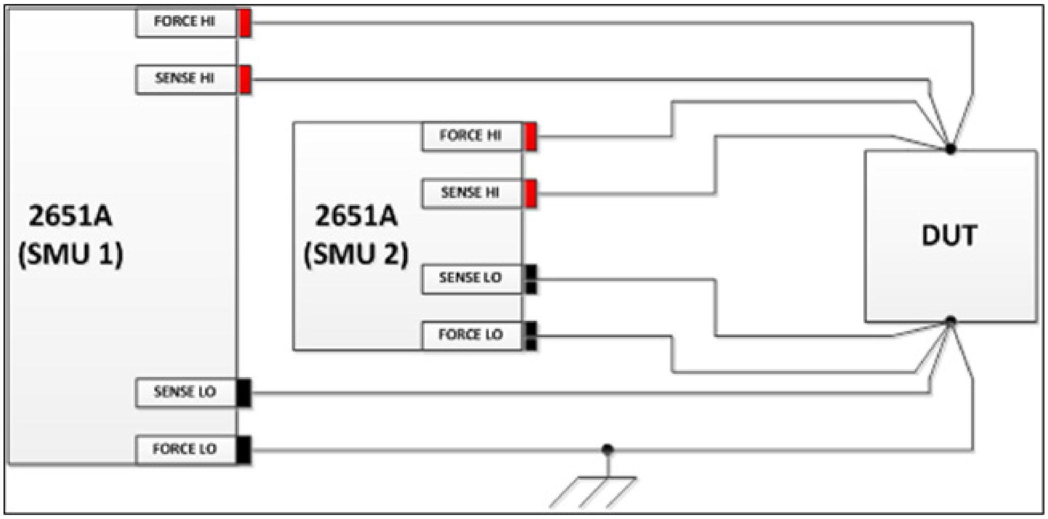
Connection Mode
● Parallel (100A) ○ Series (80V, back to back)

Ok Cancel

Click the Combine SMUs icon and a dialog box opens where you choose the SMUs to combine (only two 2651A SMUs can be combined) and where you choose the Connection Mode (see previous figure):

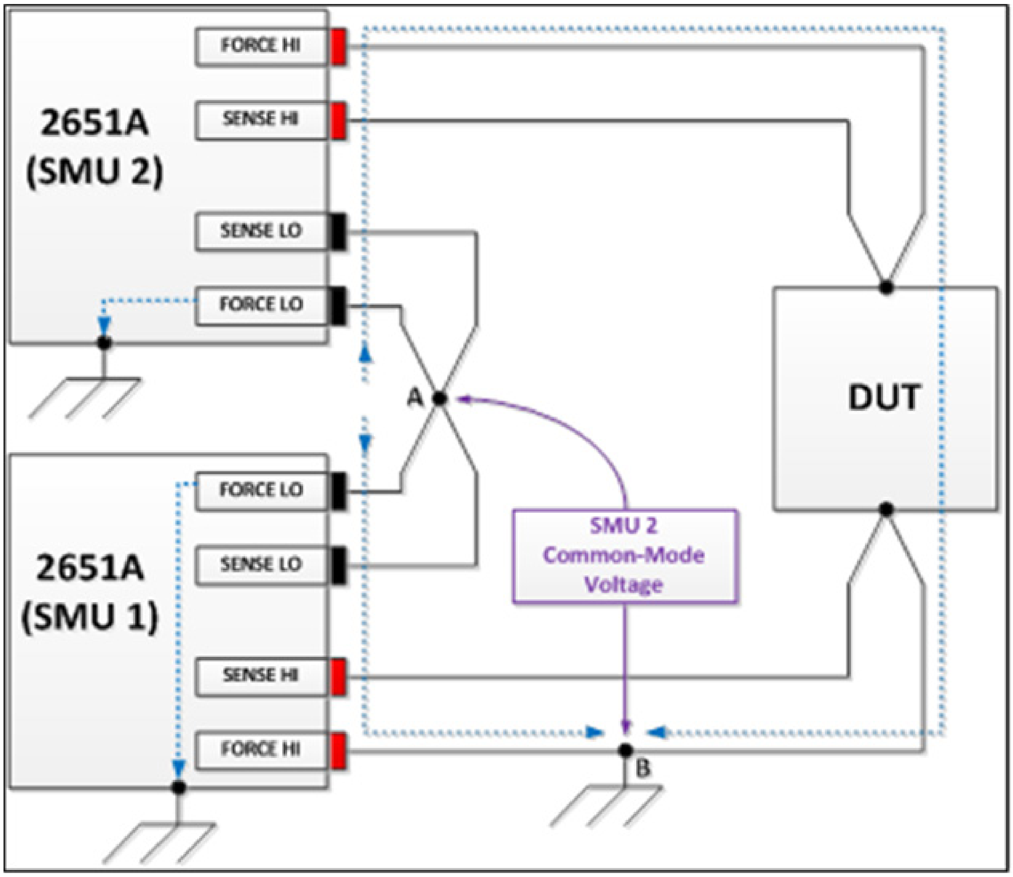
1. Parallel (100A). Two 2651A SMUs are in parallel for higher current output. The composite SMU is named SMUx_100A (x represents the SMU number).

Figure 43: Parallel (100A) composite SMU connections



2. Series (80V, back to back). Two 2651A SMUs are in back-to-back series for a higher voltage output. The composite SMU has two terminals: SMUx_80V_HI and SMUx_80V_LO (x represents the SMU number).

Figure 44: Series 2650A composite SMU connections

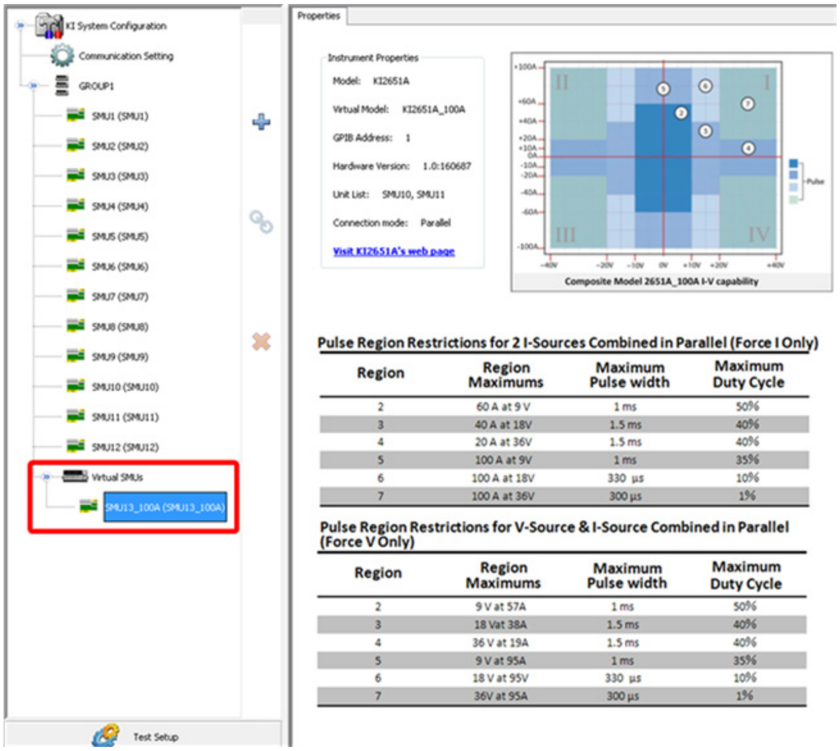


NOTE

The parallel (100A) SMU can be used in an ITM and trace mode for certain modules. The series 2650A (80V, back to back) SMU can only be used in a 2-terminal device in trace mode.

In the configuration navigator, the composite SMU is labeled "Virtual SMUs" (see next figure). The properties of the composite SMU are displayed in the work area.

Figure 45: Virtual SMUs



Instrument Properties

Model: KI2651A

Virtual Model: KI2651A_100A

GPIB Address: 1

Hardware Version: 1.0:160687

Unit List: SMU10, SMU11

Connection mode: Parallel

Visit KI2651A's web page

Pulse Region Restrictions for 2 I-Sources Combined in Parallel (Force I Only)

Region	Region Maximums	Maximum Pulse width	Maximum Duty Cycle
2	60 A at 9 V	1 ms	50%
3	40 A at 18V	1.5 ms	40%
4	20 A at 36V	1.5 ms	40%
5	100 A at 9V	1 ms	35%
6	100 A at 18V	330 µs	10%
7	100 A at 36V	300 µs	1%

Pulse Region Restrictions for V-Source & I-Source Combined in Parallel (Force V Only)

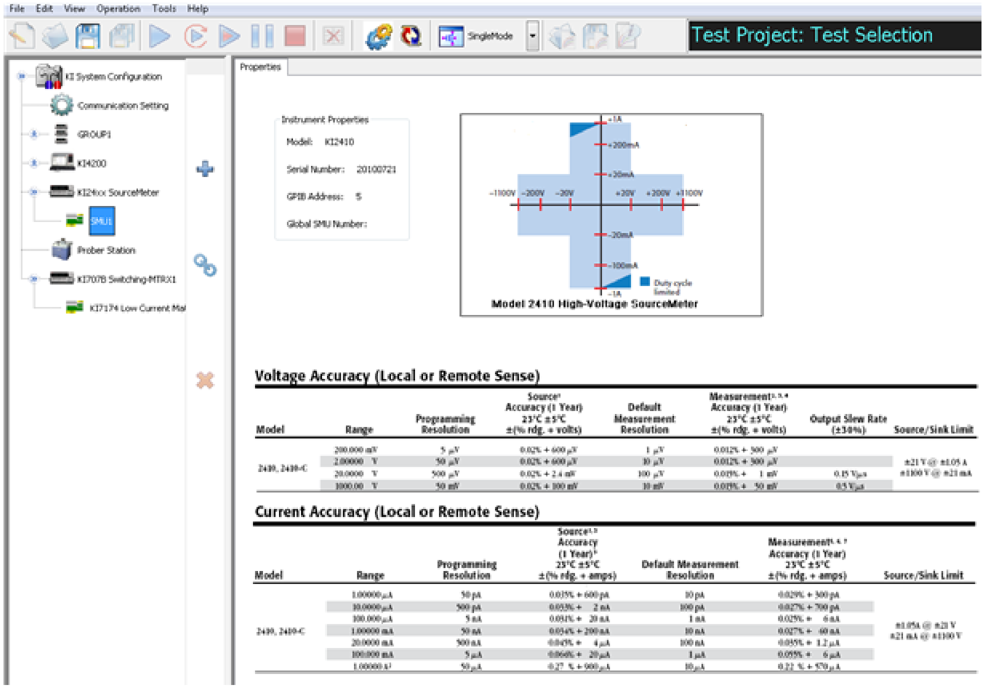
Region	Region Maximums	Maximum Pulse width	Maximum Duty Cycle
2	9 V at 57A	1 ms	50%
3	18 Vat 38A	1.5 ms	40%
4	36 V at 19A	1.5 ms	40%
5	9 V at 95A	1 ms	35%
6	18 V at 95V	330 µs	10%
7	36V at 95A	300 µs	1%

Series 2400 system properties

Series 2400 SMU configuration

When a Series 2400 SMUx node is selected in the configuration navigator, the work area shows the SMU properties, specifications, and power capacity (see next figure).

Figure 46: Model 2400 SMU information



Instrument Properties

Model: KI2410

Serial Number: 20100721

GPIB Address: 5

Global SMU Number:

Voltage Accuracy (Local or Remote Sense)

Model	Range	Programming Resolution	Source[1] Accuracy (1 Year) 23°C ±5°C ±(% rdg. + volts)	Default Measurement Resolution	Measurement[2,3,4] Accuracy (1 Year) 23°C ±5°C ±(% rdg. + volts)	Output Slew Rate (±30%)	Source/Sink Limit
2410, 2410-C	200.000 mV	5 μV	0.02% + 600 μV	1 μV	0.012% + 300 μV		±21 V @ ±1.05 A ±1100 V @ ±21 mA
	2.00000 V	50 μV	0.02% + 600 μV	10 μV	0.012% + 300 μV		
	20.0000 V	500 μV	0.02% + 2.4 mV	100 μV	0.015% + 1 mV	0.15 V/μs	
	1000.00 V	50 mV	0.02% + 100 mV	10 mV	0.015% + 50 mV	0.5 V/μs	

Current Accuracy (Local or Remote Sense)

Model	Range	Programming Resolution	Source[1,5] Accuracy (1 Year)[6] 23°C ±5°C ±(% rdg. + amps)	Default Measurement Resolution	Measurement[3,4,7] Accuracy (1 Year) 23°C ±5°C ±(% rdg. + amps)	Source/Sink Limit
2410, 2410-C	1.00000 μA	50 pA	0.035% + 600 pA	10 pA	0.029% + 300 pA	±1.05A @ ±21 V ±21 mA @ ±1100 V
	10.0000 μA	500 pA	0.033% + 2 nA	100 pA	0.027% + 700 pA	
	100.000 μA	5 nA	0.031% + 20 nA	1 nA	0.025% + 6 nA	
	1.00000 mA	50 nA	0.034% + 200 nA	10 nA	0.027% + 60 nA	
	20.0000 mA	500 nA	0.045% + 4 μA	100 nA	0.035% + 1.2 μA	
	100.000 mA	5 μA	0.066% + 20 μA	1 μA	0.055% + 6 μA	
	1.00000 A[1]	50 μA	0.27 % + 900 μA	10 μA	0.22 % + 570 μA	

The instrument Properties area displays the following information:

- SMU Model
- SMU Serial Number
- GPIB Address
- Instrument name

Matrix instruments

Models 707A/707B and 708A/708B switch matrices

Models 707A, 707B, 708A, and 708B switch matrices can be added to the ACS Basic hardware configurations as external instruments with GPIB connections. Models 707B and 708B can also be automatically scanned and added to the configuration through the GPIB and TSP link as a master or subordinate instrument.

Note that ACS Basic supports multiple matrices, such as Models 707A, 707B, 708A, 708B, and Series 3700A instruments in the hardware configuration. Additionally, ACS Basic supports multiple matrices as a group or as individual instruments.

ACS Basic also supports both device dependent command (DDC) mode and instrument control language (ICL) mode for Models 707B and 708B:

- DDC mode instruments must be manually added to the hardware configuration
- ICL mode instruments are automatically scanned and added to the hardware configuration

Model 707B and 708B configuration

When a Model 707B or 708B node is selected in the configuration navigator, the instrument Properties, Instrument Connection Scheme, and specifications are viewable in the work area.

The Model 707B or 708B Properties tab contains:

- Instrument Properties:
  - model of instrument
  - associated group (ICL mode)
  - node (ICL mode)
  - work mode (ICL or DDC)
  - card list (ICL mode)
  - GPIB address (DDC mode).
- S530 System: Kelvin mode enabled or disabled (Model 707A/707B only).
- Instrument Connection Scheme:
  - Row-Column or Instrument-Card
  - Local Sense or Remote Sense
  - Common LO or Independent LO
- Switch Card: select matrix cards for the system (see next figure):
  - Model 7071 Matrix Card
  - Model 7072 Matrix Card
  - Model 7072 HV Matrix Card
  - Model 7136 Low Current MUX Card
  - Model 7174 Low Current Matrix Card
  - Model 9174 Semiconductor Matrix Card

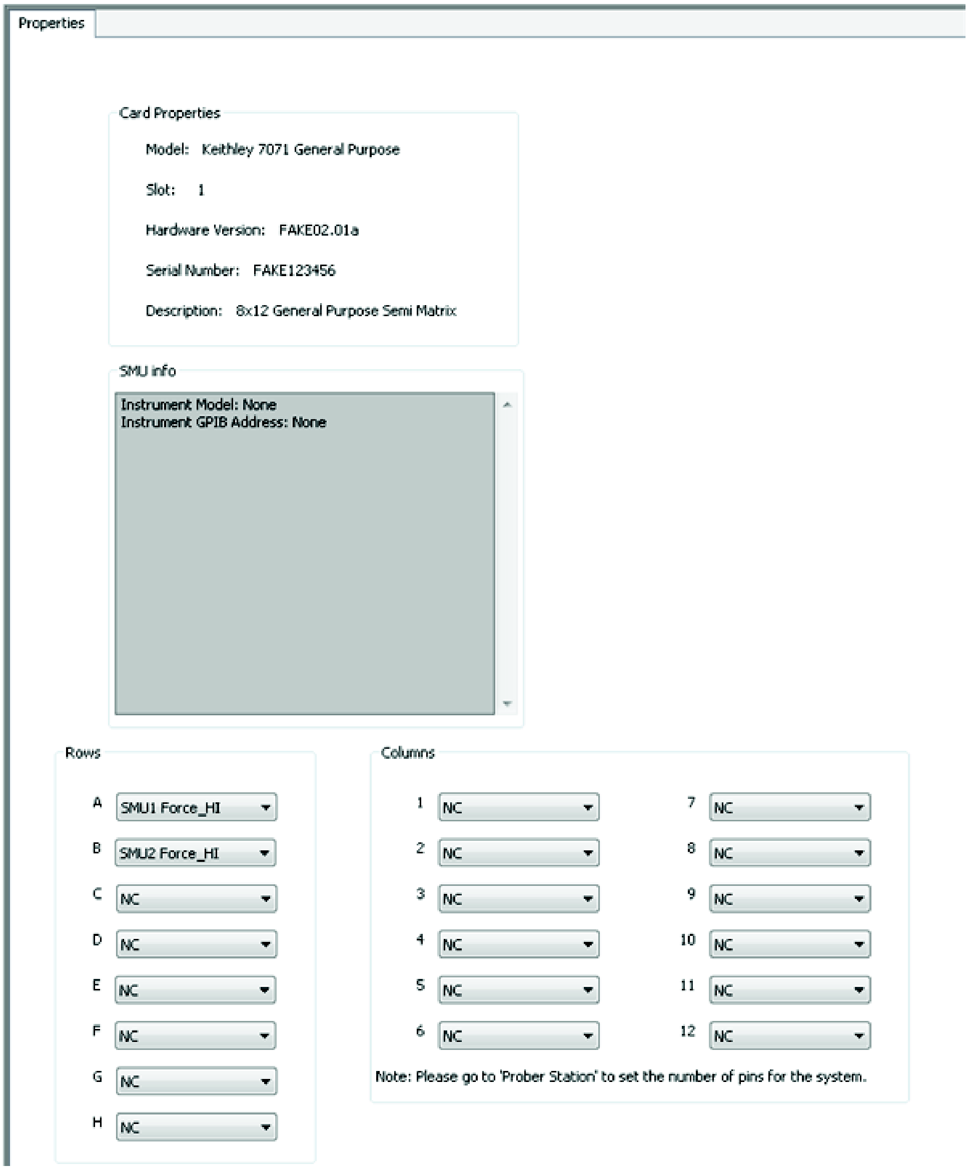
NOTE

If any of the matrices in the system are set to Remote Sense or the S530 Kelvin mode, all of 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 Preference settings Misc tab will be disabled. When all the matrices are set to Local Sense, all of 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.

Matrix Card information

When you click one of the matrix cards on the hardware configuration navigator, a GUI dialog box opens. The GUI is where you configure the hardware (see next figure).

Figure 47: Model 7071 matrix card properties



Card Properties

Model: Keithley 7071 General Purpose

Slot: 1

Hardware Version: FAKE02.01a

Serial Number: FAKE123456

Description: 8x12 General Purpose Semi Matrix

SMU info

Instrument Model: None
Instrument GPIB Address: None

Rows

A	SMU1 Force_HI
B	SMU2 Force_HI
C	NC
D	NC
E	NC
F	NC
G	NC
H	NC

Columns

1	NC	7	NC
2	NC	8	NC
3	NC	9	NC
4	NC	10	NC
5	NC	11	NC
6	NC	12	NC

Note: Please go to 'Prober Station' to set the number of pins for the system.

The Matrix Card Properties tab contains:

- Card Properties: instrument model, slot number, hardware version, serial number, and the description of the card.
- SMU info: instrument model, GPIB address, maximum voltage and current, voltage and current range.
- Rows: A through H correspond to the eight (8) rows of all Model 707B/708B compatible matrix cards. Use the drop-down arrow to connect the rows to various instrument terminals.
- Columns: one through 12 correspond to columns for the Model 707B/708B compatible matrix cards. Use the drop-down arrows to connect the columns to various instrument terminals or prober test-fixture pins, or any combination of both terminals and prober pins depending on your needs.

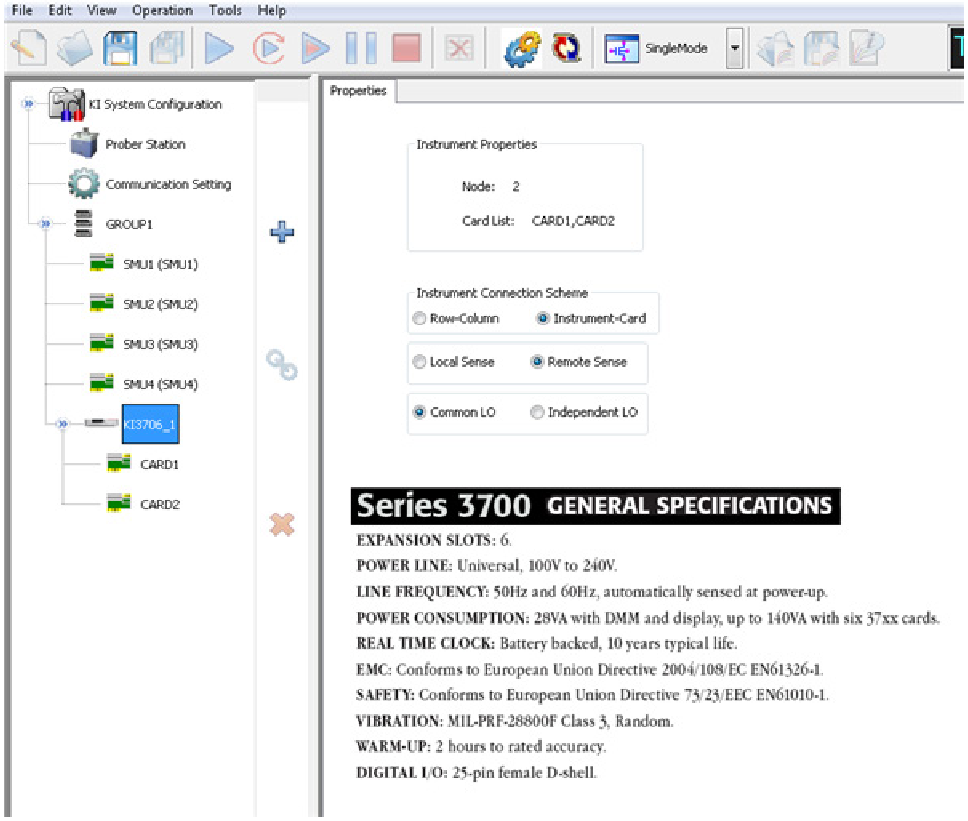
NOTE

Due to the Model 2651A high current SMU (up to 50A), and the Model 2657A high power SMU (up to 3000V), you will not be able to add a Model 2651A SMU and a Model 2657A SMU to the matrix card row or column.

Model 3706A configuration

When a Model 3706A node is selected in the configuration navigator, the work area shows the Properties, Instrument Connection Scheme, and specifications (see next figure).

Figure 48: Model 3706 information



Instrument Properties

Node: 2

Card List: CARD1,CARD2

Instrument Connection Scheme

○ Row-Column ● Instrument-Card

○ Local Sense ● Remote Sense

● Common LO ○ Independent LO

Series 3700 GENERAL SPECIFICATIONS

EXPANSION SLOTS: 6.

POWER LINE: Universal, 100V to 240V.

LINE FREQUENCY: 50Hz and 60Hz, automatically sensed at power-up.

POWER CONSUMPTION: 28VA with DMM and display, up to 140VA with six 37xx cards.

REAL TIME CLOCK: Battery backed, 10 years typical life.

EMC: Conforms to European Union Directive 2004/108/EC EN61326-1.

SAFETY: Conforms to European Union Directive 73/23/EEC EN61010-1.

VIBRATION: MIL-PRF-28800F Class 3, Random.

WARM-UP: 2 hours to rated accuracy.

DIGITAL I/O: 25-pin female D-shell.

A Model 3706A Properties tab contains the following:

- Instrument Properties: displays the node and the two types of cards: multiplexer or matrix.

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 the instrument FORCE terminals) or Remote Sense (connections to both the instrument FORCE and SENSE terminals).
- Series 3700A General Specifications: displays the information about the specifications.

Series 3700A: multiplexer card

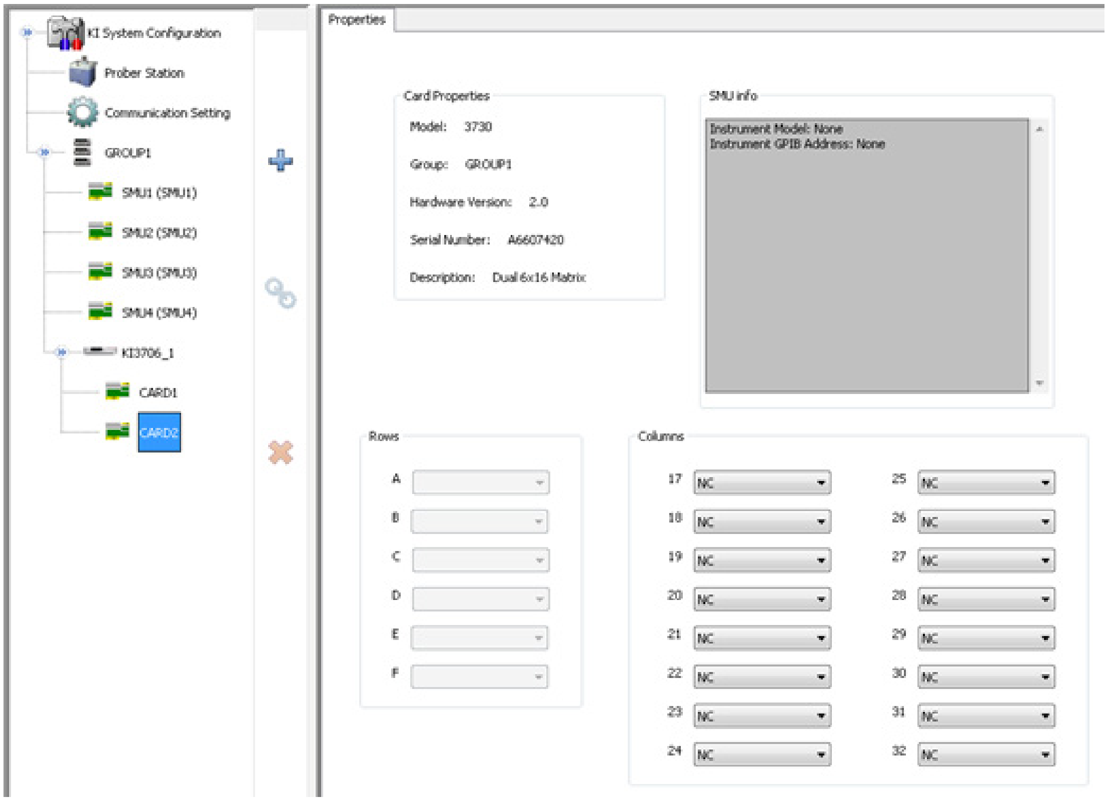
The next figure shows two cards are inserted in a Series 3700A instrument. They are the multiplexer card (CARD1) and the switch card (CARD2). When the multiplexer (CARD1) node is selected, the work area shows the properties and specifications for the multiplexer (CARD1).

The Card Properties group box of a multiplexer card contains information about the model, group, hardware version, serial number, and description. Specifications for the card are also listed.

Series 3700A: matrix card

When the matrix card (CARD2) node is selected, the work area shows the properties and connection options that are defined for the device (see next figure).

Figure 49: Matrix card information



Model: 3730

Group: GROUP1

Hardware Version: 2.0

Serial Number: A6607420

Description: Dual 6x16 Matrix

Instrument Model: None
Instrument GPIB Address: None

17	NC	25	NC
18	NC	26	NC
19	NC	27	NC
20	NC	28	NC
21	NC	29	NC
22	NC	30	NC
23	NC	31	NC
24	NC	32	NC

The Card Properties area contains information about the model, group, hardware version, serial number, and description. There is also a SMU info area, and two additional areas that provide information about the connection method:

1. Rows: In the Rows area, the edit boxes labeled A through F correspond to the 6 rows for the Model 3706A compatible matrix cards. Use the drop-down arrows to select the rows for your instrument terminals.
2. Columns: In the Columns area, the edit boxes labeled 1 through 16 correspond to the 16 columns for the Model 3706A compatible matrix cards. Use the drop-down arrows to select the columns for your instrument terminals and/or prober/test-fixture pins.

NOTE

You can only connect instrument terminals to the matrix card columns when the Instrument Connection Scheme setting is active (see next figure).

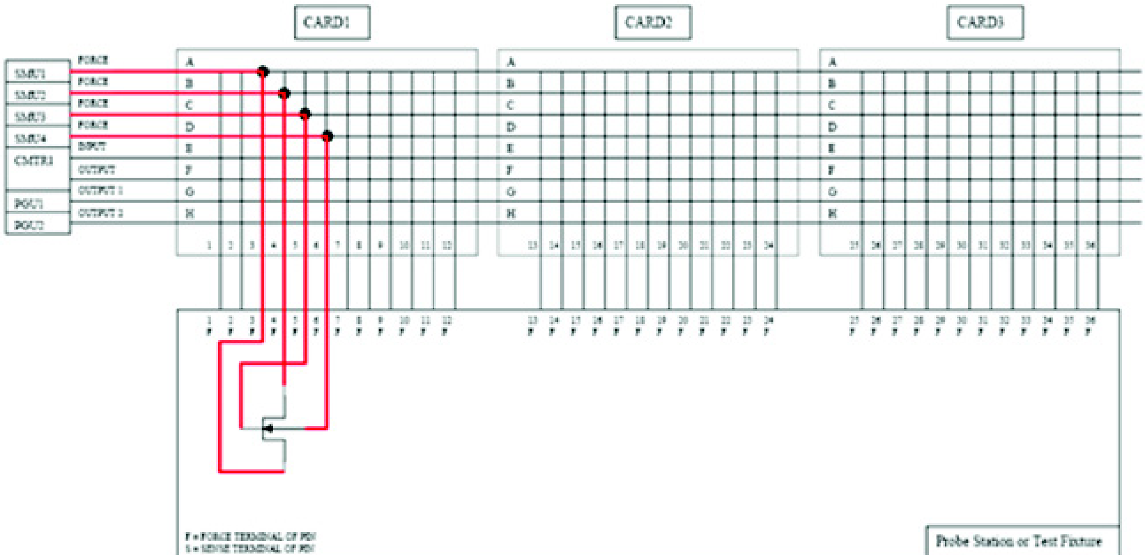
Matrix instruments

Row-Column: instruments are connected to switch matrix rows, and prober/test fixture pins are connected to switch matrix columns (see next figure). This is the simplest connection scheme. Instrument signals can route to the prober/test-fixture pins through one matrix card. However, the Row-Column scheme limits the number of external instruments.

NOTE

Prober or test-fixture pins are always connected to matrix card columns.

Figure 50: Row-column, local sense connection scheme example



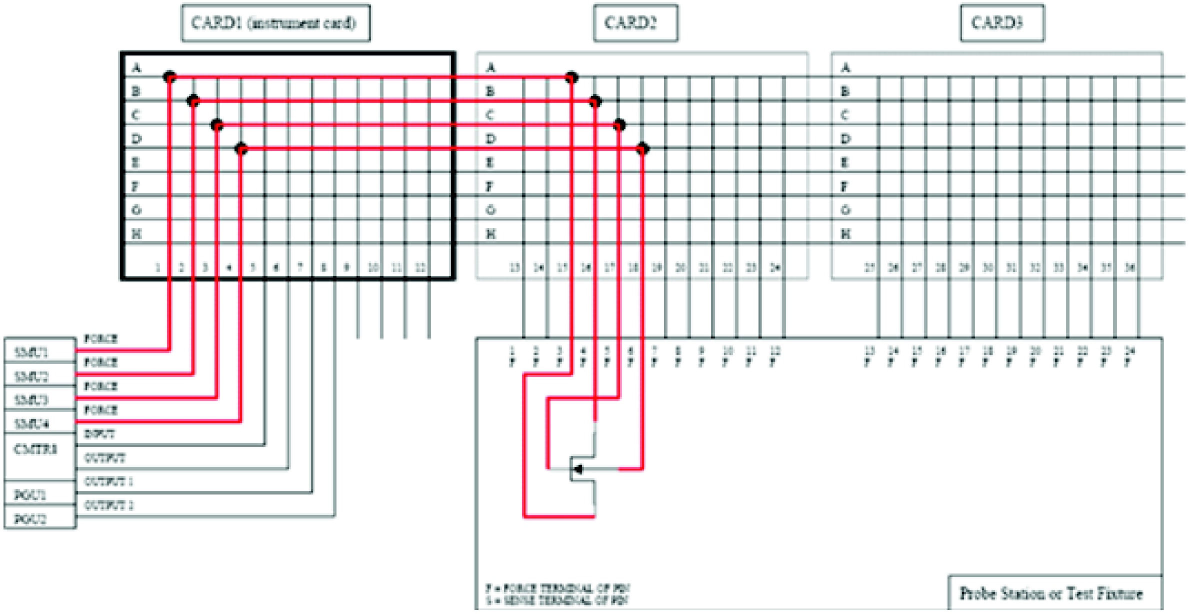
If you need to connect multiple external instruments to the prober/test-fixture, use the Instrument Card scheme.

Instrument Card: both instruments and prober/test-fixture pins are connected to the switch matrix columns. Instrument signals can route to the prober/test-fixture pins through two or more matrix cards. This connection scheme can support more instruments compared to the Row-Column scheme (see next figure).

NOTE

Due to the Model 2651A high current SMU (up to 50A), you will not be able to add a Model 2651A SMU to the matrix card row or column.

Figure 51: Instrument card, local sense connection scheme example



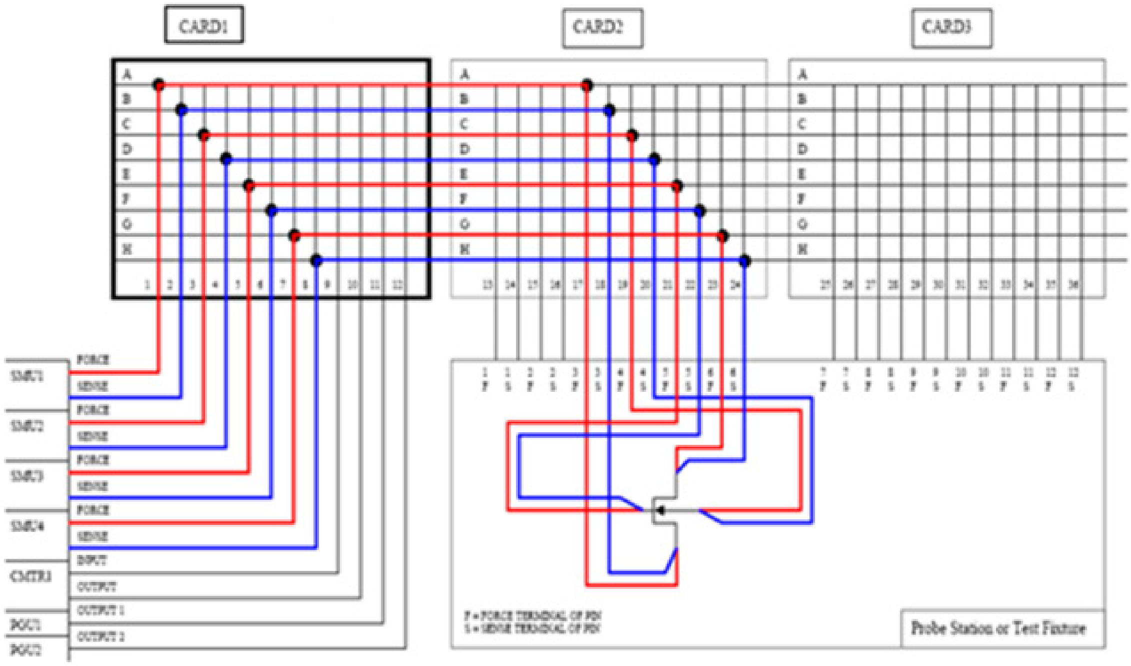
- Two-wire local sense: use this when the measurement-pathway resistance is small and the associated voltage errors are negligible. The measurement pathway is comprised of the following conductors, connected in series:
  a. Cables used to connect the instruments to the matrix
  b. Internal matrix-card signal path
  c. 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.

Four-wire remote sense can be used when sourcing and/or measuring voltage in a low-impedance test circuit, and there can be errors associated with IR drops in the test leads. Voltage source and measure accuracy are optimized by using four-wire remote sense connections. When sourcing voltage, four-wire remote sensing ensures that the programmed voltage is delivered to the DUT. When measuring voltage, only the voltage drop across the DUT is measured.

- Use four-wire remote sense for the following source-measure conditions
  a. Sourcing and/or measuring voltage in low impedance (<1 kΩ) test circuits.
  b. Enforce voltage compliance limit directly at the DUT

Figure 52: Instrument card, remote sense connection scheme example

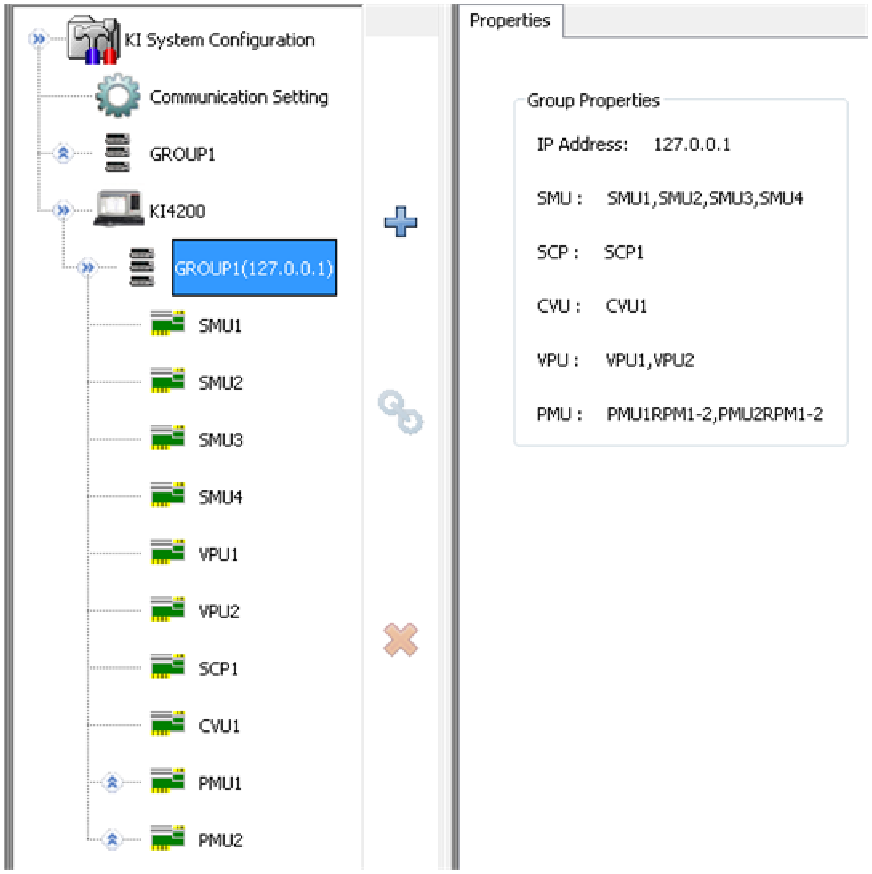


Model 4200 system properties

When you select the KI 4200 node in the configuration navigator, the Model 4200-SCS system properties are displayed in the work area (see next figure).

The work area shows the Model 4200 Group Properties, which contains the IP address, source measure unit (SMU) name, scope (SCP) name, capacitance-voltage unit (CVU) name, voltage pulse unit (VPU) name, and pulse measure unit (PMU) name.

Figure 53: Model 4200 group properties



Group Properties

IP Address: 127.0.0.1

SMU : SMU1,SMU2,SMU3,SMU4

SCP : SCP1

CVU : CVU1

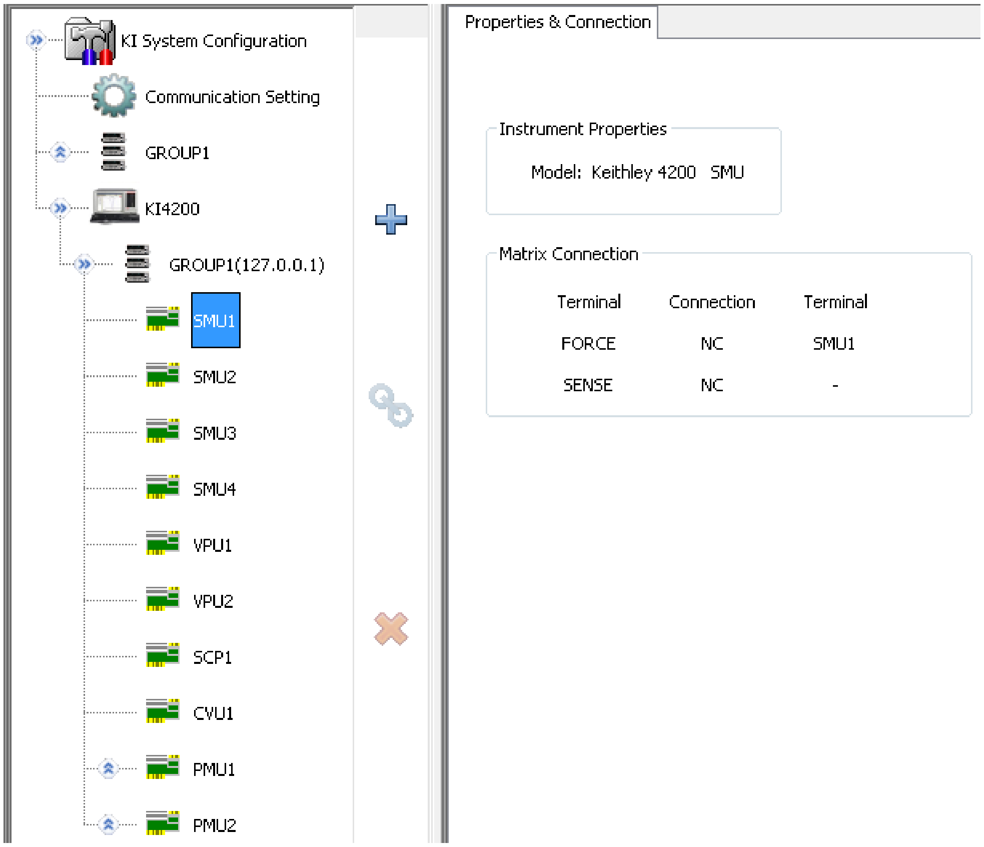
VPU : VPU1,VPU2

PMU : PMU1RPM1-2,PMU2RPM1-2

When you select the KI 4200 SMU node in the configuration navigator, the Model 4200-SCS SMU system properties are displayed in the work area (see next figure).

The work area shows the Model 4200 SMU where you can see the Instrument Properties (the model of the SMU). Additionally, you can see the Matrix Connection of the SMU that shows the Terminals, Connection, Force, and Sense status.

Figure 54: Model 4200 SMU information



Instrument Properties

Model: Keithley 4200 SMU

Matrix Connection

Terminal	Connection	Terminal
FORCE	NC	SMU1
SENSE	NC	-

Model 590 CV analyzer

When you select the Model 590 CV Analyzer in the configuration navigator, its Properties & Connections tab opens in the work area.

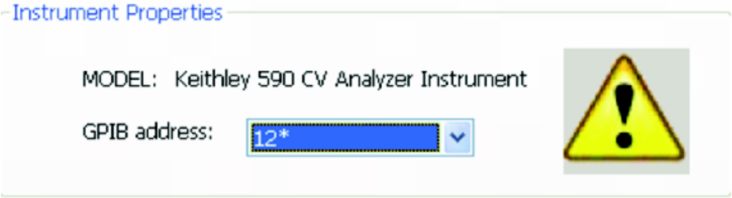
The Properties & Connections tab provides access to the Model 590 instrument properties and useful switch-matrix connection information. The Instrument Properties area of this Properties & Connections tab provides access to the following Keithley Instruments Model 590 instrument properties:

- Model: Full vendor name, model number, and instrument description.
- GPIB Address: you can use the drop-down arrow to select the address. Addresses that are in use are displayed with asterisks (*) next to them. The minimum address value is 0; the maximum is 30 (GPIB address 31 is reserved as the Model 4200-SCS controller address). If the selected GPIB address conflicts with the GPIB address of another instrument in the configuration, a warning exclamation-point symbol (!) is displayed next to the address (see next figure).

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 55: GPIB address properties



Instrument Properties

MODEL: Keithley 590 CV Analyzer Instrument

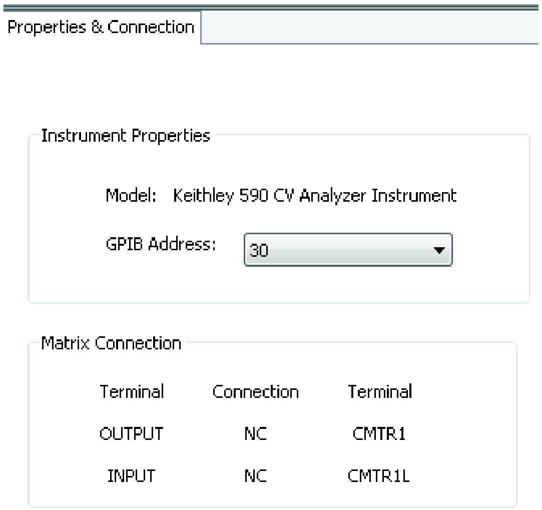
GPIB address: 12*

When a matrix is included in the system configuration, the Matrix Connections area of this Properties & Connections 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 (see next figure).

Figure 56: Configuration of capacitance meter



Instrument Properties

Model: Keithley 590 CV Analyzer Instrument

GPIB Address: 30

Matrix Connection

Terminal	Connection	Terminal
OUTPUT	NC	CMTR1
INPUT	NC	CMTR1L

Save the hardware configuration

NOTE

You must close and restart ACS Basic after you make a change to the hardware configuration or the new configuration will not be functional.

Check the hardware configuration

The hardware configuration utility is used to manage the hardware configuration. In order to check your configuration, follow these steps:

1. Click the Hardware configuration icon on the toolbar or choose Configure Hardware in the Tools menu of ACS Basic to start the hardware configuration (see next figure).
2. Click the Scan hardware icon on the toolbar. The hardware information is displayed in the work area.
3. Using the configuration navigator, navigate through the listed hardware. Verify that the hardware information matches the configuration of the physical system.
4. Click the Save icon to save the new configuration. Close and restart ACS Basic.

Configure the hardware in demo mode

You can use the scan hardware function to set up a custom SMU configuration. If you run ACS Basic in Demo mode, it can simulate a virtual system, such as including the Series 2600B, Series 2400, Series 3700A, and Model 4200.

For example, when you click the Scan Hardware icon , the hardware configuration dialog box opens (see next figure). This is where you can choose a specific instrument model in the drop-down list (for instance, KI2561A). Then you input the Total Groups and SMUs per Group information.

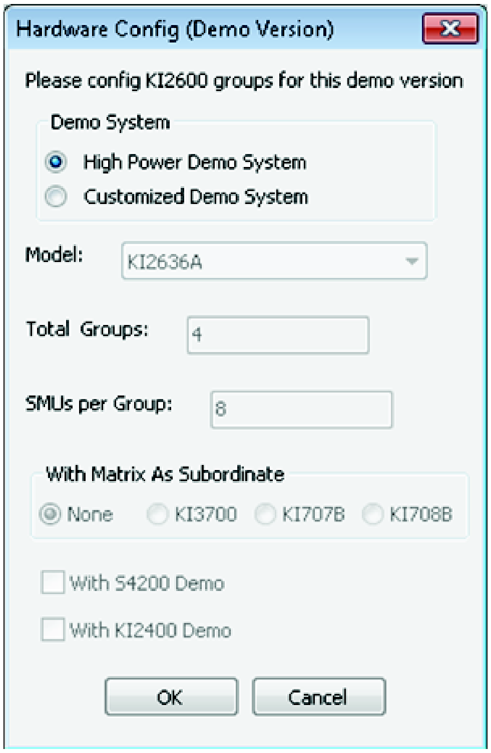
If you want a matrix (**KI3700A**, **KI707B**, or **KI708B**) in your configuration as a subordinate item, you have three choices, otherwise you can choose **None**.

If you want to simulate a scan of the Model 4200-SCS, select the **With S4200 Demo** item.

If you want to simulate a scan of a Series 2400 instrument select the **With KI2400 Demo** item.

For external instruments, you can also add and remove as needed.

Figure 57: Hardware configuration in demo mode



Hardware Config (Demo Version)

Please config KI2600 groups for this demo version

Demo System

● High Power Demo System

○ Customized Demo System

Model: KI2636A

Total Groups: 4

SMUs per Group: 8

With Matrix As Subordinate

● None ○ KI3700 ○ KI707B ○ KI708B

□ With S4200 Demo

□ With KI2400 Demo

Once you have completed your choices, click **OK** and the system configuration navigator is displayed in the left pane, with corresponding information displayed in the work area.

Section 5

ACS Basic test setup

In this section:

Test setup introduction

The ACS Basic interface consists of a variety of graphical user interfaces (GUIs) that allow you to do the following:

- Create tests for one or more devices
- Build and edit test sequences
- View test results in tabular and graphical forms
- Analyze test results using built-in parameter extraction tools

Test modes

ACS Basic has three test modes:

1. Single-test mode
2. Multi-test mode
3. Trace mode

The Single-test mode is used to open and run a single test on a single device at a time.

In Multi-test mode, multiple test modules can be created on one or more devices and organized into a single project. For example, Multi-test mode can perform multiple tests on a single device (target application) or multiple tests on multiple devices (secondary use case). You can work with external instruments (DMMs, switches, etc.) in Multi-test mode much easier than in any other mode.

Trace mode is for interactive operation with a source meter unit. You can adjust the maximum sweep value and maximum power to the device. Also, you can easily transition between DC and pulse forcing modes.

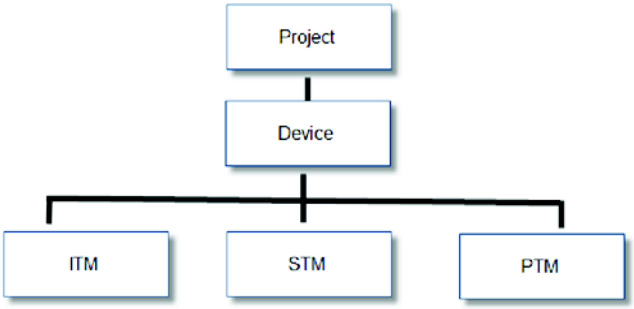
Test project

The test project defines and sequences all of the devices and specifies the tests to be performed on each device. Projects are only valid for Multi-test mode.

Devices and their respective tests are organized within the configuration navigator, which appears in the left panel of the ACS Basic interface. Use the configuration navigator to insert new test modules, copy and paste existing test modules, and sequence the order in which tests are performed.

In Multi-test mode, the ACS Basic configuration navigator follows a logical hierarchy:

Figure 58: Multi-test mode hierarchy



Devices

Each project contains one or more devices to be characterized including transistors, diodes, resistors, triacs, and capacitors.

Test modules

There are three types of test modules in ACS Basic:

1. Graphical interactive test module (ITM)
2. Script test module (STM)
3. Python language test module (PTM)

All three types of test modules have the same data sheet and data plot analysis functions. The differences between ITMs, STMs, and PTMs include the following:

Graphical interactive test module (ITM)

An ITM allows you to define a test interactively using a graphical user interface (GUI). ITMs are available for the Model 4200-SCS SMUs or Series 2600B System SourceMeter® instruments.

Script test module (STM)

An STM supports test script processor (TSP™) files. A TSP file is written for testing a device with a Series 2600B System SourceMeter, Model 3706A System Switch/Multimeter, or Model 707B/708B switch matrix in ICL mode. STM examples are installed with ACS Basic. These scripts can be imported and modified in the STM workspace. STMs can be used with a user-created GUI in an XRC format to create custom tests with a graphical interface.

Python test module (PTM)

A PTM is a user-generated python script that can control instruments through a GPIB cable. You can create an optional GUI to interface with the PTM in an XRC format, or in a customized GUI format. PTM script examples are installed when ACS Basic is installed. These scripts can be imported and modified in the PTM test module.

The following table indicates the instruments and communication interfaces supported with the appropriate test modules (table legend: Y = supported, Y+ = recommended, -- = not supported).

Instrument Series	ACS Basic installed on	Communication Interface	ITM	STM	PTM
2400	PC/laptop	GPIB	--	--	Y
2600	PC/laptop or 4200	GPIB, TSP subordinate	Y+	Y	Y
2600B	PC/laptop or 4200	GPIB, Ethernet, TSP subordinate	Y+	Y	Y
2651A & 2657A	PC/laptop or 4200	GPIB, Ethernet, TSP subordinate	Y+	Y	Y
4200	4200 only	GPIB	Y+	--	Y
3700A	PC/laptop or 4200	GPIB, Ethernet, TSP subordinate	--	Y+	Y
Other external instruments	PC/laptop or 4200	GPIB	--	--	Y

NOTE

If you copy and paste a test module, the device settings will not carry over. It is necessary to reassign the device settings if the test module is pasted in a different device. For example, if you copy an ITM from a nMOSFET device and paste it in a npnBJT device, you will need to manually reassign the device settings.

See Test Setup for more information on the types of test modules.

Single-test mode

Introduction

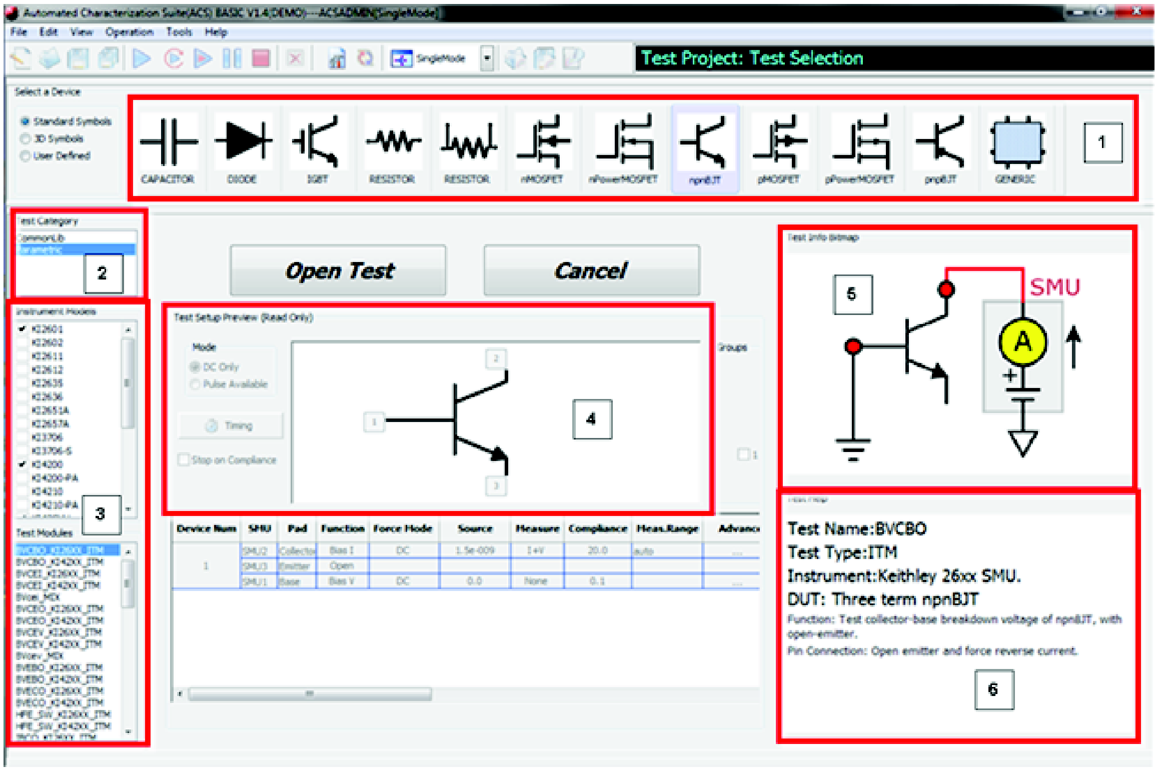
Single-test mode is used for importing and running one test module on one device at a time. The single test can be an ITM, STM, or PTM.

Test selection GUI

The test selection GUI of the Single-test mode is shown in the next figure and includes the following sections:

1. **Select a Device** - select a device view (Standard, 3D Symbols, or User Defined).
2. **Test Category** - click on the desired device and the selected icon will highlight.
3. **Instrument and Test Modules** - select the test instrument model. The test modules applicable for this instrument are listed in the Test Modules area. If two or more instruments are selected, the Test Modules are will display all of the modules available to any of the selected instruments.
4. **Test Setup Preview** - is a visual presentation of the type of test.
5. **Test Info Bitmap** - this is an image of the type of test selected.
6. **Test Help** - displays detailed specific information about the test.

Figure 59: Single-test mode test selection GUI



Device Num	SMU	Pad	Function	Force Mode	Source	Measure	Compliance	Meas.Range	Advance
1	SMU2	Collector	Bias I	DC	1.5e-009	I+V	20.0	auto	...
	SMU3	Emitter	Open						
	SMU1	Base	Bias V	DC	0.0	None	0.1		...

Test Name:BVCBO
Test Type:ITM
Instrument:Keithley 26xx SMU.
DUT: Three term npnBJT
Function: Test collector-base breakdown voltage of npnBJT, with open-emitter.
Pin Connection: Open emitter and force reverse current.

Select a test module in single-test mode

1. Select a device view (Standard, 3D Symbols, or User Defined).
   - Standard Symbols uses electrical symbol images.
   - 3D Symbols uses three-dimensional images of electrical symbols.
   - User Defined items are created using the Graphically Define a New Device dialog box. For more information on how to create a user defined device, refer to the Graphically define a new device topic.
2. Click on the desired device and the selected icon will highlight.
3. All test categories belonging to this device are listed in the Test Category area. Select your desired category:
   - CommonLib: The common library for test devices, including some tests that are for specific instruments, such as the CV_4200CVU test for the Model 42xx CVU, SWITCH_Control for the Models 707/708 matrix, KI37XX_DMM_R_2Wire for the Model 37xx, among others.
   - Parametric: Test libraries for device parametric tests. When you select a device, and select the instruments in the Instrument Models area, the Test Modules area will list all of the parametric tests available for the device. These test modules use specific SMUs for device parametric tests, such as, IdVd, IdVg test for MOSFET, among others.
   - WLR_script: The wafer level reliability tests for selected devices, such as the HCI (Hot Carrier Injection) nMOSFET device.

NOTE

Not all devices have a test library in all three test categories.

4. Select the test instrument model. The test modules applicable for this instrument are listed in the Test Modules area. If two or more instruments are selected the Test Modules area will display all of the modules available to any of the selected instruments.
5. Select a test module.
6. The test information preview is displayed on the right panel. This information includes Test Setup Preview, Test Information Bitmap, and Test Help.

Open and edit the test module

NOTE

Click the **Return to library** icon at any time to terminate the open test process and return to the previous test.

1. Click the **Open Test** button to open the test view GUI. In this GUI, you can modify the test module settings and run the test module.

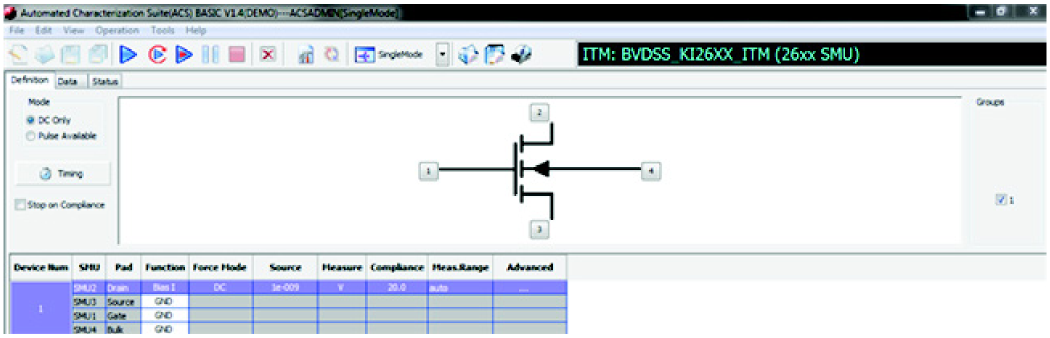
2. In the test view GUI, click the **Device view** icon to open the device view GUI (see next figure). Use Device View to designate the test and measurement instruments that are connected to each device terminal.

To configure an ITM, refer to Configure a graphical interactive test module (ITM).

To configure an STM, refer to Configure a script test module (STM).

To configure a PTM, refer to Configure a python test module (PTM).

Figure 60: Test view GUI


Automated Characterization Suite(ACS) BASIC V1.4(DEMO)---ACSADMIN(SingleMode)
File
Edit
View
Operation
Tools
Help
SingleMode
Definition
Data
Status
Timing
Groups
1
2
3
4

ITM: BVDSS_KI26XX_ITM (26xx SMU)

Mode
● DC Only
○ Pulse Available

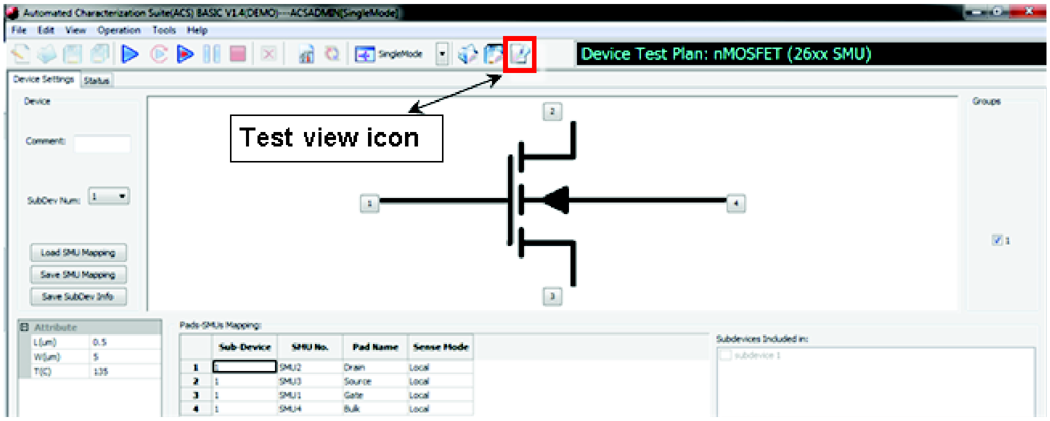
□ Stop on Compliance

☑ 1

Device Num	SMU	Pad	Function	Force Mode	Source	Measure	Compliance	Meas.Range	Advanced
1	SMU2	Drain	Bias I	DC	1e-009	V	20.0	auto	...
	SMU3	Source	GND						
	SMU1	Gate	GND						
	SMU4	Bulk	GND						

1. In the device view GUI, click the **Test view** icon to return to the test view GUI (see next figure).

Figure 61: Device view GUI



Device Test Plan: nMOSFET (26xx SMU)

Comment:

SubDev Num: 1

☑ 1

Attribute	
L(um)	0.5
W(um)	5
T(C)	135

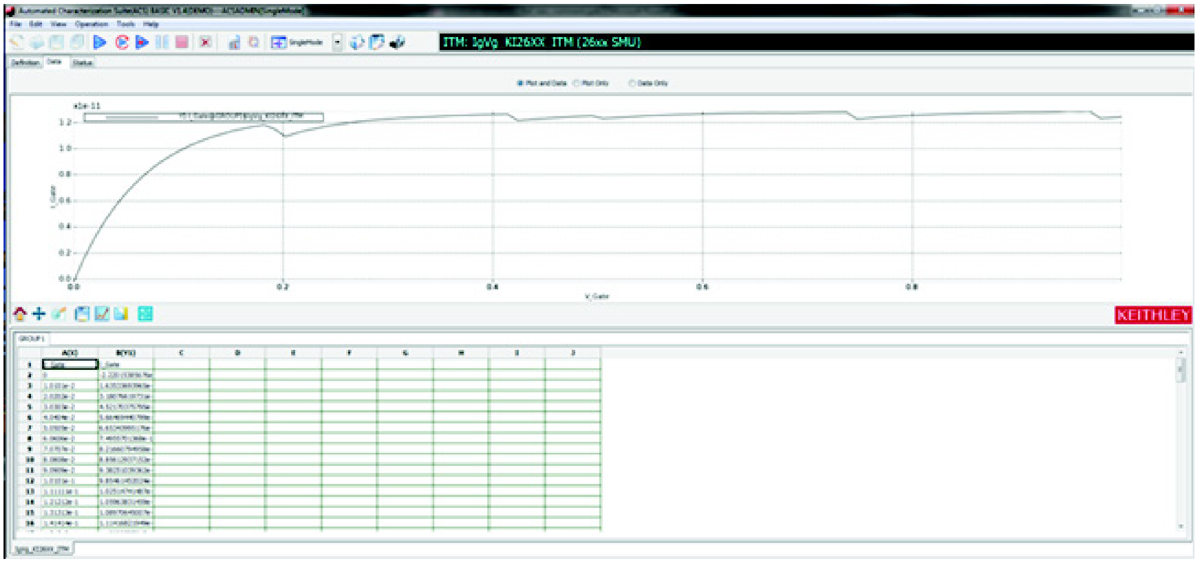
	Sub-Device	SMU No.	Pad Name	Sense Mode
1	1	SMU2	Drain	Local
2	1	SMU3	Source	Local
3	1	SMU1	Gate	Local
4	1	SMU4	Bulk	Local

Run the test module

Once the test setup is completed, click the **Run** icon ▷. For more information on running the test module, refer to Run individual tests.

The test results appear in the Data tab (see next figure). Refer to Data plot for details on the Data tab.

Figure 62: Test Data tab

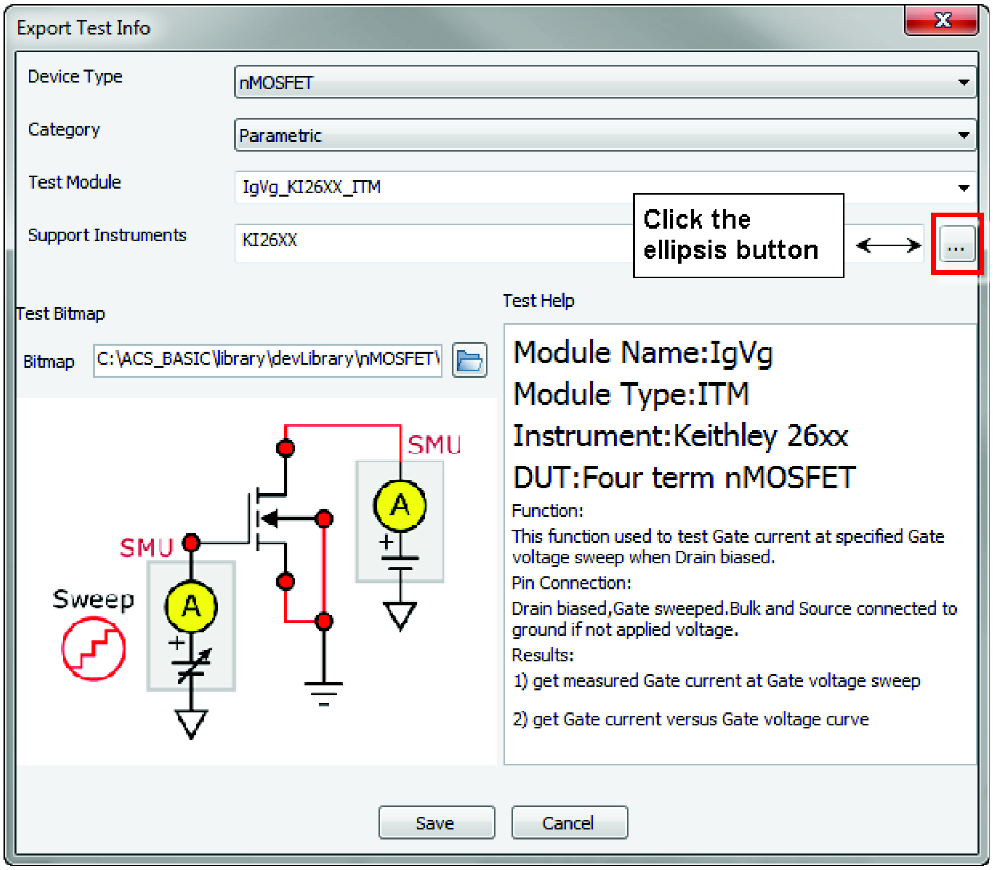


Save the test module to the library

The test module can be saved to the device library for future use. The saved information includes both test settings and test results.

Click the **Export Test** icon on the toolbar; the save test information GUI opens (see next figure).

Figure 63: Save Test Info dialog box



Export Test Info

Device Type: nMOSFET

Category: Parametric

Test Module: IgVg_KI26XX_ITM

Support Instruments: KI26XX

Click the ellipsis button

Test Bitmap

Bitmap: C:\ACS_BASIC\library\devLibrary\nMOSFET\

Test Help

Module Name:IgVg
Module Type:ITM
Instrument:Keithley 26xx
DUT:Four term nMOSFET
Function:
This function used to test Gate current at specified Gate voltage sweep when Drain biased.
Pin Connection:
Drain biased,Gate sweeped.Bulk and Source connected to ground if not applied voltage.
Results:
1) get measured Gate current at Gate voltage sweep
2) get Gate current versus Gate voltage curve

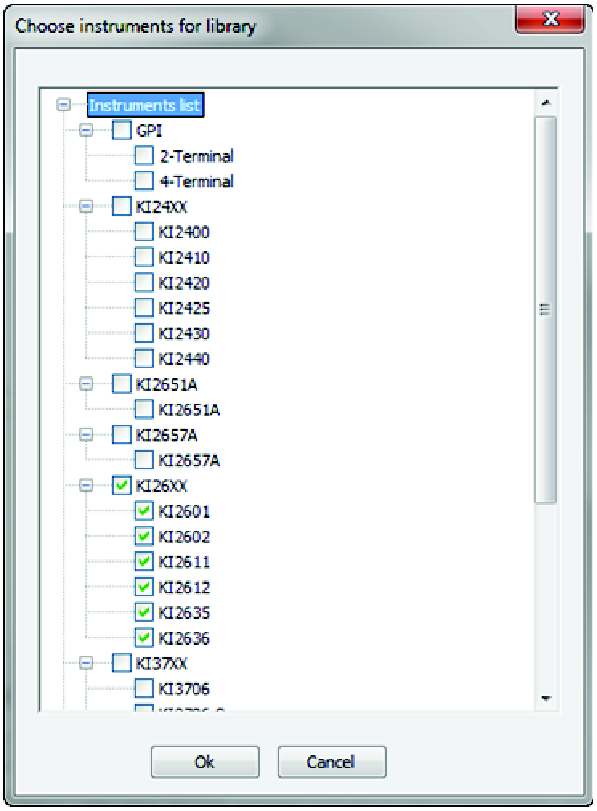
a. Select a device type for the module in the Device Type drop-down list.

b. Select a test category in the Category list.

c. Input a test name in the Test Module edit box. Note that the new Test Module name should be different from the existing name to avoid overwriting the original module.

d. Click the ellipsis button (see previous figure) of the Support Instruments edit box. In the pop-up dialog box choose your desired instruments for the library by clicking the box next to the instrument (see next figure). Click **OK**.

Figure 64: Save selected instruments for testing



Choose instruments for library

- Instruments list
  - ☐ GPI
    - ☐ 2-Terminal
    - ☐ 4-Terminal
  - ☐ KI24XX
    - ☐ KI2400
    - ☐ KI2410
    - ☐ KI2420
    - ☐ KI2425
    - ☐ KI2430
    - ☐ KI2440
  - ☐ KI2651A
    - ☐ KI2651A
  - ☐ KI2657A
    - ☐ KI2657A
  - ☑ KI26XX
    - ☑ KI2601
    - ☑ KI2602
    - ☑ KI2611
    - ☑ KI2612
    - ☑ KI2635
    - ☑ KI2636
  - ☐ KI37XX
    - ☐ KI3706

e. If desired, change the test bitmap by clicking on the folder icon to find a new bitmap.

f. If desired, change the description by entering the text in the right editing panel.

Click **OK** to save the test module to the test library. After the test module is saved, ACS Basic returns to the Open Test GUI where the previous test can be opened and used again.

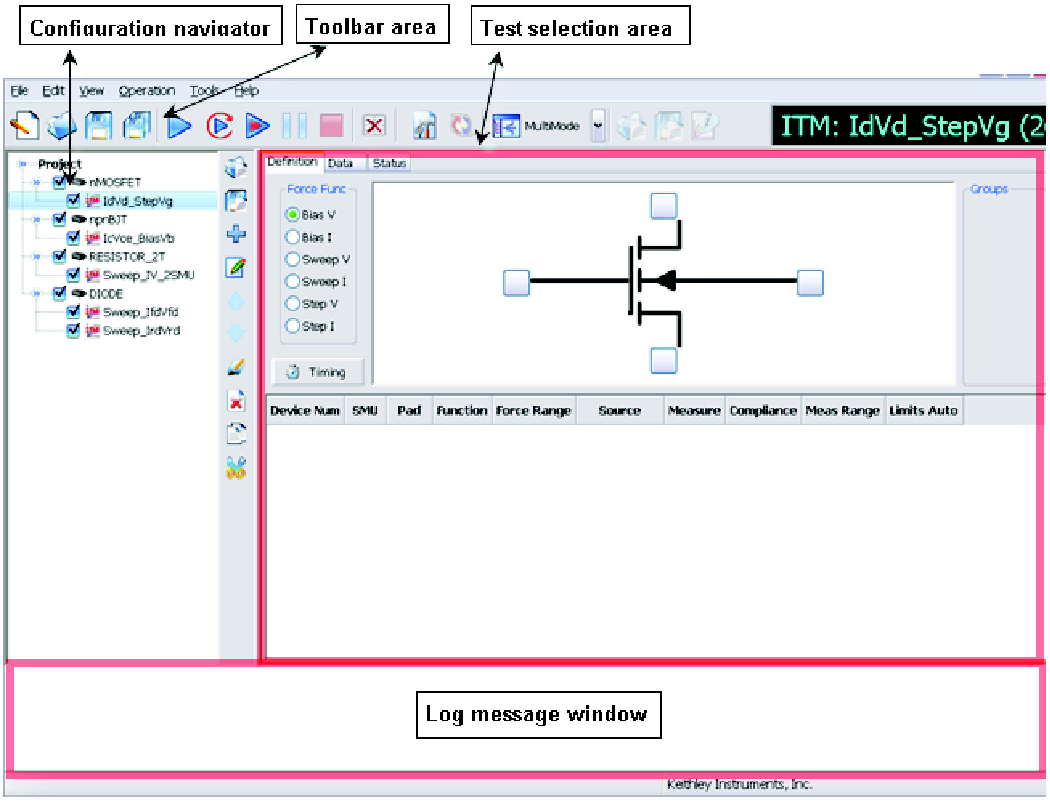
Multi-test mode

Introduction

Multi-test mode uses a project to organize devices and tests. The configuration navigator (see next figure) has the following logical hierarchy:

- Project
  - Device 1
    - Test module 1 (ITM, STM, or PTM)
    - Test module 2 (ITM, STM, or PTM)
  - Device 2
    - Test module 1 (ITM, STM, or PTM)
    - Test module 2 (ITM, STM, or PTM)

Figure 65: Multi-test mode GUI



The configuration navigator is the primary interface for building, editing, and viewing a project plan, and for specifying and accessing each project plan component.

When you select a navigator component (test or device) you can do the following:

- Add a new component
- Delete an existing component
- Run the tests associated with this component

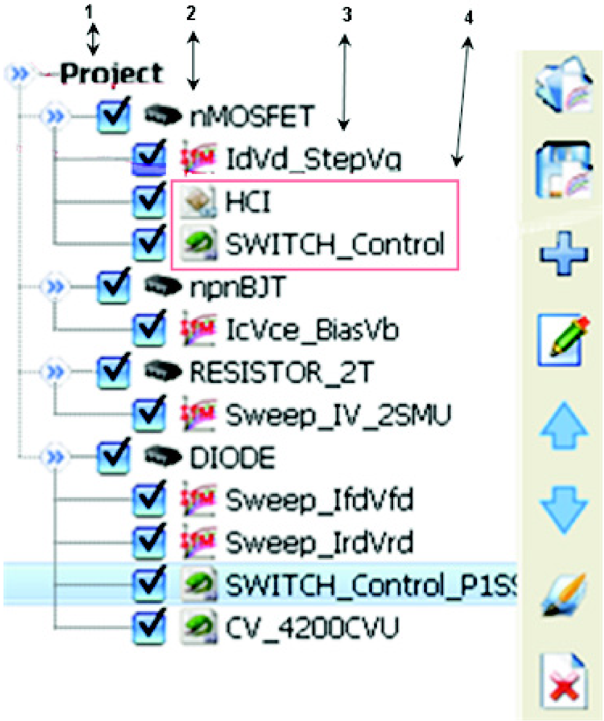
Clicking on a navigator component opens the configuration interface on the right panel that includes settings, test results, and status information.

The next figure displays the typical project plan components that are displayed in the configuration navigator. The following explanation of the numbered items correspond to the numbers in the the next figure:

1. **Project** Plan: Defines and sequences all devices tested, and all tests/operations to be performed at each. It typically corresponds to one die on a wafer.
2. **Device** Plan: Defines and sequences all tests for a specific device, including transistors, diodes, and resistors.
3. **ITM**: Completely defines a parametric test without programming, using a series of easily configured graphical user interfaces which show data numerically and graphically in real time. Provides for the display of both raw data and analyzed data.
4. **STM/PTM**: Defines an operation; an external instrument operation, using a TSP or PTM user module. These test modules are connected to the STM and are configured with user-supplied parameter values (several STMs may be associated with the with the same user module). Configuration is done using a supplied library or may be created by the user with Script Editor's simple graphical user interface.

For details about building a project plan using the configuration navigator, refer to the Project plan topic.

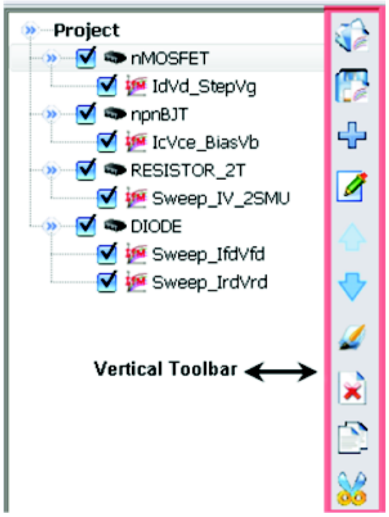
Figure 66: Configuration navigator



Vertical toolbar

The vertical toolbar is only accessible in Multi-test mode. See the next figure for the location in the Multi-test mode GUI:

Figure 67: Vertical toolbar location



Project
- ☑ nMOSFET
  - ☑ IdVd_StepVg
- ☑ npnBJT
  - ☑ IcVce_BiasVb
- ☑ RESISTOR_2T
  - ☑ Sweep_IV_2SMU
- ☑ DIODE
  - ☑ Sweep_IfdVfd
  - ☑ Sweep_IrdVrd

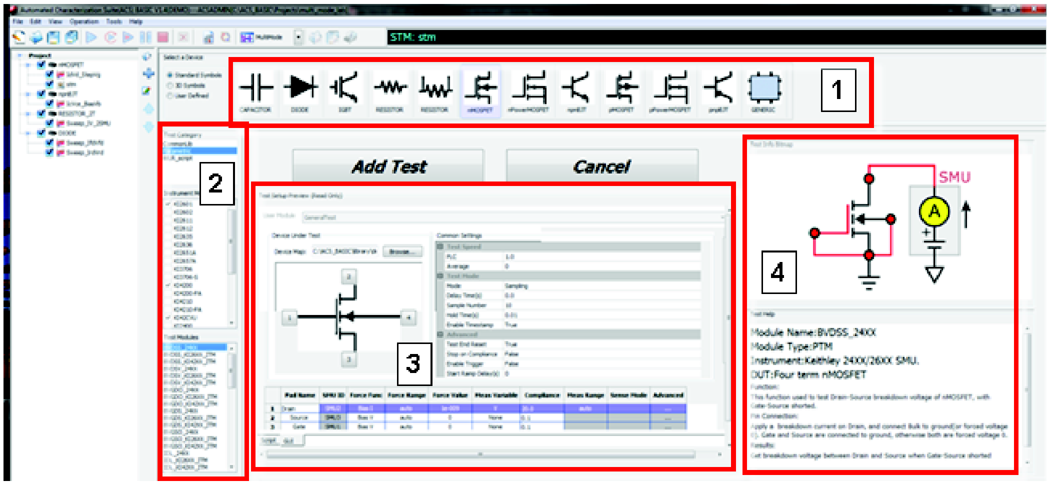
Multi-test mode GUI

When ACS Basic is opened in Multi-test mode, you will see the configuration navigator and the work area that shows the settings, test results, and status information.

The test selection GUI of the Multi-test mode is shown in the next figure. Note that there are four main sections:

1. Select a Device
2. Test Category - select a test module
3. Test Setup Preview
4. Test Information

Figure 68: Multi-test mode work area



Module Name:BVDSS_24XX
Module Type:PTM
Instrument:Keithley 24XX/26XX SMU.
DUT:Four term nMOSFET

Build a project plan

Create a new project:

1. Click the **New** icon on the toolbar (or you can click **New** in the File menu). The New Test Project dialog box opens.
2. Enter the following information:
   - **Template**: Use to load an existing project template. Enter the storage folder name or use the Browse function to find the location. This field may be left blank.
   - **Project Name**: Enter the new project name (this is required).
   - **Project Directory**: Select a directory where you want to save the project, or use the default directory. Use the Browse function to find a location.
   - **Technology**: Optional device information.
   - **Product**: Optional device information.
3. Click **OK** to complete the new test project.

When you open ACS Basic and a test project, the project directory path displays in the title bar of the ACS Basic GUI. All project information is saved to this directory.

Add a test module to the project (see Select a test module in single-test mode for more details):

1. Select a device view (Standard, 3D Symbols, or User Defined).
2. Click on the desired device and the selected icon will highlight.
3. All test categories belonging to this device are listed in the Test Category area. Select your desired category.
4. Select the test instrument model. The test modules applicable to this instrument are listed in the Test Modules area. If two or more instruments are selected the Test Modules area will display all of the modules available to any of the selected instruments.
5. Select a test module.
6. The test information preview is displayed on the right panel. This information includes Test Setup Preview, Test Information Bitmap, and Test Help.
7. Click the Add Test function to add the test module to the configuration navigator.

If the test module imported belongs to the same kind of device, as an existing device in the configuration navigator, the module will be added to the existing device automatically.

If the imported test module does not belong to the same kind of device, a new device node will be added to the configuration navigator.

Configure the test module:

After the test module is added, click the device in the configuration navigator to view and edit test details.

Execute individual device plans

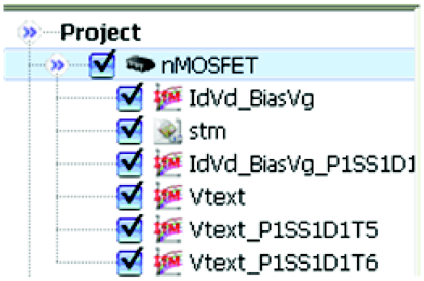
After a test plan has been built, you can execute individual parts of a project plan at the test module level, the device level, or the project level.

NOTE

Each time you execute a test or test sequence using the Run icon ▶, the data from each test is inserted to its own Data worksheet. Each new run updates this worksheet (for more information on worksheets, refer to the Data plot topic. You can also generate appended worksheets for tests and test sequences. Refer to the Append and clear append topic.

You can execute an individual Device Plan with the selected components, ITMs, PTMs, ans STMs, assigned to it in the order that they appear in the configuration navigator. In the configuration navigator, select the box corresponding to the device to run. For example, in the next figure, the nMOSFET has been selected.

Figure 69: Select the device to run



- Project
  - ☑ nMOSFET
    - ☑ IdVd_BiasVg
    - ☑ stm
    - ☑ IdVd_BiasVg_P1SS1D1
    - ☑ Vtext
    - ☑ Vtext_P1SS1D1T5
    - ☑ Vtext_P1SS1D1T6

1. Enable the tests assigned to this device by checking the box next to the test name; or disable a test by un-checking the box.
2. To save the plan, click on the Save icon 💾 in the ACS Basic toolbar, or select the Save function in the ACS Basic File menu.
3. Start execution by clicking the Run icon ▶ in the toolbar. The Run toolbar icon turns gray while the test executes, and the Abort Test/Plan icon illuminates for the duration of the test. The test sequences will run one at a time.

NOTE

Selecting only an individual ITM, STM, or PTM on the test tree will run only that specific test.

Trace mode

Introduction

Trace mode provides interactive control of SMU instruments to achieve real-time performance. Only Series 2600B instruments (Models 2601B/2602B, 2611B/2612B, 2635B/2636B), Series 2650A (Models 2651A, 2657A) and Series 2400 instruments (Models 2400, 2401, 2410, 2420, 2425, 2440) are supported. ACS Basic can perform DC and pulse tests on Series 2600B, and 2650A instruments in Trace-mode, but it can only perform DC measurements on Series 2400 instruments.

Trace-mode allows you to quickly and easily setup the instrument for fast, visual results and save the data for later comparison or analysis

NOTE

Trace-mode only supports the Series 2400, Series 2600B, and the Models 2651A/2657A. However, you cannot connect a Series 2400 and a Series 2600B instrument to the same device. If you are using a Series 2400 on one device terminal, you must use a Series 2400 instrument on the other device terminal(s). However, you can use a Series 2600B and a Model 2651A instrument in any combination. Also, Trace-mode only supports the SCPI protocol for Series 2400 SMUs. For Series 2600B, it only supports the GPIB communication mode to the master and does not support LXI communication to the master.

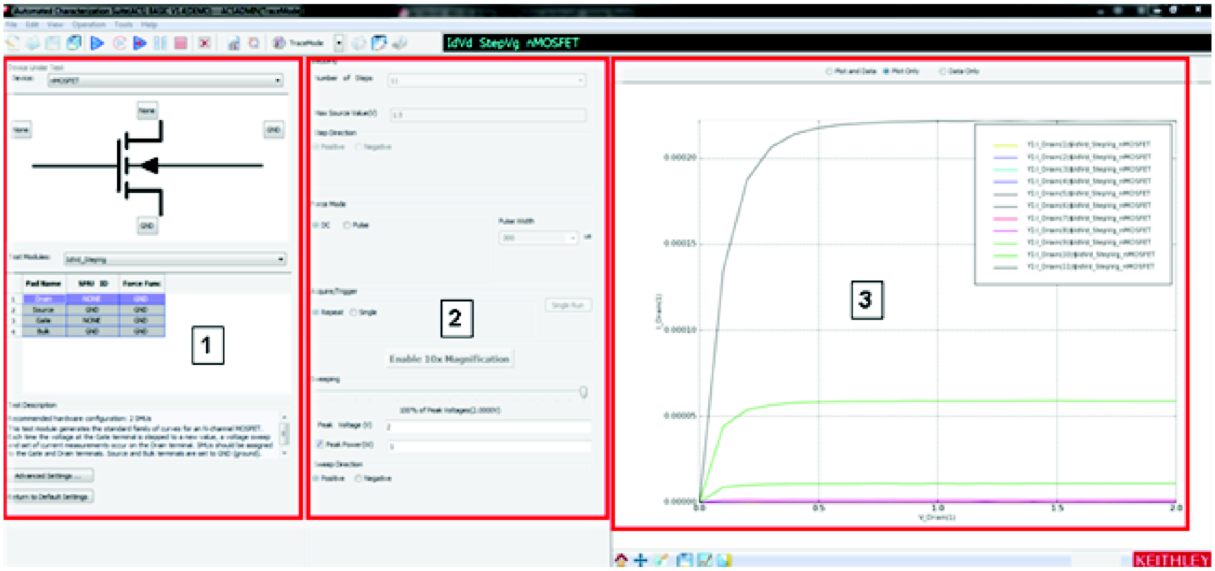
Test GUI

Select **TraceMode** after clicking the drop-down arrow from the operation toolbar at the top of the ACS Basic GUI. The Trace-mode GUI is shown in the next figure.

The GUI includes:

1. Device setting: used to map the SMU, select the test module, and set the advanced settings.
2. Test setting: used to set the test information when the test module is enabled.
3. Trace Test Data: used to show the test result.

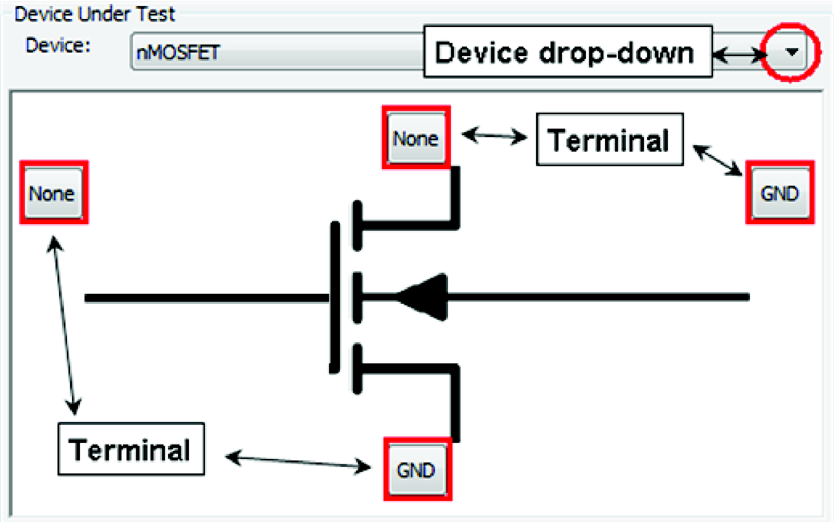
Figure 70: Trace-mode test GUI



Open a test module in Trace mode

1. Select a device in the Device drop-down list (in the Device Under Test area)(see next figure).

Figure 71: Device Under Test area

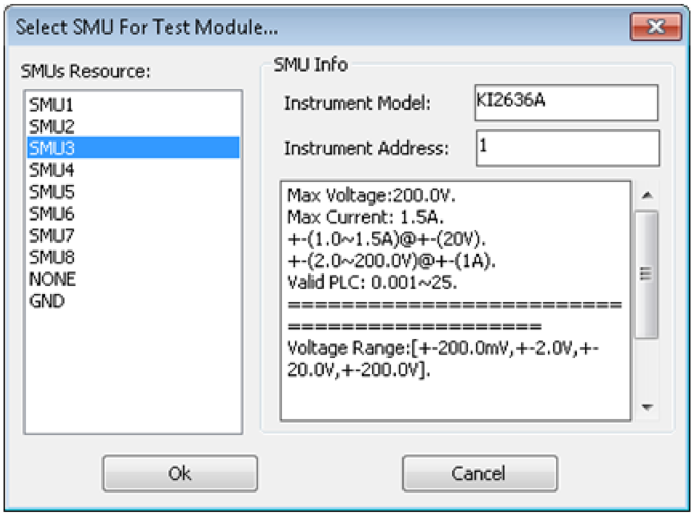


Device Under Test

Device: nMOSFET

2. Select a test module for the device. All test modules belonging to this device are listed in the Test Modules drop-down list.
3. Connect an SMU to the desired device terminals.
   a. Click the button next to the device terminal and the Select SMU For Test Module dialog box opens (see next figure).
   b. Select a desired SMU resource for the terminal in the SMUs Resource box. The SMU Info box indicates the Instrument Model and Instrument Address.
   c. Click **OK** to complete the SMU setting.

Figure 72: Select SMU For Test Module



Select SMU For Test Module...

SMUs Resource:

SMU1
SMU2
SMU3
SMU4
SMU5
SMU6
SMU7
SMU8
NONE
GND

SMU Info

Instrument Model: KI2636A

Instrument Address: 1

```
Max Voltage:200.0V.
Max Current: 1.5A.
+-(1.0~1.5A)@+-(20V).
+-(2.0~200.0V)@+-(1A).
Valid PLC: 0.001~25.
=========================
====================
Voltage Range:[+-200.0mV,+-2.0V,+-
20.0V,+-200.0V].
```

NOTE

Series 2400 instruments can only be used with other Series 2400 instruments connected to the same device. Also, Series 2600B instruments can only be used with other Series 2600B or Model 2651A and 2657A instruments connected to the same device. Additionally, when a Model 2651A or 2657A is connected to the drain of a MOsFET or collector of a BJT or IGBT, the Series 2600B cannot be connected to the Source terminal or Emitter terminal. These terminals must be connected to GND. This is to prevent damaging the Series 2600B SMUs with high current from the Model 2651A or damaging the 2657A with high current from the Series 2600B. Also, if there is more than one group of Series 2600B SMUs in the system, only the SMUs in group one will appear in the SMU Resource box.

Configure Trace mode

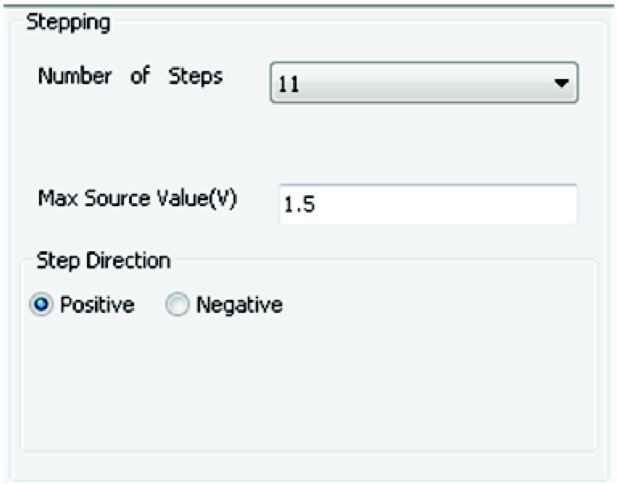
NOTE

ACS Basic has default settings for Trace mode. You can change the settings as needed. If you decide to use the default settings, you can click the **Return to Default Settings** function which is found under the Test Description.

Step setting

See the next figure for an example of the Stepping variables.

Figure 73: Stepping variables



Stepping

Number of Steps: 11

Max Source Value(V): 1.5

Step Direction

● Positive ○ Negative

NOTE

Not all test modules have the step test. Only tests that include Step V or Step I as a force function have the "Stepping" panel enabled. The default setting is determined if the current test type is sweep, on the Collector or Drain terminal, and step, on the Base or Gate terminal.

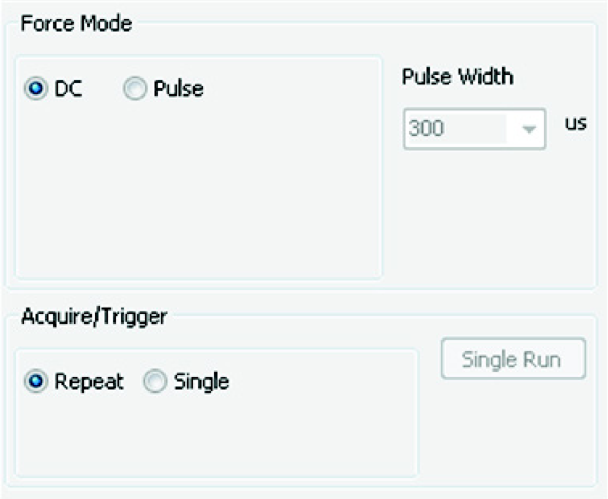
- **Number of steps**: for testing the step voltage measurements. Valid from 1 to 11. If the input is 1, the step terminal will be applied a constant bias at the Max Source Value.
- **Max Source Value**: is the stop amplitude value of the step; it’s always positive. If the test is step V, the units are in volts; if it’s step I, the units are in amps.
- **Step Direction**: indicates the direction in which the source value of the Step SMU changes. When set to positive, the step test executes from 0 to maximum; when negative, from 0 to negative maximum.

For more information on step and sweep definitions, refer to the Configure an SMU's Bias/Sweep/Step table topic.

Force Mode (for sweeping SMU)

See next figure for an example of possible Force Mode functions.

Figure 74: Force setting box



Force Mode

● DC ○ Pulse

Pulse Width 300 us

Acquire/Trigger

● Repeat ○ Single

- **DC/Pulse**: is applied to both the stepping and sweeping SMUs. If pulse mode is selected, the Pulse Width input box will be enabled.

NOTE

If the sweeping SMU is a Series 2400, only DC mode can be applied. For a combined 2651A SMU, only Pulse mode can be applied.

- **Pulse width**: The width of the pulse in pulse mode. The units are microsecond.
- **Acquire/Trigger mode**: Select Single to perform the test once. Select Repeat to perform the test until Stop is selected. When Single is selected the Single Start button is enabled.

Pulse period defaults to the maximum allowable value. You can adjust the duty cycle in the Advanced Settings GUI. Refer to Advanced Setting Preferences for more information.

For the Series 2600B SMU (non-2651A), the maximum pulse width is 1000μs.

For the 2651A SMU, if current is in the 10A to 50A range, and voltage is in the 10V to 40V range, the maximum pulse width is 300μs. Otherwise, the maximum pulse width is 1000μs.

For the 2651A SMU 100A/80V combined SMU, if voltage is higher than 10V, the maximum pulse width is 300μs. Otherwise, the maximum pulse width is 1000μs.

Pulse mode cannot be used on Series 2400 SMUs in Trace mode.

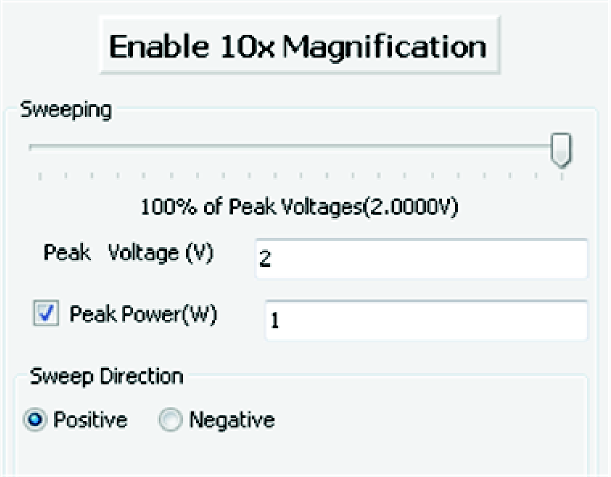
⚠ WARNING

In Single Trigger mode, the SMU output remains on after each test at zero volts. If you want to change connections or insert a new device, first click Stop. When the test has stopped, the SMU output is disabled.

Sweep control setting

The next figure shows the Sweeping control settings.

Figure 75: Sweep variables



100% of Peak Voltages(2.0000V)

Peak Voltage (V) 2

☑ Peak Power(W) 1

Sweep Direction

● Positive ○ Negative

- **Enable 10X Magnification**: allows you to toggle between enable or disable and is only available during testing. When enabled, the step value will decrease by 10 times, and the function will change to "Disable 10x Magnification." You can disable by selecting the function again.

Sweeping setting

- **Sweep slider**: controls the actual final value of the sweeping SMU. The slider position is a percentage of the Peak Value. There are always 21 points in the sweep. For example, sliding the sweep slider is similar to turning the knob on a curve tracer. Sliding the slider to the right increases the maximum value while sliding to the left decreases the maximum value. The sweep slider can be controlled in the following ways:
  - Use your mouse to drag the slider. It has a resolution of 1% of the Peak Value.
  - Use your mouse wheel to control the slider. Move the wheel down moves the slider to the left, and move the wheel up moves the slider to the right. It has a resolution of 0.5% of the Peak Value.
  - Use the left and right arrow keys on your keyboard to control the slider. It has a resolution of 0.1% of the Peak Value.
  - Use the up and down arrow keys on your keyboard to control the slider. It has a resolution of 1% of the Peak Value.
- **Peak Voltage**: is the absolute maximum stop value for the sweeping SMU. The actual stop value depends on the current position of slider. The units are volts.
- **Peak Power**: sets the maximum power output at any point of the sweep. Current compliance is calculated at each point of the sweep by peak power/programmed voltage level.

NOTE

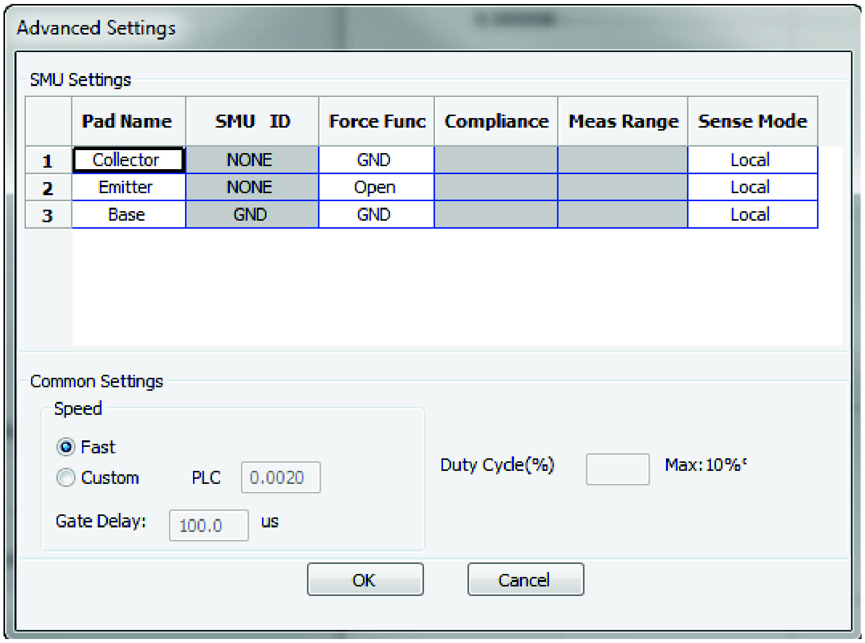
Power compliance only affects the SMU designated as the sweeping SMU. For instance, in the IdVg_StepVd test, the SMU assigned to the gate is the sweeping SMU. Therefore, power compliance limits the maximum power to the gate terminal of the FET, not the drain terminal.

Advanced Settings

The Advanced Settings can be used to change the measure range, number of power line cycles (NPLC), and other settings

Click the **Advanced Settings** function and a dialog box opens (see next figure).

Figure 76: Advanced Settings dialog box



Advanced Settings

SMU Settings

	Pad Name	SMU ID	Force Func	Compliance	Meas Range	Sense Mode
1	Collector	NONE	GND			Local
2	Emitter	NONE	Open			Local
3	Base	GND	GND			Local

Common Settings

Speed

● Fast

○ Custom PLC 0.0020

Gate Delay: 100.0 us

Duty Cycle(%) Max: 10%

There are several settings that you can adjust:

- **Pad Name**: can be edited here, but will be disabled in the test setup GUI.

NOTE

You will get a Warning message if you change a pad name. The warning states that changing the pad name will cause a Pad/SMU mismatch of all related test modules in the library.

- **SMU ID**: is disabled here. It can only be assigned in the device map field in the test setup GUI.
- **Force Func**: allows you to select SweepV/StepV/StepI/GND/Open from a drop-down list. This column will be disabled in the test setup GUI.
- **Compliance**: sets the compliance value for a given SMU. The value is limited by the specifications of the SMU. Note that this specification is not the same in Pulse and DC mode. If the compliance value is only valid in Pulse mode, you will be unable to select DC as the Force Mode.
- **Meas Range**: will match the force function if the measure function matches the force function. The measurement range also affects the Force Mode. If the measurement range selected is only valid in Pulse mode, then DC will be disabled in the Force Mode field. If a measurement is not assigned to the corresponding pad, the Meas Range cell is disabled.

NOTE

Current measurement ranges in Trace mode may be restricted based upon the SMU to ensure a fast and interactive response in Trace mode. Should more sensitive measurement ranges be required, use Single-test Mode or Multi-test Mode. For example, for the Model 2636A, the minimum settable range is 100nA. Therefore, the 10nA, 1nA and 100pA ranges will not be accessible in Trace mode.

- **Speed**: sets the measurement NPLC to Fast or Custom. For Series 2600B SMUs, fast speed is defined as 0.002PLC, for Series 2400 SMUs, it is defined as 0.01. Refer to the The Speed area to learn more about the speed area in the common settings.

A custom PLC should be limited according to the SMU's specifications. When you setup a custom PLC, the pulse width in the test setup GUI should be larger than 2*PLC/local frequency.

To achieve fast speeds, a PLC value less than 0.1 is recommended.

- **Gate Delay**: sets the delay time after gate bias has been applied. This setting will ensure that the gate bias is applied before the drain voltage or current bias. When the Force Mode is Pulse and the test device is a MOSFET or IGBT, the pulse period and the gate pulse width will increase by the amount specified in the input value. If the value is set to a value larger than 50% of the pulse width, then the Gate delay will be set to 50% of the pulse width.
- **Sense Mode**: sets the sense mode of the SMU to either Local or Remote. The test modules for IGBTs are all set to remote sense to guarantee measurement accuracy which is required in high current tests.
- **Duty Cycle**: displays the maximum allowable duty cycle and the currently selected duty cycle. By default, the duty cycle is set to the maximum possible. Adjust this value to reduce the duty cycle.

	Maximum current	Maximum voltage	Maximum duty cycle
Model 2651A SMU	20A	40V	10%
	50A	20V	10%
	50A	40V	1%
Model 2651As combined in parallel	100A	18V	10%
		36V	1%
Model 2651As combined in series	45A	20V	10%
		80V	1%

For Series 2600B SMU (non-2651A), the maximum duty cycle is adjusted by the current compliance or max source value. If the current value applied for tests requires more than 0.1 A, the max value is limited at 1%.

After configuring the Advanced Settings, click **OK**.

The Advanced Settings items are the same as in an ITMs in Single or Multi-test mode. For more information on how to set the Force Func, Compliance, Meas Range, and other items in the Advanced Settings, refer to the Configure an SMU's Bias/Sweep/Step table topic.

Run the test module

After the test is configured, click the **Run** icon ▷.

The trace mode test will run repeatedly or once by clicking Single Run (for details about repeat and single testing, refer to the Force Mode topic).

The test result (plot and data) are displayed in real time in the Test Data area. If testing is set to Repeat, the test data is refreshed repeatedly to show the latest test result. If testing is set to Single, the sweeping test will run when you click the Run icon ▷ and the test result is displayed in the Test Data area. Each time you click the Run icon ▷, the test will run and the results will display in the Test Data area.

When the trace mode test is running, you can use the sweep slider to adjust the stop value of the sweep in real time. When you adjust the slider to a certain percentage, such as 60% of the peak value (for example 10V), the next time sweep will run from start to 60% of 10V through 6V which will become the stop value. And the Test Data area will immediately show the results of new sweeping test. Click the Stop icon 🟥 to end the trace mode test.

Save a test module to the library

The test modules for Trace mode can be saved to the device library. The saved information includes both test settings and test results.

1. Click the Save Test icon 💾 on the toolbar; the save test information dialog box opens.
   - Note that the new Test Module name should be different from the existing name to avoid overwriting the original module.
2. Click **OK**. All your changes will be saved to the library.
3. After the test module is saved, ACS Basic returns to the Open Test GUI.

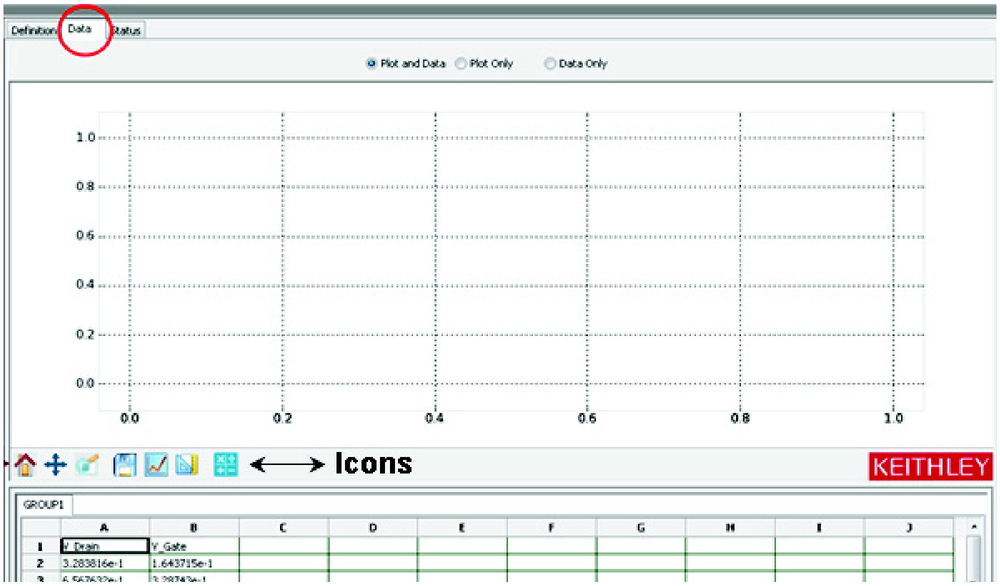
Data plot

Open a graph sheet

The Graph sheet is integrated with the Data sheet. There are two ways to open a Graph sheet:

Option 1: When setting up an ITM, STM, or PTM, you can switch to the Graph sheet of the current test by clicking the **Data** tab. Results of the current test are displayed, and graphs can be drawn according to the results. The next figure shows a graph sheet of a test with a data sheet at the bottom.

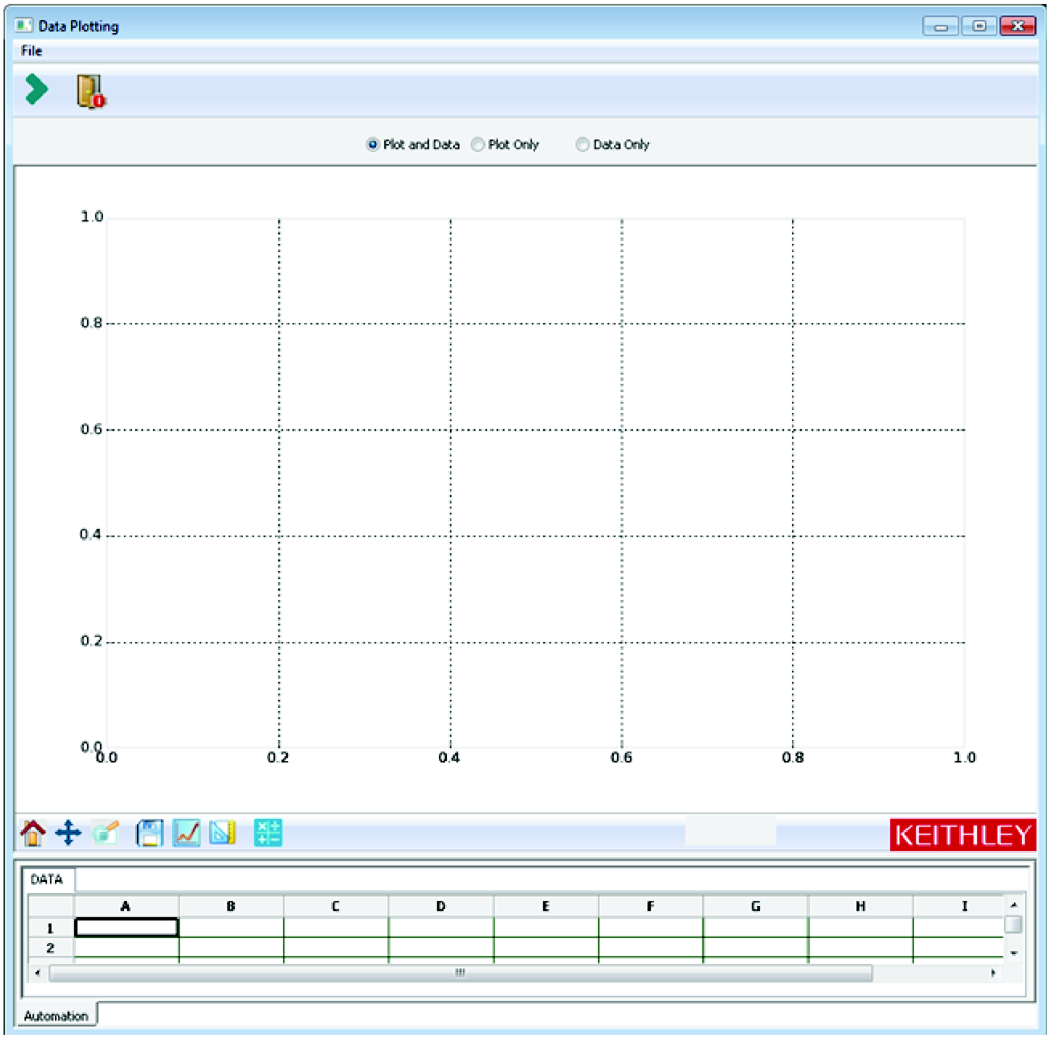
Figure 77: Un-configured graph sheet of a test



	A	B	C	D	E	F	G	H	I	J
1	V_Drain	V_Gate								
2	3.283816e-1	1.643715e-1								
3	6.567632e-1	3.28743e-1								

Option 2: Click **Tools** on the menu bar, then choose Offline Data Plotting and the Data plotting tool dialog box will open (see next figure). Select a blank Data Plotting sheet similar to the one shown in the previous figure. Data and graphs on this sheet are not limited to the current test or the current project. Click the import icon to import data. Click the exit icon to exit the page.

Figure 78: Data plotting tool



● Plot and Data ○ Plot Only ○ Data Only

Control the properties of the graph

Items in the Graph Settings menu and the Graph toolbar control the properties of the graph.

Toolbar icons

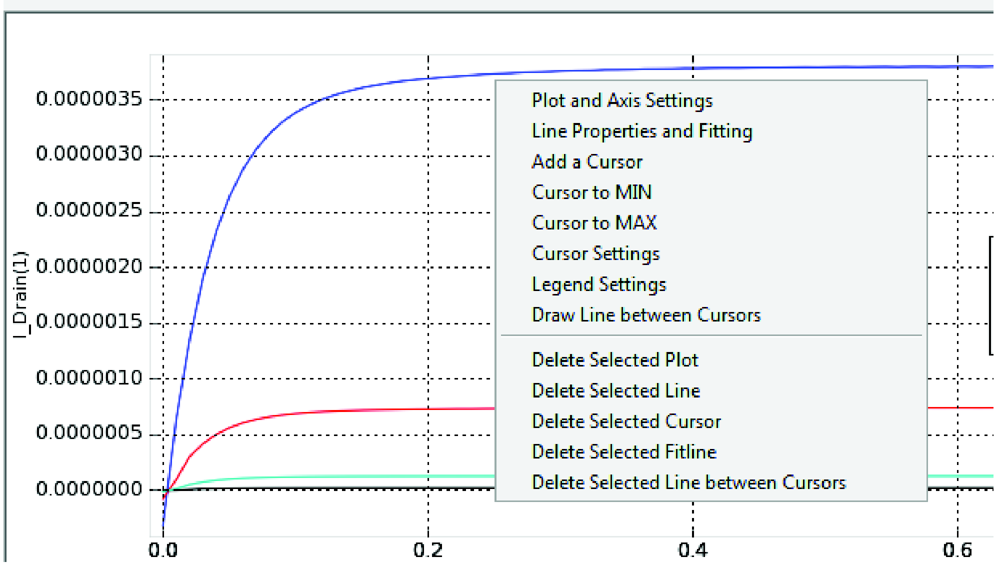
The toolbar is located between the graph section and the data section and contains seven icons:

- **Home**: is used to return the graph to its original size (after zoom).
- **Pan**: permits panning of the graph.
- **Zoom**: enlarges a small, selected part of the graph.
- **Save**: saves the plot as a separate graphics file. Click the Save icon 💾 to display the Save to file dialog box. You can save the plot as a .png, .ps, .eps, .csv, or .svg format.
- **Plot**: creates a graph out of the data selected (see the Define a graph and plotting topic for more information).
- **Plot Auto Scale**: auto scales all the graphs on the graph sheet.
- **Formular**: performs data calculations on test data, as well as on the results of other Formulator calculations (see the Formulator topic for more information).

Access the graph settings menu

To view the Graph Settings menu, right-click on any portion of the graph (see next figure).

Figure 79: Graph settings menu



The Graph Settings Menu functions are summarized below:

Plot and Axis Settings: sets general properties of the selected graph and the parameters of the axis. Many sub-functions are provided. For more information, refer to the Define a graph and plotting topic.

Line Properties and Fitting: sets the properties of a selected line and makes the fitting if the cursors are set. For more information, refer to the Define a graph and plotting topic.

Add a Cursor: adds a floating cursor to the location where you have your mouse pointer.

Cursor to MIN: moves the selected cursor to the minimum point of the attached curves.

Cursor to MAX: moves the selected cursor to the maximum point of the attached curves.

Cursor Settings: adds a cursor to a graph and sets the cursor properties. For more information, refer to the Define a graph and plotting topic.

Draw Line between Cursors: draws a linear line between two selected cursors.

Delete Selected Plot: deletes the plot below where the menu is displayed from the graph sheet. Also, all other plots will get resized.

Delete Selected Line: deletes a selected line from the plot. To delete a specific line, right-click the line and choose Delete Selected Line from the menu.

Delete Selected Cursor: deletes the selected cursor. To delete a cursor, right-click the cursor and choose Delete Selected Cursor in the menu.

Delete Selected Fitline: deletes the selected fitline cursor. To delete a fitline, right-click the fitline and choose the Delete Selected Fitline in the menu.

Delete Selected Line between Cursors: deletes the selected line between the two cursors. To delete a line between cursors, right-click the line and choose the Delete Selected Line between Cursors in the menu.

Define a graph and plotting

There are two steps to define a graph and plot the data:

1. Define the data to be graphed.
2. Plot the data.

Define the data to be graphed

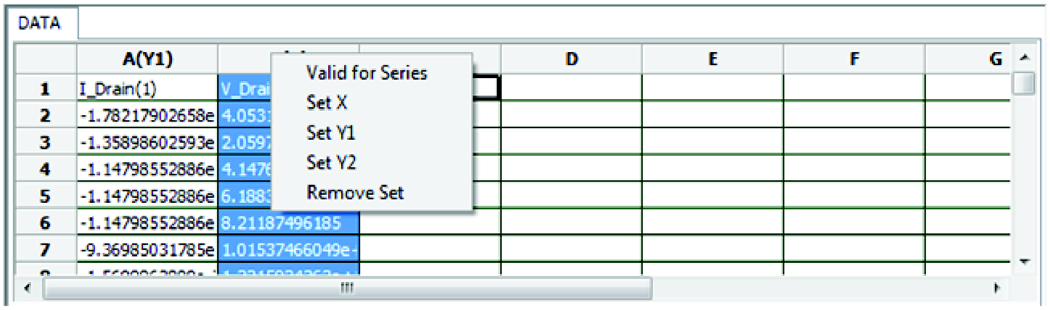
There are two steps to define and graph data:

1. Assign the data to an axis.
2. Assign the axis to a plot.

Assign data to an axis

Right-click on the title of any column in the data sheet to display the graph definition menu (see next figure).

Figure 80: Graph definition menu



	A(Y1)			D	E	F	G
1	I_Drain(1)	V_Drai					
2	-1.78217902658e	4.053					
3	-1.35898602593e	2.059					
4	-1.14798552886e	4.147					
5	-1.14798552886e	6.188					
6	-1.14798552886e	8.21187496185					
7	-9.36985031785e	1.01537466049e-					

Graph definition menu descriptions:

Valid for Series: is used when your test is multi-step and there are a series of results (see previous figure). If the Valid for Series command is selected, the settings for any column are valid for all of the other columns in the same series. For example, if column B (I_Drain(1)) is set as X, column D (I_Drain(2)) will also be set as X.

Set X: sets the selected column as the X Axis on the graph.

Set Y1: sets the selected column as the Y Axis on the left side of the graph.

Set Y2: sets the selected column as the Y Axis on the right side of the graph.

Remove Set: deletes the selected column from the plot.

To draw a graph, at least one X and one Y should be set.

NOTE

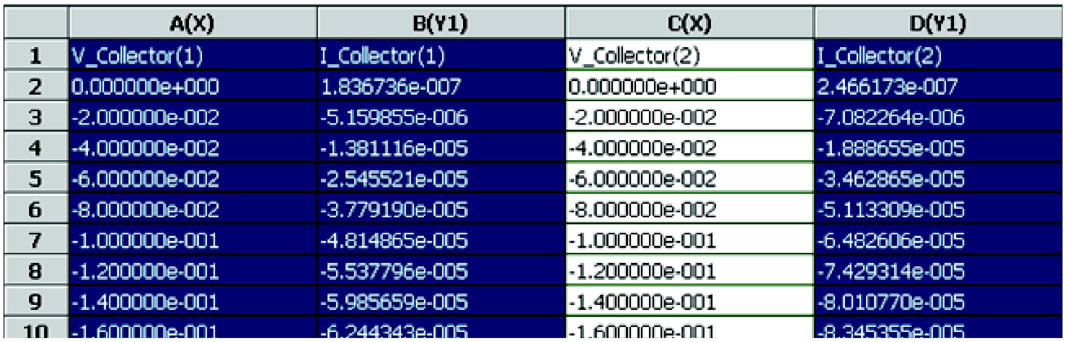
ACS Basic supports multi-plot using one data sheet. This allows different columns to be plotted to different graphs.

Assign the axis to a plot

After an axis is set, you must choose the data you want plotted on the graph. To choose data, click the title of a column (the column will highlight).

To choose more than one column, drag the mouse over the desired columns, or execute a Ctrl + click on the desired column. At least one X and one Y axis (either Y1 or Y2) must be selected. The next figure shows the results of choosing multiple columns.

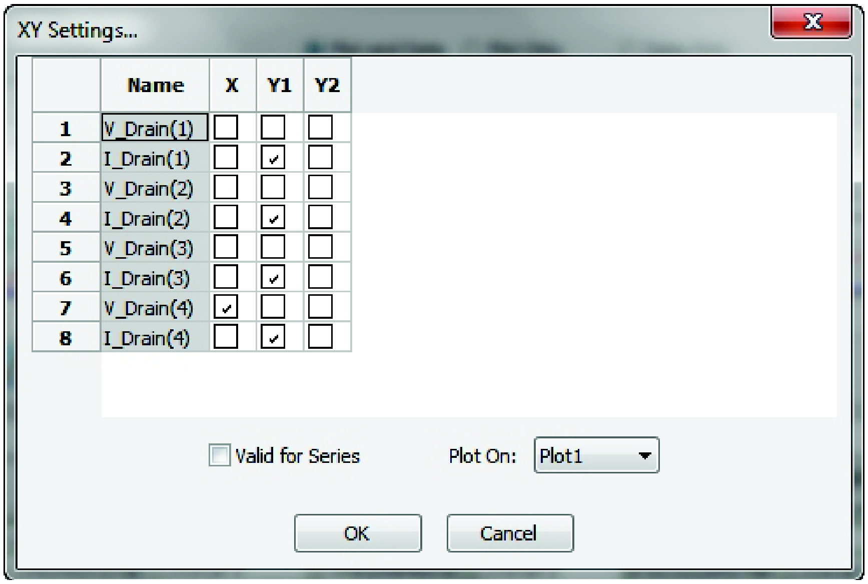
Figure 81: Result of data selection



	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 data has not been selected, click the Plot icon and the XY setting dialog box opens. Select the settings that you want to plot at one time (see next figure).

Figure 82: XY setting dialog



	Name	X	Y1	Y2
1	V_Drain(1)	□	□	□
2	I_Drain(1)	□	☑	□
3	V_Drain(2)	□	□	□
4	I_Drain(2)	□	☑	□
5	V_Drain(3)	□	□	□
6	I_Drain(3)	□	☑	□
7	V_Drain(4)	☑	□	□
8	I_Drain(4)	□	☑	□

□ Valid for Series

Plot On: Plot1

Plot the data

Click the **Plot** icon on the toolbar to display the Plotting menu.

After choosing the data that you want to plot, choose one of the following commands:

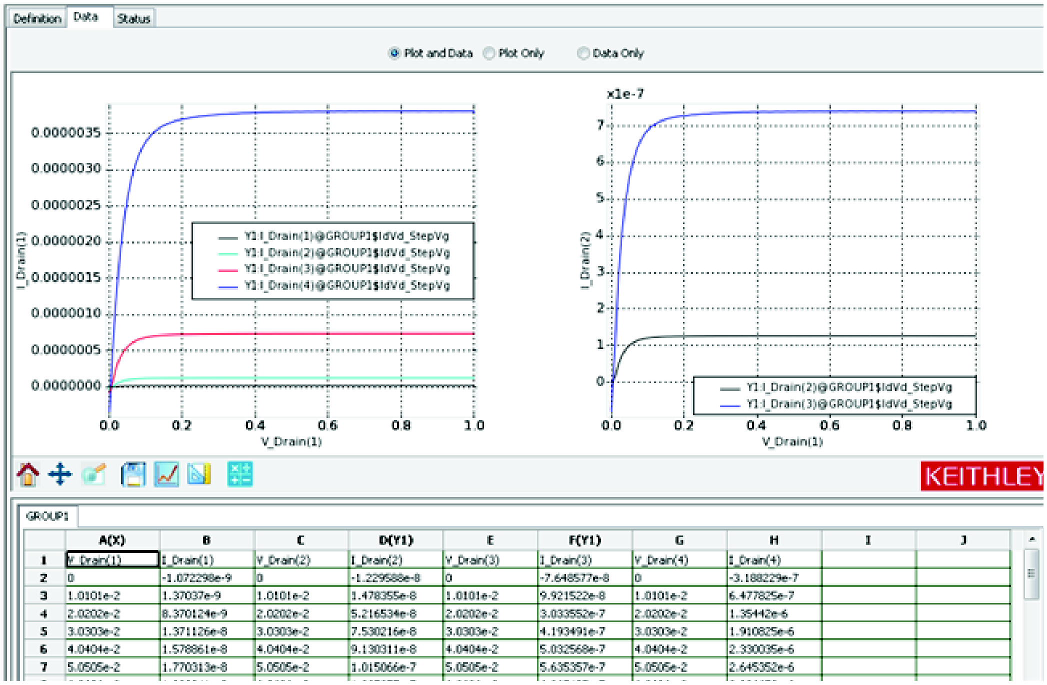
New Plot: Draws a graph on a new plot separate from the existing one.

Plot1: Draws a graph on plot1, which already exists. If plot1 is not blank, the original lines will remain instead of being deleted.

NOTE

If the test is run on multiple groups, the results of all the groups under test are shown on one plot (see next figure).

Figure 83: Plot shows two ids_vd plots



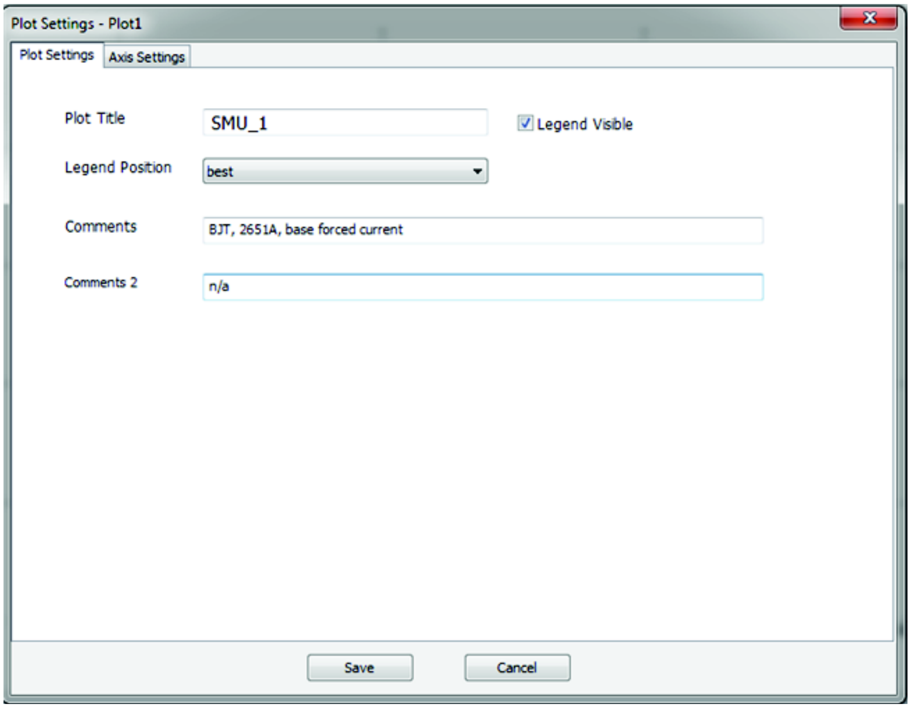
	A(X)	B	C	D(Y1)	E	F(Y1)	G	H	I	J
1	V_Drain(1)	I_Drain(1)	V_Drain(2)	I_Drain(2)	V_Drain(3)	I_Drain(3)	V_Drain(4)	I_Drain(4)		
2	0	-1.072298e-9	0	-1.229588e-8	0	-7.648577e-8	0	-3.188229e-7		
3	1.0101e-2	1.37037e-9	1.0101e-2	1.478355e-8	1.0101e-2	9.921522e-8	1.0101e-2	6.477825e-7		
4	2.0202e-2	8.370124e-9	2.0202e-2	5.216534e-8	2.0202e-2	3.033552e-7	2.0202e-2	1.35442e-6		
5	3.0303e-2	1.371126e-8	3.0303e-2	7.530216e-8	3.0303e-2	4.193491e-7	3.0303e-2	1.910825e-6		
6	4.0404e-2	1.578861e-8	4.0404e-2	9.130311e-8	4.0404e-2	5.032568e-7	4.0404e-2	2.330035e-6		
7	5.0505e-2	1.770313e-8	5.0505e-2	1.015066e-7	5.0505e-2	5.635357e-7	5.0505e-2	2.645352e-6		

Define properties of a graph and axis

Right-click on any portion of the graph to be edited, then choose the **Plot and Axis Settings** command in the Graph Settings menu. The Plot Setting dialog box opens and it contains two tabs: **Plot Setting** and **Axis Setting** (see next figures).

Plot setting tab

Figure 84: Plot Settings tab



Plot Title: SMU_1

☑ Legend Visible

Legend Position: best

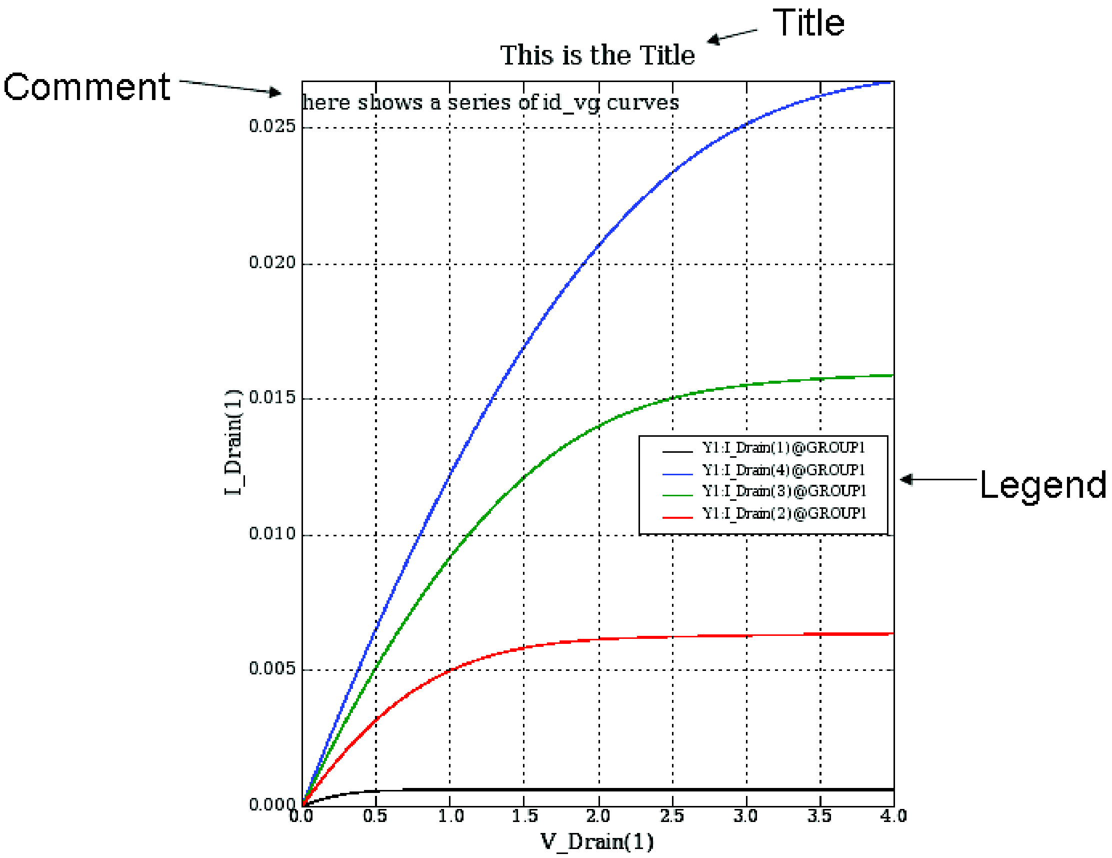
Comments: BJT, 2651A, base forced current

Comments 2: n/a

- **Plot Title**: sets the title of the selected plot.
- **Legend Visible**: causes the legend to appear on the plot when selected.
- **Legend Position**: selects the position for the legend on the plot. There are many positions available. However, the recommended position choice is "best."
- **Comments**: allows you to write comments which will be displayed on the selected plot.

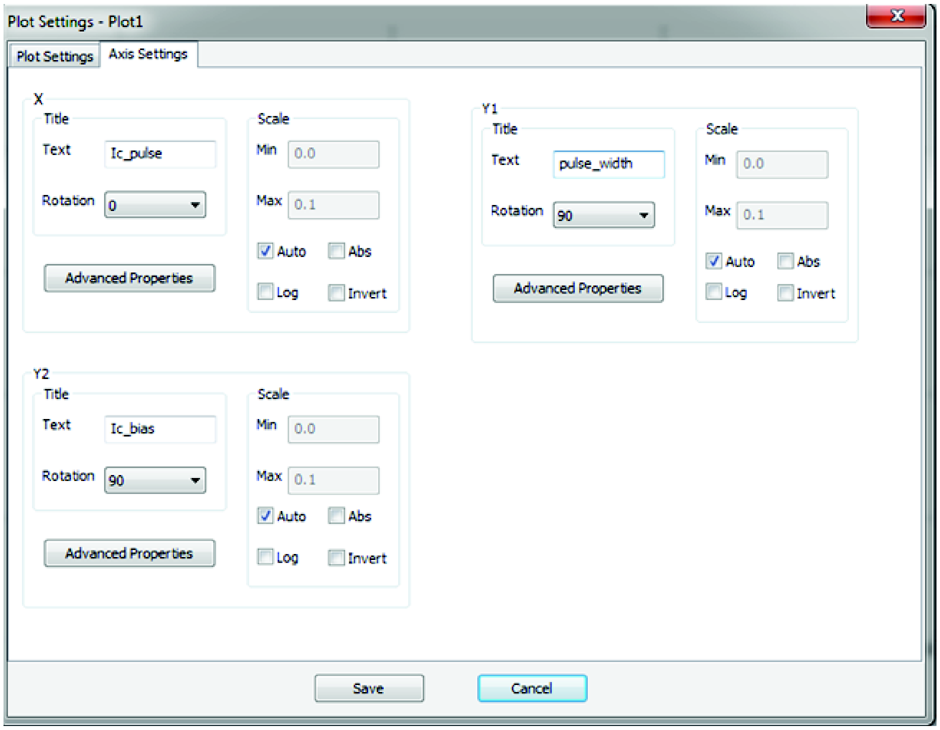
The next figure is an example of a plot with a title on top, legend on the right, and comments on the top left corner.

Figure 85: Results from the plot settings tab



Axis setting tab

Figure 86: Axis Settings tab



Plot Settings - Plot1

Plot Settings | Axis Settings

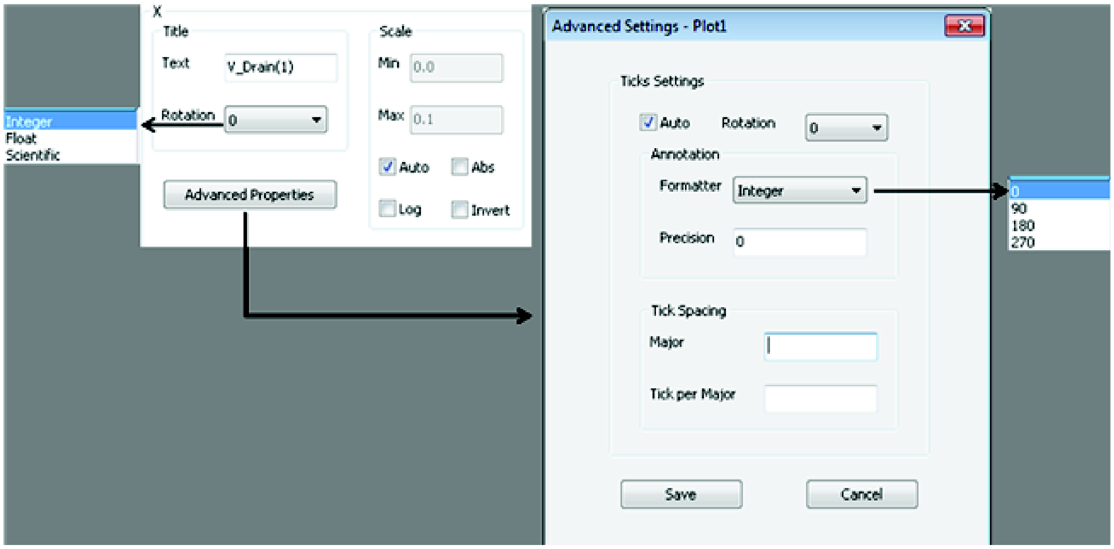
X
Title
Text: Ic_pulse
Rotation: 0
Scale
Min: 0.0
Max: 0.1
☑ Auto ☐ Abs
☐ Log ☐ Invert
Advanced Properties

Y1
Title
Text: pulse_width
Rotation: 90
Scale
Min: 0.0
Max: 0.1
☑ Auto ☐ Abs
☐ Log ☐ Invert
Advanced Properties

Y2
Title
Text: Ic_bias
Rotation: 90
Scale
Min: 0.0
Max: 0.1
☑ Auto ☐ Abs
☐ Log ☐ Invert
Advanced Properties

The axis setting tab contains three areas that correspond to the settings of the X, Y1, and Y2 axes. The items in these three areas are the same for each axis. Only one axis is described in the next figure and it shows all of the items and sub-items for one axis on the Axis Settings tab.

Figure 87: Items on the axis settings tab



X
Title
Text: V_Drain(1)
Rotation: 0
Scale
Min: 0.0
Max: 0.1
☑ Auto ☐ Abs
☐ Log ☐ Invert
Advanced Properties

Advanced Settings - Plot1
Ticks Settings
☑ Auto Rotation: 0
Annotation
Formatter: Integer
Precision: 0
Tick Spacing
Major:
Tick per Major:

Title section

Text: is 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 deselected.

Max: sets the maximum value of the axis. Valid when Auto is deselected.

Auto: will auto-scale the axis according to the data. Auto is the default setting.

> NOTE
>
> To specify exact numbers for the minimum and maximum values of the scale, deselect Auto and enter the desired 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 linear.

Invert: changes the minimum value of the axis to the maximum value and the maximum value to the minimum value.

Advanced Properties

Click the **Advanced Properties** button (see next figure) to open the dialog box. Items in the Advanced Setting dialog box 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.

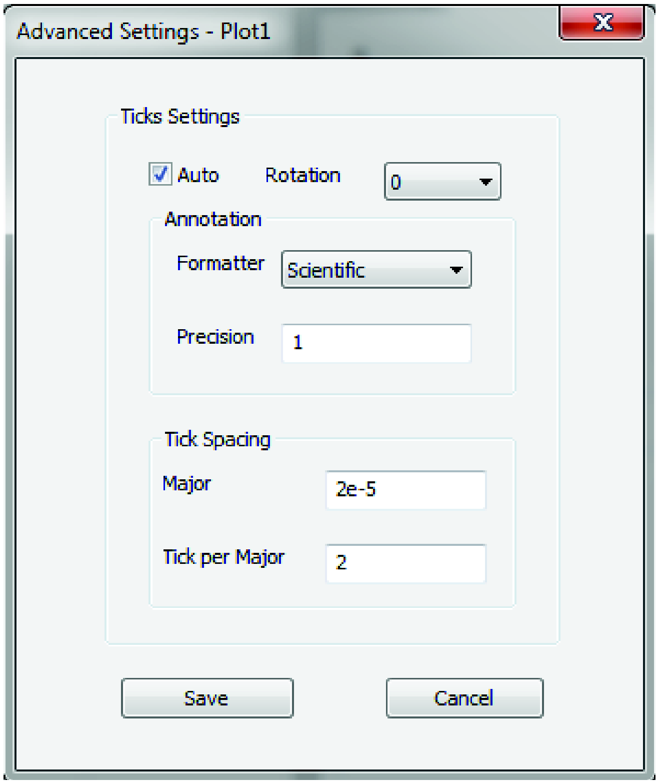
Formatter: sets the format for the annotation of the ticks. Three formats are provided: Integer, Float, and Scientific.

Precision: sets the precision of the annotation and is valid when Float or Scientific format is chosen.

Major: sets the value between major ticks along the axis.

Tick per Major: sets the number of sub-ticks inside a major tick.

Figure 88: Plot Settings Advanced Properties X axis



Advanced Settings - Plot1

Ticks Settings

☑ Auto Rotation 0

Annotation

Formatter Scientific

Precision 1

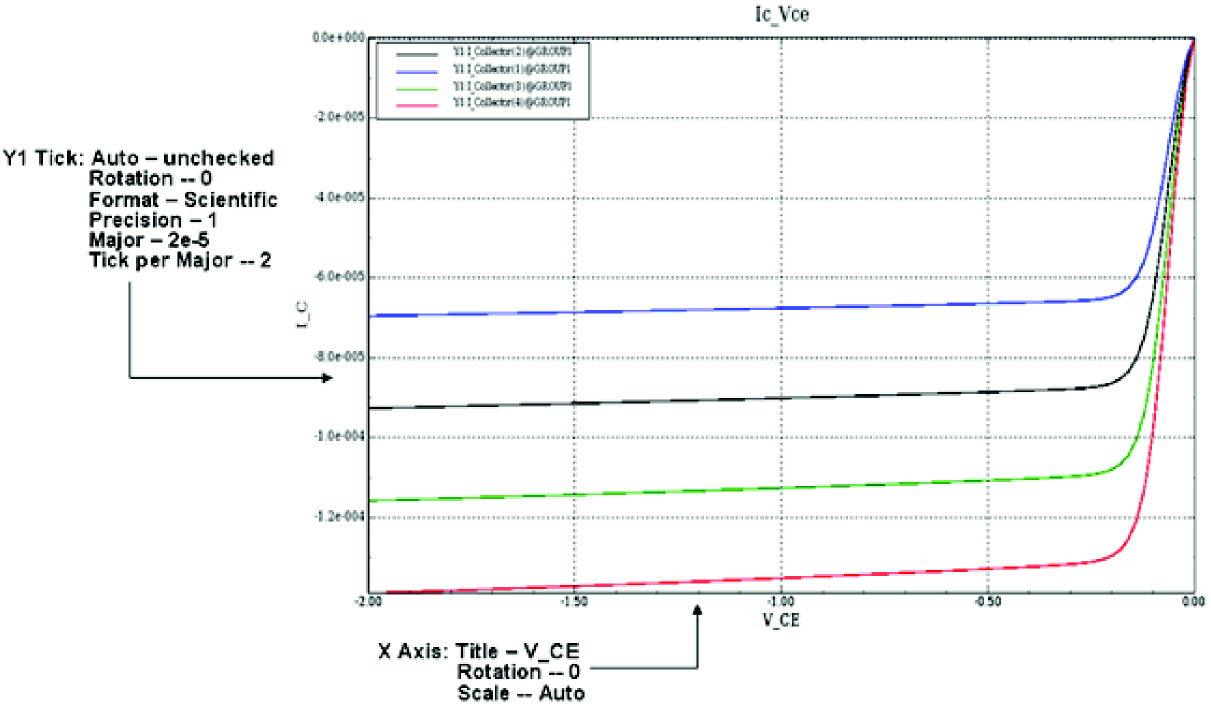
Tick Spacing

Major 2e-5

Tick per Major 2

The next figure shows a series of Ic_Vce curves with marked features.

Figure 89: Ic_Vce curves example



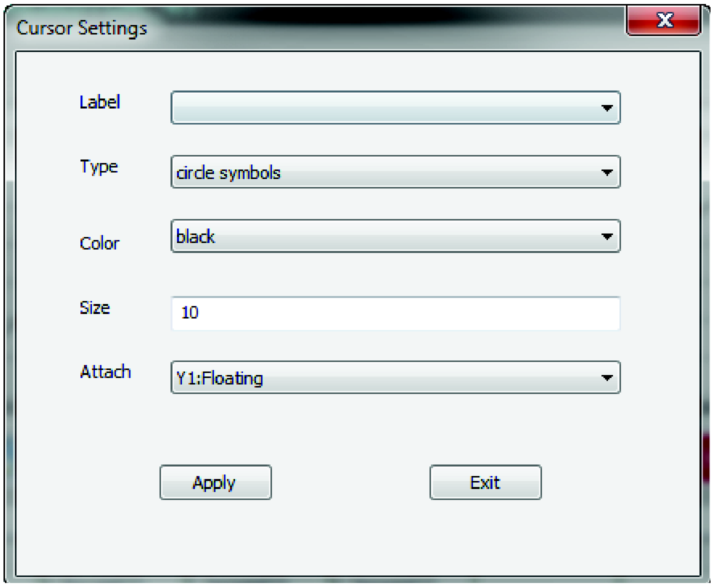
Y1 Tick: Auto – unchecked
Rotation -- 0
Format – Scientific
Precision – 1
Major – 2e-5
Tick per Major -- 2

X Axis: Title – V_CE
Rotation -- 0
Scale -- Auto

Define a cursor

Right-click on any portion of the plot and choose Cursor Settings from the Graph Settings menu. The Cursor Settings dialog box opens. The next figure shows the Cursor Settings dialog box and cursors on the I_V curve of a diode.

Figure 90: Cursor Settings dialog box



Label

Type: circle symbols

Color: black

Size: 10

Attach: Y1:Floating

Items in the Cursor Setting dialog box are as follows:

Label: selects an existing cursor name. Choose a cursor using the drop-down list.

Type: selects the shape of the chosen cursor.

Color: selects the color of the chosen cursor.

Size: changes the size of the chosen cursor

Attach: allows you to select between Floating or a specific curve. If set to Floating, the cursor can go anywhere on the plot, otherwise it can only run along the selected curve.

Apply: applies the settings for the chosen cursor.

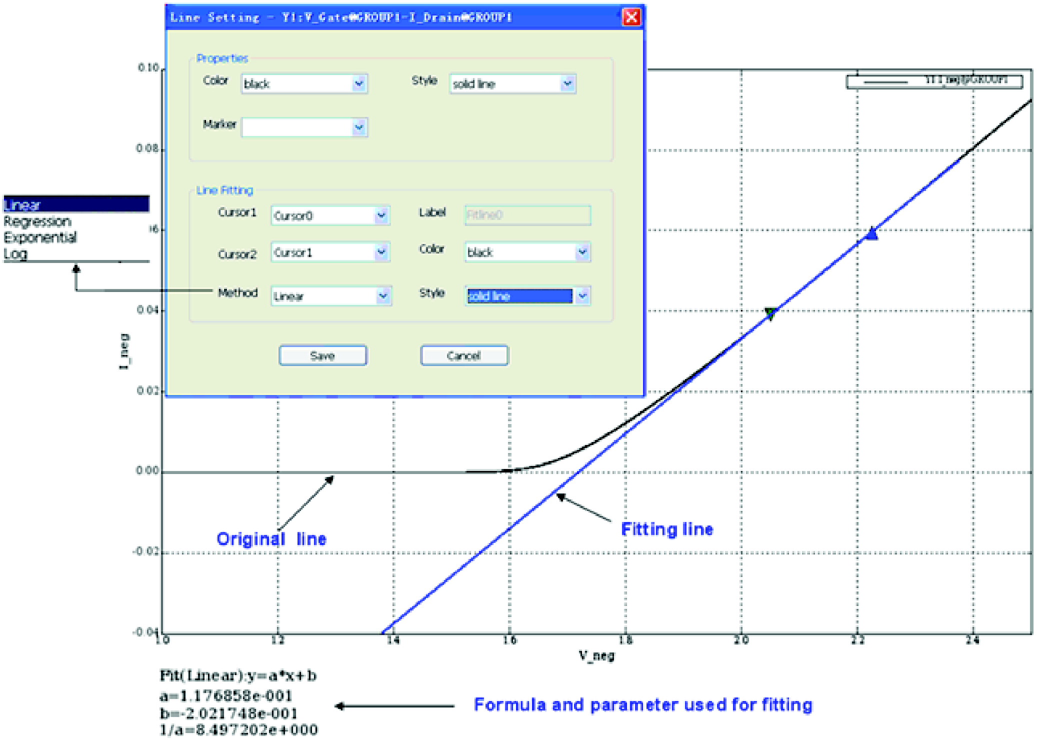
Exit: exits the dialog box. If Apply is not selected before you Exit, all of the settings are ignored.

After clicking **Apply**, the cursor displays on the plot. If you click the cursor, the coordinates (X, Y) of the current position display on the bottom plot (see previous figure).

Line properties and fitting

Click the line to be edited, then right-click, and select the **Line Properties and Fitting** command in the Graph Settings menu. The Line Settings dialog box opens (see next figure).

Figure 91: I_V curve of a diode with the linear fitting line and the formula used



Line Setting - Y1:V_Gate@GROUP1-I_Drain@GROUP1

Properties

Color: black

Style: solid line

Marker:

Line Fitting

Cursor1: Cursor0

Cursor2: Cursor1

Method: Linear

Label: Fitline0

Color: black

Style: solid line

Original line

Fitting line

```
Fit(Linear):y=a*x+b
a=1.176858e-001
b=-2.021748e-001
1/a=8.497202e+000
```

Formula and parameter used for fitting

The previous figure shows the original line (black) together with the fitting line (blue) without features. The formula used is displayed at the bottom left corner of the plot.

The Line Setting dialog box contains several items:

Properties

- **Color**: allows the user to choose the color of the selected line.
- **Style**: selects the style of the selected line: solid, dashed, dotted, or dash-dot.
- **Line Width**: sets the width of the selected line.
- **Marker**: selects the type marker for each data point on the line. There are no markers by default.
- **Marker Color**: selects the color of the markers.
- **Marker Size**: selects the size of the markers.

Line fitting

- **Cursor1**: selects the first cursor to define the line segment to be fitted (pick from the drop-down list).
- **Cursor2**: selects the second cursor to define the line segment to be fitted (pick from drop-down list).
- **Method**: selects the function used to fit the segment of the line (pick from Linear, Regression, Exponential, or Log).
- **Label**: shows the fit line label.
- **Color**: Selects the fit line color (choose from the drop-down list).
- **Style**: Selects the fitline style: solid, dashed, dotted, or dash-dot.

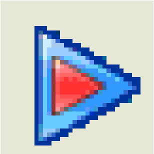
NOTE

An original line can only have one fit line. If the number of fit lines on the graph sheet is two or more, only one formula will display.

Append and clear append

The append icon is shown below (see next figure).

Figure 92: Append Run icon



The append function

The Run function has only one data worksheet associated with each specific test (refer to Data plot). This data worksheet contains the data from the last test and each new Run execution updates the data in the worksheet.

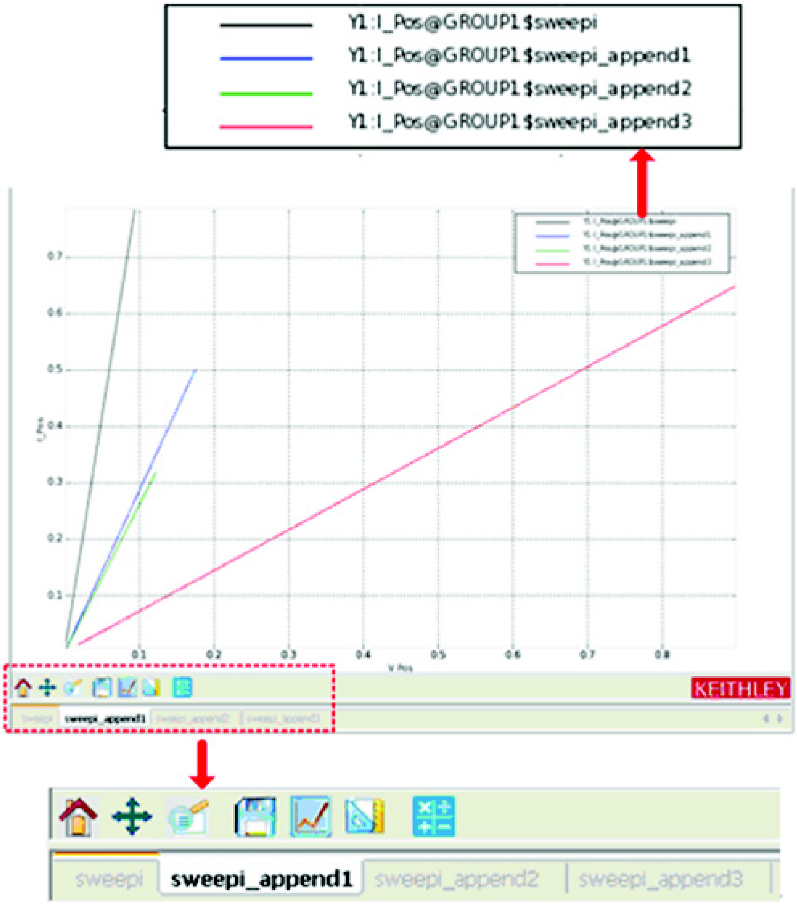
The Append function operates as Run does, but it creates a new worksheet for the Append test data and all previous data is kept (including previous Append data).

- Each Append run generates an additional Append worksheet (Append1, Append2 and so on) for each additional execution of the test
- Append is operational at the test and device level
- When the project is opened, ACS Basic will check for existing append data files for each test and load the data into each data sheet accordingly
- The Plot function will plot all the append data automatically. In the Data tab, the Append data curves for a test are added to the Run data curves. The next figure shows the results of four Appended tests.
- The Append data is added in the data sheet (in a new tab) next to the previous data.
- When you choose to output a .csv file, each set of appended data is output to a different .csv file. The naming convention is testname_append1.csv, testname_append2.csv, and so on where testname is the name of the test.

NOTE

The maximum number of times that you can append a test is 30. If you try to append more than 30 times, a warning dialog box opens.

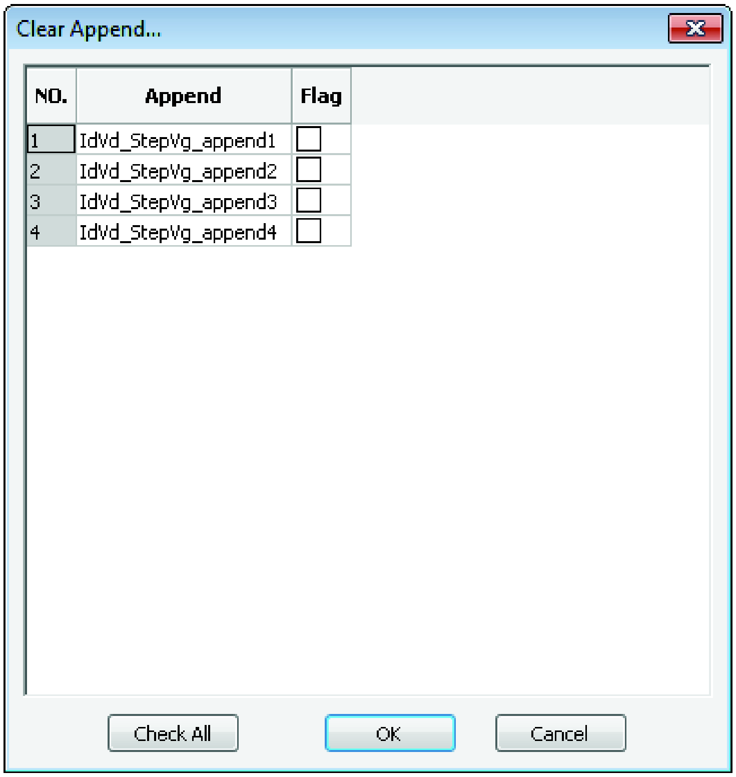
Figure 93: Appended data worksheet



Clear append data tabs

1. Click the Clear append icon from the toolbar and the dialog box opens.
2. Select the appending data tabs you want to clear (see next figure). Click OK. The selected data file(s) is deleted and the data sheet is updated.

Figure 94: Clear Append GUI



NO.	Append	Flag
1	IdVd_StepVg_append1	□
2	IdVd_StepVg_append2	□
3	IdVd_StepVg_append3	□
4	IdVd_StepVg_append4	□

Formulator

The Formulator is accessible in the Data tab. It allows you to perform data calculations on test data, as well as on the results of other Formulator calculations. The Formulator provides a variety of computational functions, common mathematical operators, and common constants.

A formula created by the Formulator is an equation composed from a series of functions, operators, constants, and arguments. For more detailed information about the Formulator functions, refer to the Formulator function reference.

Formulator arguments and constants

A formula created using the Formulator function performs calculations on any combination of the following:

- ITM, STM, or PTM test data
- Secondary data created by other Formulator formulas
- Standard constants for the list of constants

Some of the functions operate on columns of values (vectors) only. Others operate on single values (scalars) only. Still others operate on both single values (scalars) and columns of values (vectors).

The results of some calculations may be a column of values (vector) or a column containing only a single value.

Real-time functions, operators, and formulas

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 runs for each data point generated by the ITM just after the point is generated. The results of a real-time formula can 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 PTM data. A PTM Data sheet does not update until the test is completed.

The following operators and functions are real-time operators and functions:

1. Operators: +, -, *, /, ^ (exponential)
2. Functions: ABS, DELTA, DIFF, EXP, LN, LOG10, SQRT

Real-time formulas run as follows:

1. If a real-time formula is created before the ITM has been run, the formula runs automatically during each ITM run.
2. If a real-time formula is created after an ITM has been run, the formula runs initially upon adding it to the ITM and automatically during each subsequent ITM run.

Post-test only functions and formulas

A formula containing any one (or more) of the remaining Formulator functions is a post-test only formula. It will execute at the end of each run of the ITM, STM, or PTM where the formula is defined.

The results of a post-test formula can be viewed in the Data sheet or plotted 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, and VTSATGM.

The formula below is a post-test only formula, because MAX is a post-test only function:

```
RESULT2 = MAX(ABS(DELTA(I_drain)))
```

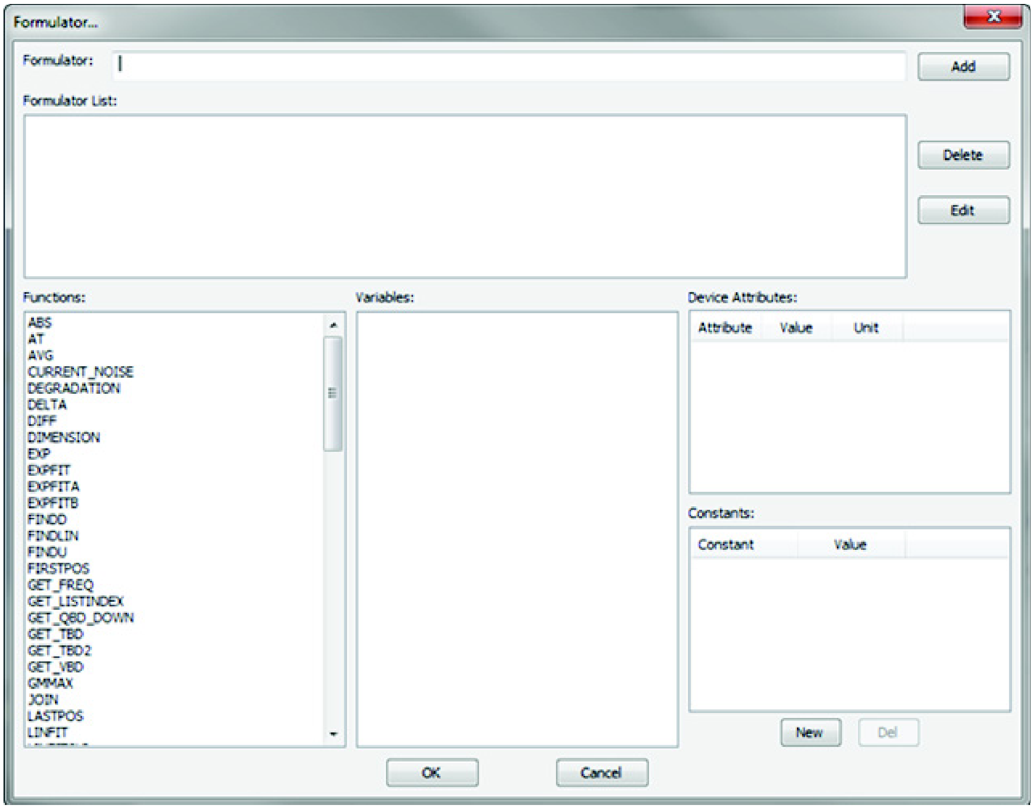
Post-test only formulas run as follows:

1. If a post-test only formula is created before the ITM, STM or PTM has been run; the formula performs automatically at the conclusion of each run.
2. If a post-test only formula is created after an ITM, STM, or PTM has been run, the formula performs initially upon adding it to the ITM, STM, or PTM and automatically at the conclusion of each subsequent run.

Start the formulator function

1. Open the data tab for the test that you want to analyze.
2. Click the **Formulator** function and the dialog box opens (see next figure).

Figure 95: Formulator dialog box



The Formulator dialog box

- **Formulator**: is used to create new formulas or edit existing formulas.
- **Formulator List**: displays previously created formulas for ITMs, STMs, or PTMs.
- **Add**: starts the calculation for the formula in the Formulator, and moves the formula to the Formulator List.
- **Delete**: will delete a formula when the formula is selected in the Formulator List.
- **Edit**: allows you to edit the formula that is currently selected in the Formulator List.
- **Functions**: displays a list of functions. When you select a function, it is added to the equation and displayed in the Formulator.
- **Variables**: lists the names of all columns in the Data sheet. Columns created by some functions may contain only a single value.

NOTE

Do not use X, Y1, and Y2 as variables in ACS Basic. These are reserved variables.

- **Constants**: is where you can insert each constant or value by name. When you click the constant in the constants list, it's added to the equation in the Formulator.
- **New**: opens a dialog input box that allows you to enter the name and value of the new constant. After you enter the name and value, click OK; the new constant is added to the constants list.
- **Del**: deletes the selected constant

The formulator functions

The Formulator functions, as well as operators and constants, can be used individually or in various combinations to create simple or complex analysis equations.

Multiple functions can be nested. For example, in one equation you can calculate the following:

1. Find the maximum value of a column, using the MAX function.
2. Find the absolute value of the maximum, using the ABS function.
3. Multiply the ABS value by a constant.

The equation below illustrates the use of nested Formulator functions:

```
MAXABS = 10*ABS(MAX(ColumnA))
```

The number of levels of nesting is unlimited. The purpose, format, and arguments for these functions and all other functions available in the Formulator are described in the Formulator arguments and constants topic.

Advanced operations

Projects

This section describes how to use an existing project in the standard ACS Basic library, and how to build a new project.

⚠ CAUTION

The software connections must accurately reflect the physical hardware connections at the time the test is run. Incorrect configurations result in test errors, and possibly device damage. At every test startup, you must check that the connections in the device view of the ACS Basic interface match the physical connections between the test instruments and the device.

An existing project

Once started, ACS Basic will automatically open the previously opened project by default.

To view the default project:

1. Click the **Preference** item in the Tools menu; the Preferences dialog box opens.
2. On the Path tab, you will see the Auto-Load Project and the Default Project.

To open a project other than the default:

1. Click the **Open** icon 📂 on the toolbar of the ACS Basic GUI.
2. Select the project file you would like to open.
3. Select and open the .xml file project (for example, trace_mode_lab.xml).

NOTE

If SMUs are required in the project you are trying to open, the ACS Basic software will try to detect them in the current configuration. Once the SMUs are found, the project will load the project in the configuration navigator.

If the SMU connections for this project do not match the current physical connection, the SMUs Missing dialog box opens.

The SMUs Missing dialog box informs you that some of the SMUs for a project cannot be found. To correct this, check all module settings in the project and reassign them where necessary until the SMUs Missing GUI no longer displays when opening an existing project.

Create a new project

To create a new test project, refer to the Build a project plan topic in this section.

Create a new device

Each device has a specific device configuration. The Device Settings GUI is the interface used to set the connections of the device.

For Single-test mode, in the Definition tab of the Test GUI, you can click the Device View icon to view the Device Settings GUI of the currently open test module. For Multi-test mode, click the device node on the configuration navigator, and the Device Setting GUI opens (see next figure).

There are two tabs in the device workspace:

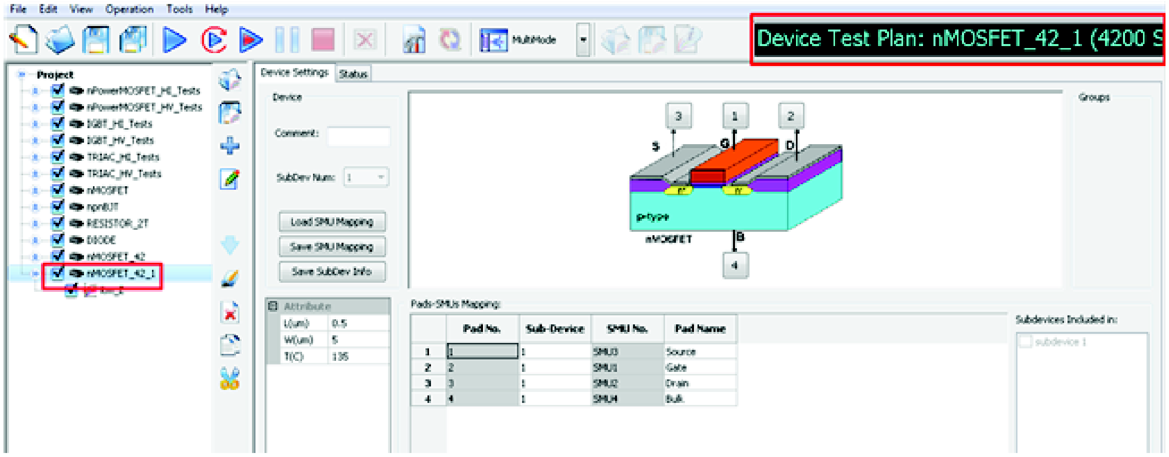
1. **Device Settings**: is the interface where you can assign the device settings according to the physical connection of the device. This tab opens by default
2. **Status**: shows the device information.

Add a new device

To add a new device to the configuration navigator in Multi-test mode:

1. Click the **Add new test** icon ✚ on the vertical toolbar; the Specify the Device/Test Type dialog box opens.
2. Select the **Device** option and click **OK**. A SMU Type dialog box will open.
3. Select the Model 42xx or 26xx SMU to add a new device to your test. Click **OK**; the Select Device Type dialog box opens.
4. Select a device, and then click **OK**. The Image Browser dialog box opens (see next figure).
5. Select a .bmp file, and click **Open**.
6. The new device is now created and added to the configuration navigator. In this example, the device name is nMosfet_42 and its Device Settings tab opens by default (see next figure). Click this new device to view its information.

Figure 96: Example 4200 SMU information



Device Test Plan: nMOSFET_42_1 (4200 S

Comment:

SubDev Num: 1

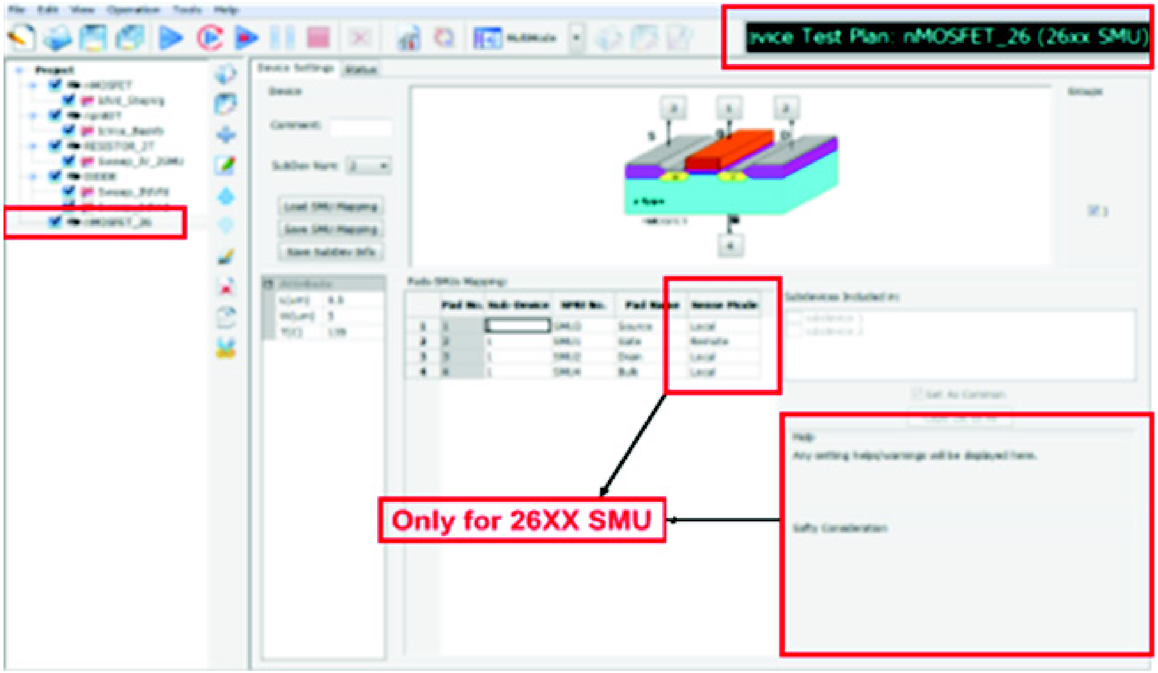
Attribute	
L(um)	0.5
W(um)	5
T(C)	135

Pads-SMUs Mapping:

	Pad No.	Sub-Device	SMU No.	Pad Name
1	1	1	SMU3	Source
2	2	1	SMU1	Gate
3	3	1	SMU2	Drain
4	4	1	SMU4	Bulk

7. If you are adding a Model 26xx SMU, you will also need to determine the Sense Mode of the device. For example, if your device is used locally, you will chose Local (local is for 2-wire) and if it is used remotely, you will choose Remote (remote is for 4-wire)(see next figure). The new 26xx SMU device in the next figure is named nMOSFET_26.

Figure 97: Example 26xx SMU information



vice Test Plan: nMOSFET_26 (26xx SMU)

Pads SMUs Mapping:

	Pad No.	Sub-Device	SMU No.	Pad Name	Sense Mode
1	1		SMU0	Source	Local
2	2	1	SMU1	Gate	Remote
3	3	1	SMU2	Drain	Local
4	4	1	SMU4	Bulk	Local

Help

Any setting help/warnings will be displayed here.

Safty Consideration

Only for 26XX SMU

Sense mode: You can select either Local or Remote for each 2 wire and 4 wire SMU.

- Local: 2 wire sensing
- Remote: 4 wire sensing

NOTE

If a Model 2651A is assigned and set to Remote sense mode on the drain or collector (IGBT), other Model 26xx SMUs should be forced to Remote sense mode.

The Device Settings tab for the Model 26xx SMU contains a Help area. This area will help you to determine if you have configured your SMU in a way that will not function properly. For instance, here are some messages that you may encounter:

- "2600 non-A/B SMUs are involved, Pulse mode test cannot be supported" - this means if non-A/B Model 26xx is part of your configuration, then pulse mode cannot be applied in the ITM device.
- "2657A SMU involved, the protection module 2657A-PM-200 should be connected to low voltage SMUs" - this means if the Model 2657A is part of your configuration, you need to ensure that any low voltage SMUs are protected by the protection module.
- "CRITICAL The 2651A and 2657A cannot be connected at the same time" - this means that the Model 2657A has high-voltage and should not be connected to a device that also has a Model 2651A connected to the same device since it will not be protected from the Model 2657A high-voltage.
- "CRITICAL Mixed use of Model 2601(A/B) and 2602 (A/B) with Model 2657A are not supported. The Model 2657A-PM-200 Protection Module only protects up to 200V" - this means that the Model 2657A has high-voltage and should not be connected to a system that also contains Models 2601(A/B) and 2602(A/B).
- "Cannot place a low current SMU in the path of high current" - this means that you should not combine the use of a high current SMU (2651A) and a low current SMU on the Drain-Source of the MOSFET, Collector-Emitter of the BJT, Collector-Emitter of the IGBT, or on a two-terminal resistor.

NOTE

It is not a good practice to mix different SMU model numbers on a device. ACS Basic will provide you a warning if you try to mix SMUs inadvertently.

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 GUI. Here are some items to consider when configuring your hardware:

Non-A 26xx SMUs cannot be mixed with Model 265xA instruments.

A Model 2657A SMU cannot be mixed with Models 2651A, 2601B, or 2602B.

A Model 2651A SMU cannot be mixed with low-current SMUs.

For two-terminal devices, a Model 2651A cannot be mixed low-current SMUs.

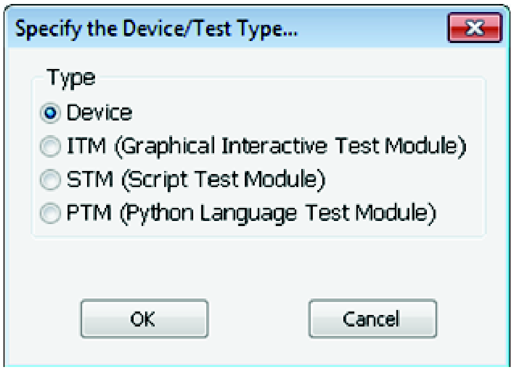
For three-terminal devices, if a Model 2651A SMU is connected to the collector, then the emitter must be connected to ground or another Model 2651A SMU. If a Model 2651A SMU is connected to the base, then the emitter and collector must be connected to ground or another Model 2651A SMU emitter and collector.

For four-terminal devices, if a Model 2651A SMU is connected to the drain, and a low-current SMU is connected to the gate, the low-current SMU cannot be connected to the source.

Add a blank test

1. Select the device you want to create the test under, then click the **Add new test** icon on the vertical toolbar; the Specify the Device/Test Type dialog box opens (see next figure).
   a. Select the desired test module type: a new ITM, STM, or PTM.
   b. Click **OK**. A test module is inserted to the configuration navigator.

Figure 98: Specify the Test Type dialog box



Specify the Device/Test Type...

Type

● Device

○ ITM (Graphical Interactive Test Module)

○ STM (Script Test Module)

○ PTM (Python Language Test Module)

2. If desired, change the name of the new test in the configuration navigator.
3. Select which Groups the test will include in the Groups area in the ITM Definition tab or on the STM Setup tab.
4. Save the test project by clicking the **Save** icon located on the toolbar. You can also use the Save function in the File menu.

Setting Preferences

Select Preferences in the Tools menu so that you can customize your settings. The Preferences dialog box opens.

There are two tabs in the Preferences setting dialog box: Path tab and the Misc tab. The Path tab opens by default.

Path tab

Items on the Path tab (see next figure) include:

Project Repository: shows the default directory path where all projects are saved. You can also click the **Browse** function and locate another directory which will change the default path.

Auto-Load Project: shows the method ACS Basic uses to decide which project will load on start-up.

Default Project: shows the default project path.

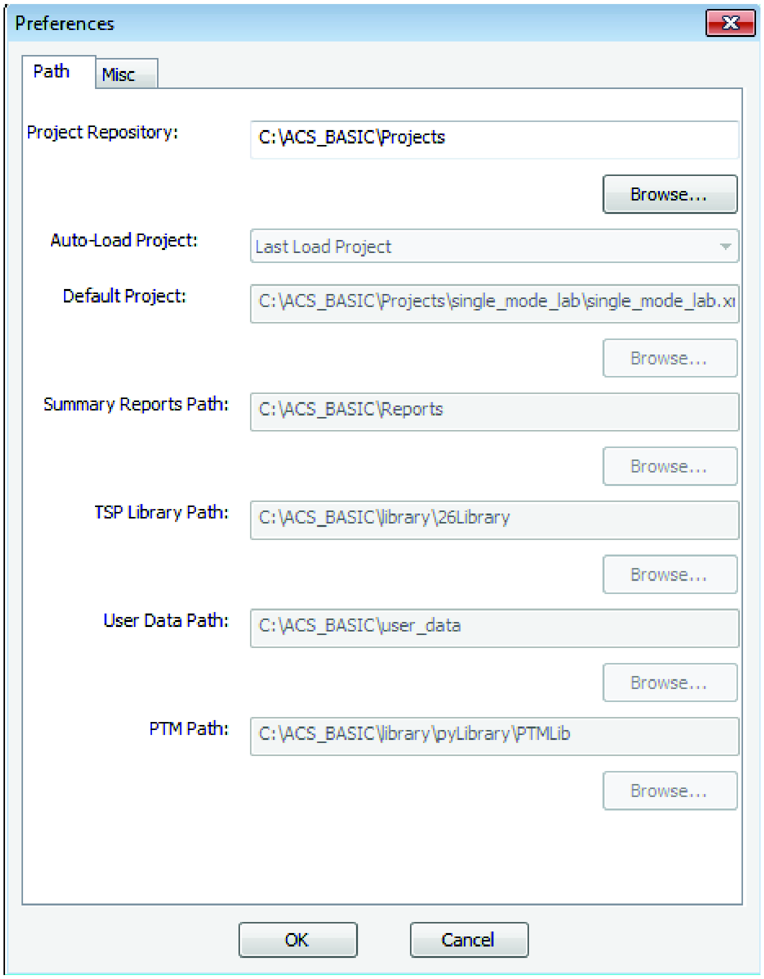
Summary Reports Path: shows the default directory path where test reports are located.

TSP Library Path: shows the default TSP library directory path used to store the STM library.

User Data Path: shows the default path where binning files and data files are located.

PTM Library Path: shows the default PTM library directory path that is used to store the python library.

Figure 99: Preferences Path tab



Project Repository: C:\ACS_BASIC\Projects

Auto-Load Project: Last Load Project

Default Project: C:\ACS_BASIC\Projects\single_mode_lab\single_mode_lab.x

Summary Reports Path: C:\ACS_BASIC\Reports

TSP Library Path: C:\ACS_BASIC\library\26Library

User Data Path: C:\ACS_BASIC\user_data

PTM Path: C:\ACS_BASIC\library\pyLibrary\PTMLib

Misc tab

Items that are active on the Misc tab (see next figure) include:

26XX Settings

- **SMU Remote Sense**: allows you to select which SMU will have remote sense for a special test, such as the four-wire resistor test.
- **Check all**: selects all the SMU Remote Sense boxes. It is deselected by default.
- **Auto Delay**: adds an auto delay to each test. It is selected by default.
- **Initialize Unused SMU**: indicates the status setting for unused SMUs in an ITM. There are three states: Default; High Impedance; Force Current 0A. Default is selected by default.
- **Clear old 26XX scripts when a project is opened**: is deselected by default.

Output Enable Action

- **OE Output Off**: To disable the Series 2600B, Models 2601B, 2602B, and the Series 2400 SMUs output when the interlock is disengaged, select the OE Output Off option.

4200 Setting

- **IP Address**: 4200 IP address setting for an ITM test. This setting is only applicable if a Model 4200 has been set up.

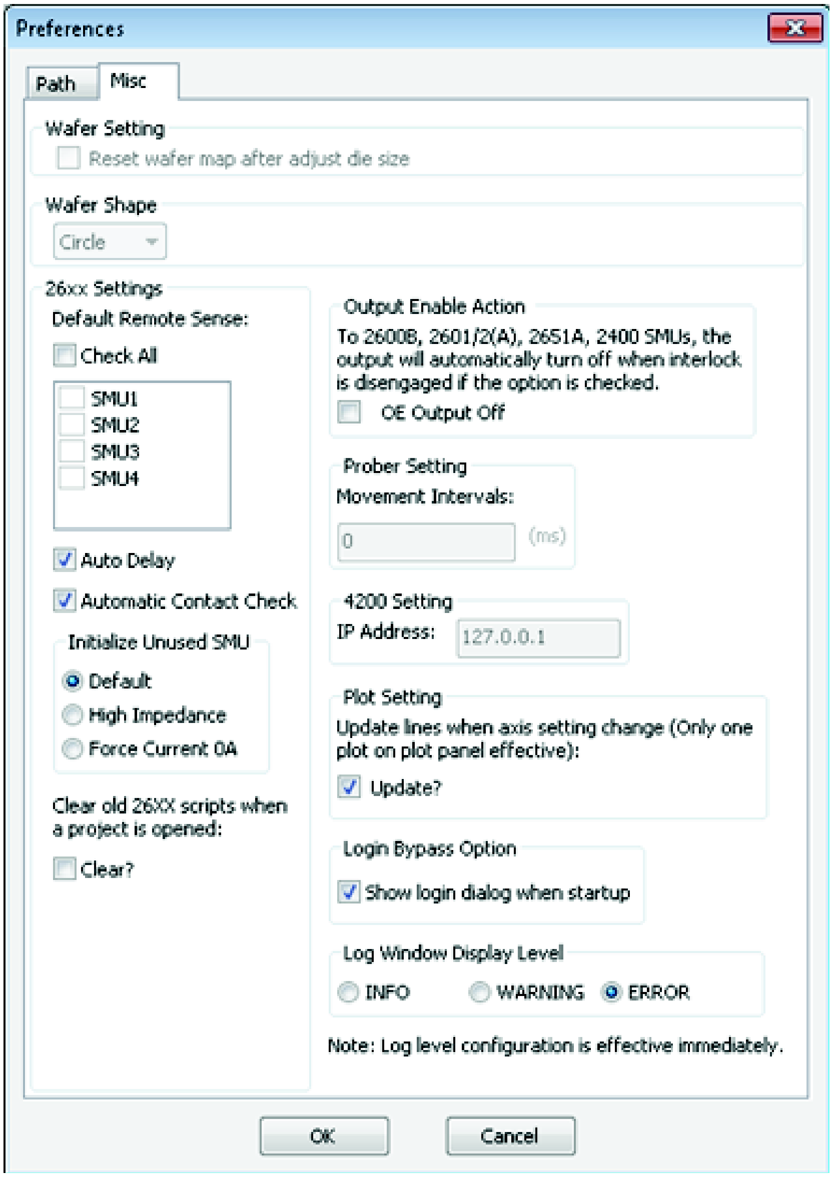
Login Bypass Option

- **Show login dialog when start up**: Selected by default.

Log Window Display Level

- **INFO**: The message area displays each command, including the WARNING and ERROR level.
- **WARNING**: The message area displays error messages as they occur, including the ERROR level.
- **ERROR**: The message area displays error messages when they occur.

Figure 100: Preferences Misc tab



Preferences

Path | Misc

Wafer Setting

□ Reset wafer map after adjust die size

Wafer Shape

Circle

26xx Settings

Default Remote Sense:

□ Check All

□ SMU1
□ SMU2
□ SMU3
□ SMU4

☑ Auto Delay

☑ Automatic Contact Check

Initialize Unused SMU

● Default
○ High Impedance
○ Force Current 0A

Clear old 26XX scripts when a project is opened:

□ Clear?

Output Enable Action

To 2600B, 2601/2(A), 2651A, 2400 SMUs, the output will automatically turn off when interlock is disengaged if the option is checked.

□ OE Output Off

Prober Setting

Movement Intervals:

0 (ms)

4200 Setting

IP Address: 127.0.0.1

Plot Setting

Update lines when axis setting change (Only one plot on plot panel effective):

☑ Update?

Login Bypass Option

☑ Show login dialog when startup

Log Window Display Level

○ INFO ○ WARNING ● ERROR

Note: Log level configuration is effective immediately.

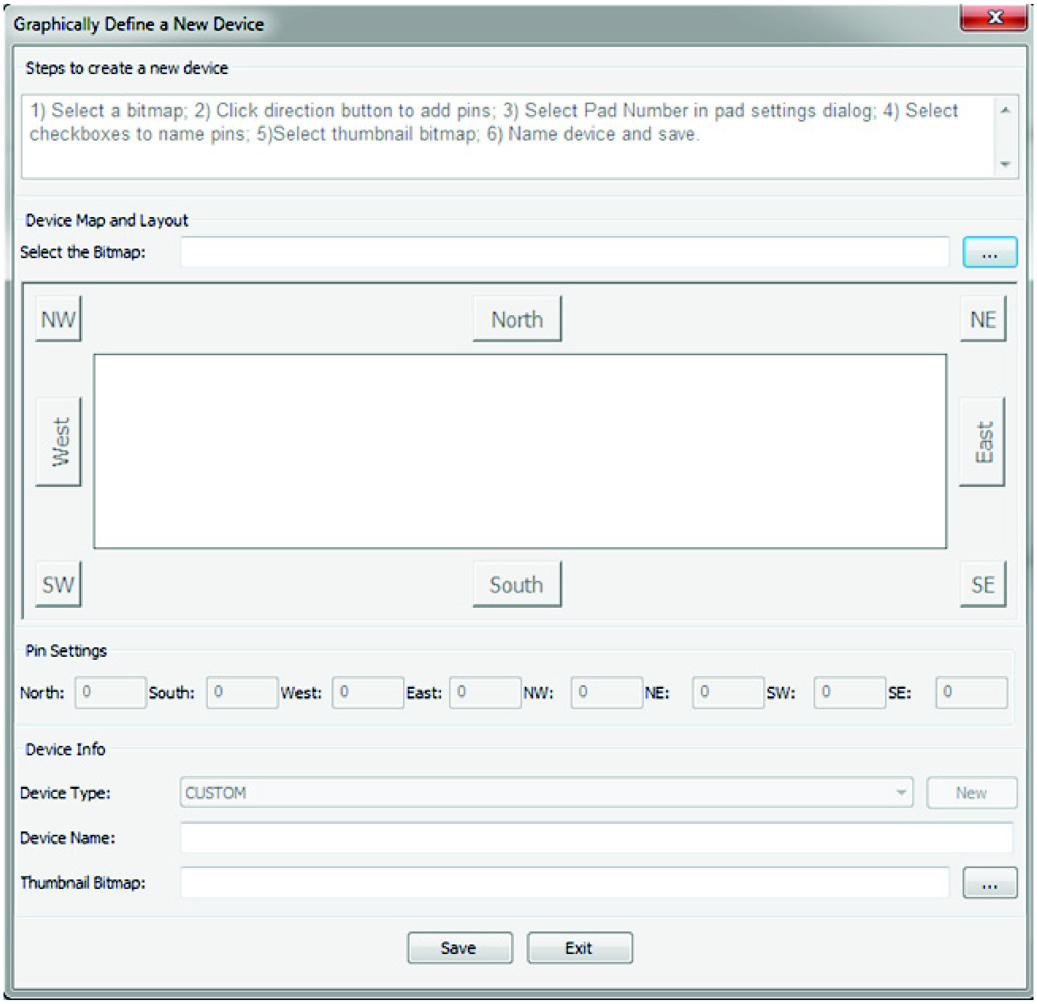
OK | Cancel

Graphically define a new device

There are two ways to open the Graphically Define a New Device dialog box.

1. In the Tools menu, click Graphically Define a New Device. The following is the GUI that opens (see next figure).

Figure 101: A blank Graphically Define a New Device



Graphically Define a New Device

Steps to create a new device

1) Select a bitmap; 2) Click direction button to add pins; 3) Select Pad Number in pad settings dialog; 4) Select checkboxes to name pins; 5)Select thumbnail bitmap; 6) Name device and save.

Device Map and Layout

Select the Bitmap:

Pin Settings

North: 0 South: 0 West: 0 East: 0 NW: 0 NE: 0 SW: 0 SE: 0

Device Info

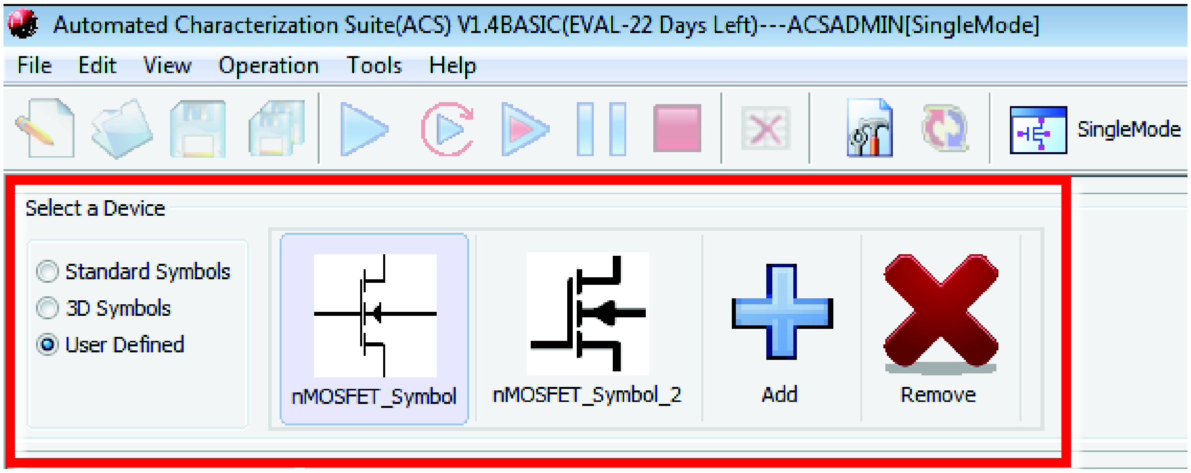
Device Type: CUSTOM

Device Name:

Thumbnail Bitmap:

2. Or, you can select the User Defined option in the Select a Device area of the select test module GUI (see next figure); then click the **Add** button.

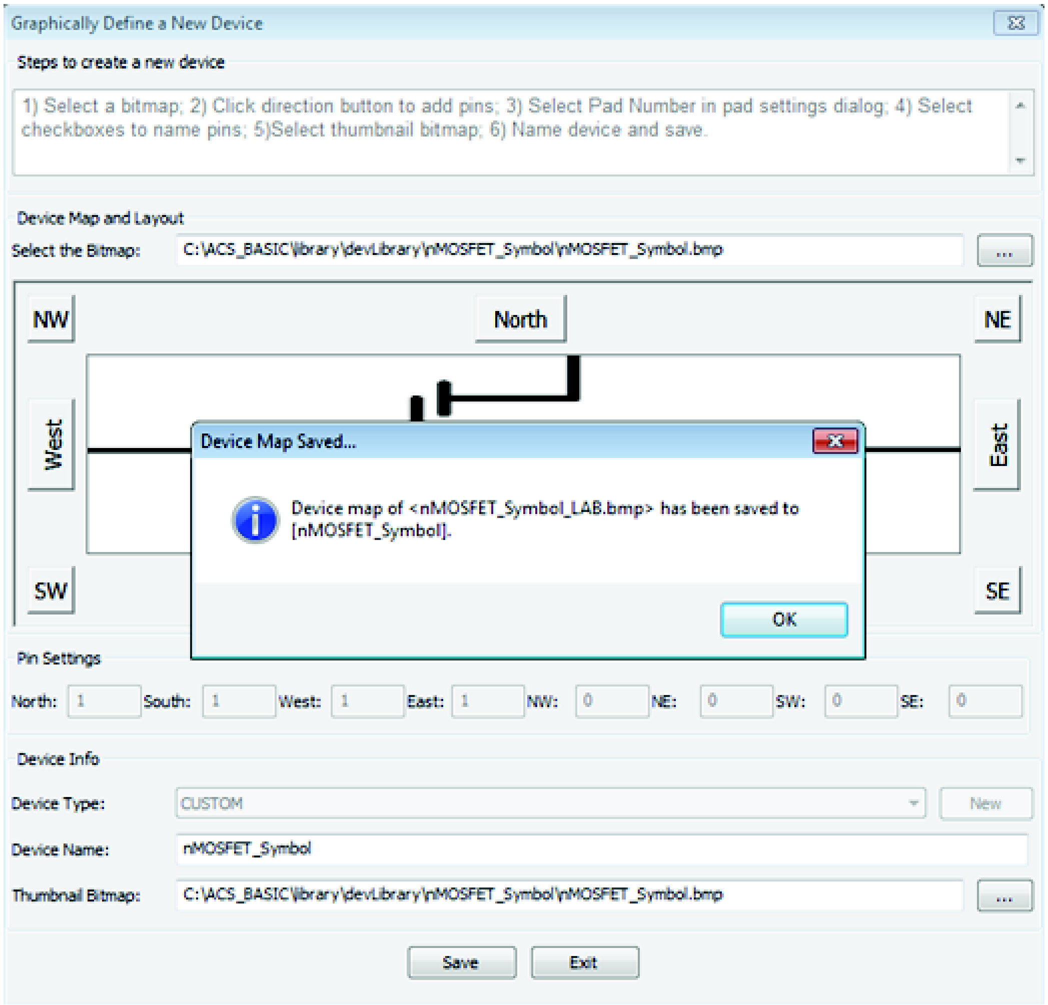
Figure 102: Add a User Defined device



3. Select a bitmap to represent the device by entering the file address or clicking the ellipsis function next to the Select the Bitmap area. The selected bitmap will load in the Device Editor GUI. The bitmap's name will appear in the Device Name edit box as the default name of this device. You can change this name by entering a new name. This device map is used as a graphic representation of the device workspace, plan, and module GUI.
4. Select the thumbnail bitmap of the device by entering the file address or clicking the ellipsis function next to the Thumbnail Bitmap area. This thumbnail is used as a graphic representation in the device selection area.
5. Enter the Pin Settings. The locations on the map are defined as North, South, West, East, NE (Northeast), SE (Southeast), NW (Northwest), and SW (Southwest).
    a. Select the location and a dialog box opens for setting the number of pads from a drop-down list.
    b. After you select the number of pads, select the check boxes under the number of pads. This will allow you to set the pad name for each pad in this location.
    c. After all of the positions are set, click **Save**.

The next figure shows the device nMOSFET_Symbol is saved and ready to use in ACS Basic.

Figure 103: Save the User Defined device



Graphically Define a New Device

Steps to create a new device

1) Select a bitmap; 2) Click direction button to add pins; 3) Select Pad Number in pad settings dialog; 4) Select checkboxes to name pins; 5)Select thumbnail bitmap; 6) Name device and save.

Device Map and Layout

Select the Bitmap: C:\ACS_BASIC\library\devLibrary\nMOSFET_Symbol\nMOSFET_Symbol.bmp

Device Map Saved...

Device map of <nMOSFET_Symbol_LAB.bmp> has been saved to [nMOSFET_Symbol].

Pin Settings

North: 1 South: 1 West: 1 East: 1 NW: 0 NE: 0 SW: 0 SE: 0

Device Info

Device Type: CUSTOM

Device Name: nMOSFET_Symbol

Thumbnail Bitmap: C:\ACS_BASIC\library\devLibrary\nMOSFET_Symbol\nMOSFET_Symbol.bmp

After you have defined and saved the device, it displays in the GUI. Now you can build and configure the test library for this device.

Section 6

ACS Basic test modules

In this section:

Configure a graphical interactive test module (ITM)

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

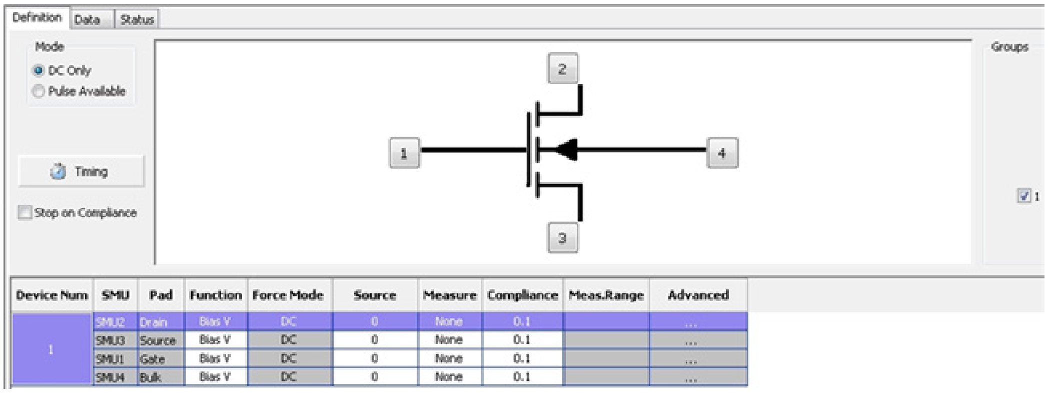
After inserting an ITM into the configuration navigator, it must be configured to meet your test requirements.

ITM

This ITM is based on the Series 2600B trigger model, and it supports both DC and pulse measurement. This ITM also supports high-power SMUs, such as models 265xA in a safe configuration.

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 (see next figure).

Figure 104: Definition tab of the Model 26XXB SMU ITM



Mode
● DC Only
○ Pulse Available

□ Stop on Compliance

Groups
☑ 1

Device Num	SMU	Pad	Function	Force Mode	Source	Measure	Compliance	Meas.Range	Advanced
1	SMU2	Drain	Bias V	DC	0	None	0.1		...
	SMU3	Source	Bias V	DC	0	None	0.1		...
	SMU1	Gate	Bias V	DC	0	None	0.1		...
	SMU4	Bulk	Bias V	DC	0	None	0.1		...

Mode:

- DC Only: All SMUs will only operate in DC mode.
- Pulse Available: Enables SMUs to operate in 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 auto ranging, will be disabled when Pulse Available is selected.

Timing: This button opens the ITM timing dialog box where you can set detailed timing settings for pulse and synchronization between SMUs (refer to the Timing dialog box for more information).

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.

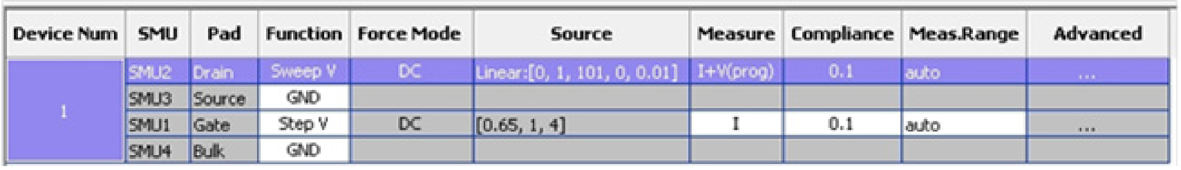
Groups: Select the group or groups where you want to run the test. The group is determined by the GPIB or Ethernet address. Make sure that all the groups you select have the same hardware configurations as your ITM.

SMU parameters: the forcing and measuring settings table is where you can set your parameters for testing.

Model 26XXB SMU parameters

The following figure displays the items that you can use to set up your testing for forcing and measuring functions for each SMU.

Figure 105: 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; based on the hardware connection.

SMU: SMU number defined in the Device work area; based on the hardware connection.

Pad: Pad name defined in the Device work area; based on the hardware connection.

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: Measures the current unit in amps and the voltage unit in volts. All options are:

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

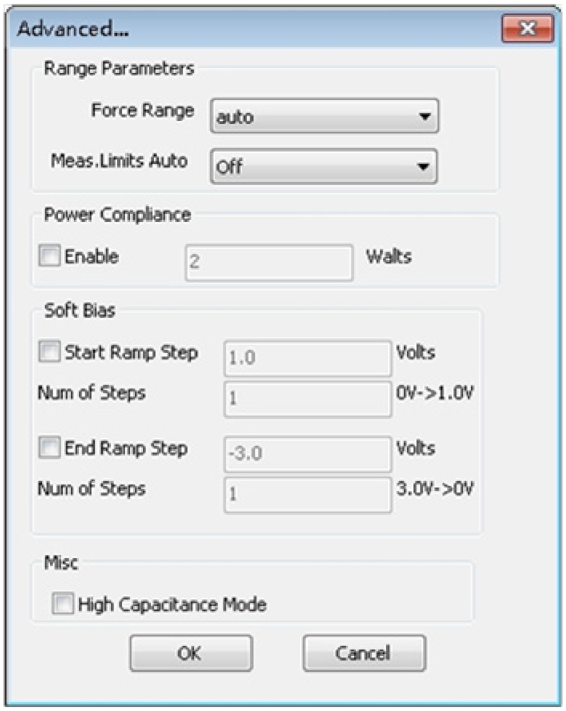
Compliance: Specifies any valid SMU current or voltage compliance limit. The current compliance is present for voltage forcing functions and the voltage compliance is used for current forcing functions.

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 I is selected in the Measure cell, ACS Basic 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.
- Numerical range options allow you to manually select a fixed current measurement range to suit your needs.

Advanced: Double click the cell will pop up a window for advanced settings for this SMU.

Figure 106: The SMU Advanced settings dialog box



Advanced...

Range Parameters

Force Range: auto

Meas.Limits Auto: Off

Power Compliance

□ Enable 2 Watts

Soft Bias

□ Start Ramp Step 1.0 Volts

Num of Steps 1 0V->1.0V

□ End Ramp Step -3.0 Volts

Num of Steps 1 3.0V->0V

Misc

□ High Capacitance Mode

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 Meas. Range is set to Auto, this option is enabled. This parameter is used with autoranging to put a lower bound on the range used. Since lower ranges generally require greater settling times, setting a lowest range limit might make measurements require less settling time.

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

Soft Bias: Apply Soft Ramp at the beginning of the Bias or first Sweep/List point and at the end of the Bias or last Sweep/List point. Typical use cases are for high power MOSFET measurements. For more information about Soft Ramp and how to use it, refer to the SMU forcing functions topic or the Configure the general purpose PTM topic.

Misc:

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

SMU forcing functions

The SMU forcing functions are selected in the SMU forcing and measuring settings table. The functions are: Bias V, Bias I, Sweep V, Sweep I, List V, List I, Step V, Step I, V Meter, GND, Open; in DC Only mode there are two more functions: Sweep2 V and Sweep2 I. Due to the selected forcing functions, the forcing value will be defined in the Source cell in the SMU forcing and measuring settings table.

Bias V, Bias I

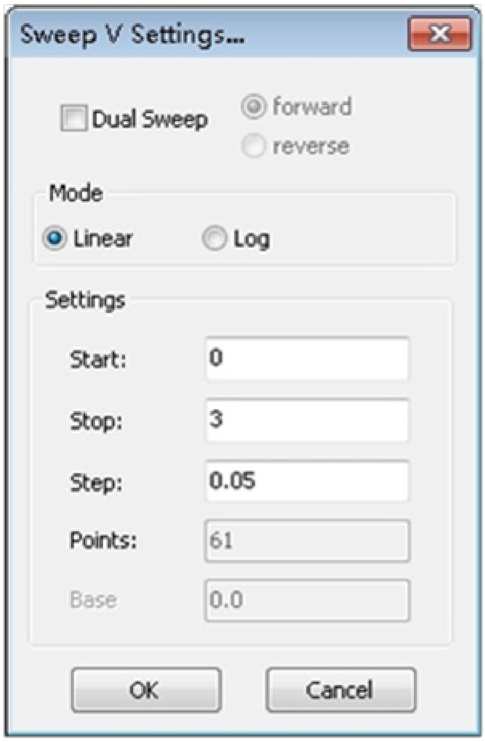
The Bias forcing function maintains a defined constant voltage or current state at the terminal, subject to your specified compliance value of the connected SMU. When Bias V or Bias I is selected, you can set a valid value in the Source cell.

Sweep V, Sweep I

A sweep increments the current or voltage in a series of uniform steps at a speed that is determined by the Speed and Timing settings. When Sweep V or Sweep I is selected, you can set the sweep settings by double clicking the Source cell and making the necessary changes in the dialog box that opens.

Double click the Source cell for SMU that is set to Sweep V/Sweep I and a dialog box for the sweep settings is available.

Figure 107: Source settings for SMU sweep functions



Sweep V Settings...

□ Dual Sweep ● forward ○ reverse

Mode

● Linear ○ Log

Settings

Start: 0

Stop: 3

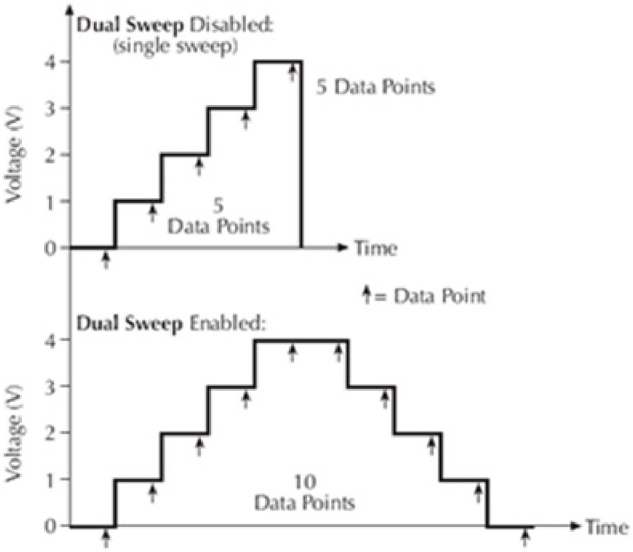
Step: 0.05

Points: 61

Base 0.0

Dual Sweep: An SMU that is configured to perform a sweep, can also be set to perform a dual sweep. With Dual Sweep enabled, the SMU will essentially perform two sweeps. The first sweep steps from the start level to the stop level. The SMU then continues with the second sweep according to the direction chosen (forward or reverse). With Dual Sweep disabled, the SMU performs a single sweep stepping from start to stop.

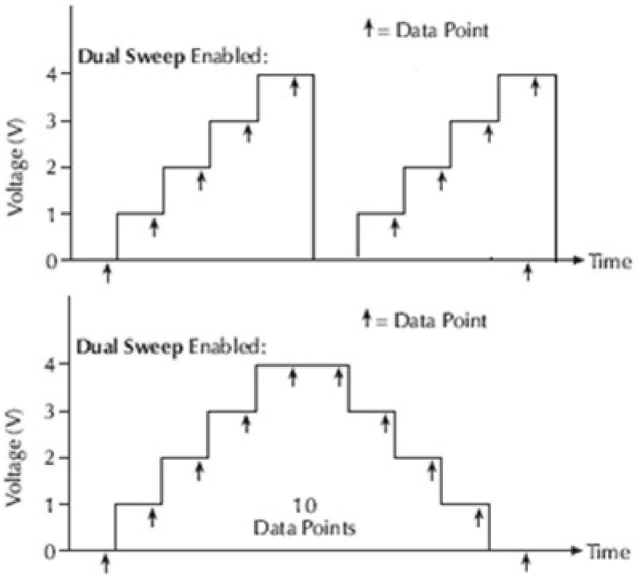
Figure 108: Single and dual sweep examples



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 109: Dual sweep mode - forward and reverse



Mode:

- **Linear**: sweep mode based on the linear axis.
- **Log**: sweep mode based on the logarithm axis.

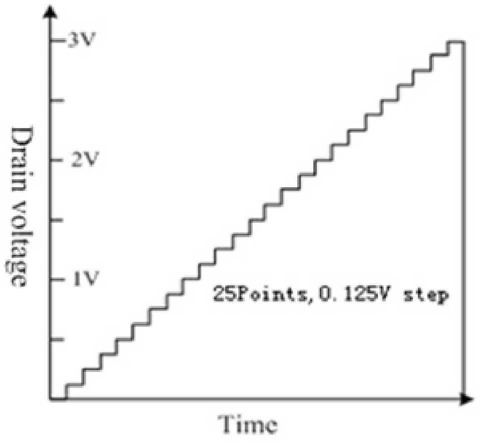
Settings:

- **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 be automatically calculated depending on the Start, Stop, and Points values.

Points: Shows the number of data points that are input when a sweep is executed.

Base: Set the Pulse base level when Pulse Available is selected.

Figure 110: Example sweep drain voltage on a MOSFET



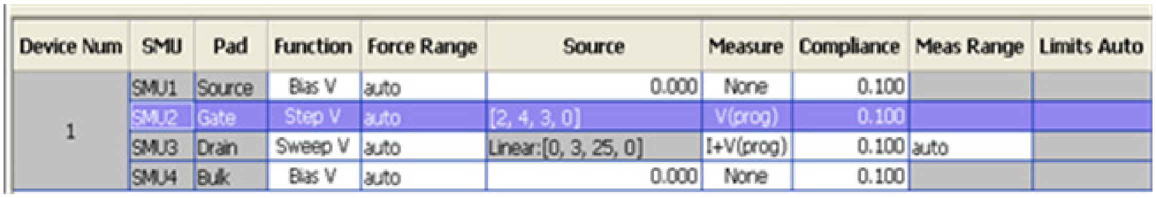
Step V, Step I

A step forcing function incrementally steps a current or voltage two or more levels, each of which is held constant during the progress of a voltage or current sweep at another terminal. For each step parametric curve data is recorded in the ITM Data tab worksheet. The combined data can be plotted in the ITM Data tab graph resulting in a series of curves.

More specifically, the Voltage Step forcing function increments through multiple, evenly-spaced constant voltages, and steps over a user-specified range that is subject to a specified current compliance. The time interval for each step is determined automatically by the time required to complete a sweep.

A typical voltage step forcing functions example is shown in the next figure. This figure illustrates how one Definition tab (for the Vds-Id ITM) specifies both Step and Sweep forcing functions.

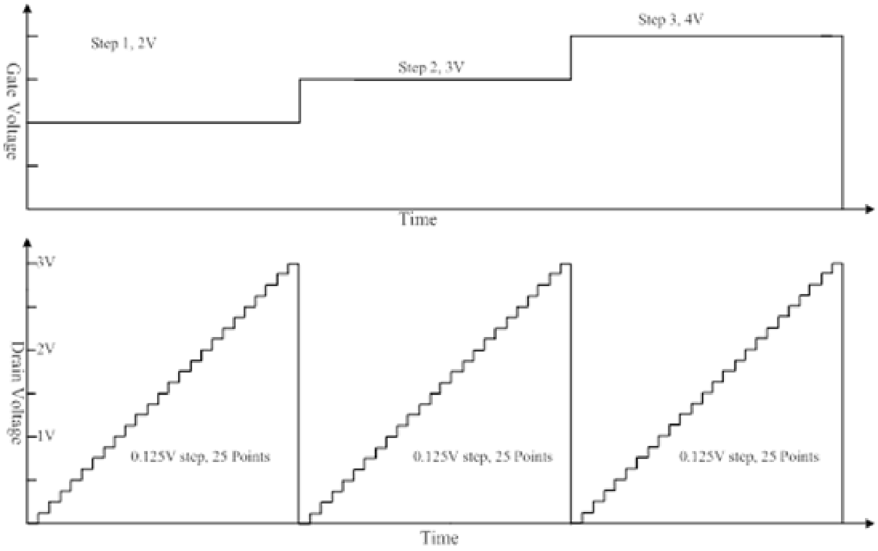
Figure 111: Typical step forcing parameters



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		

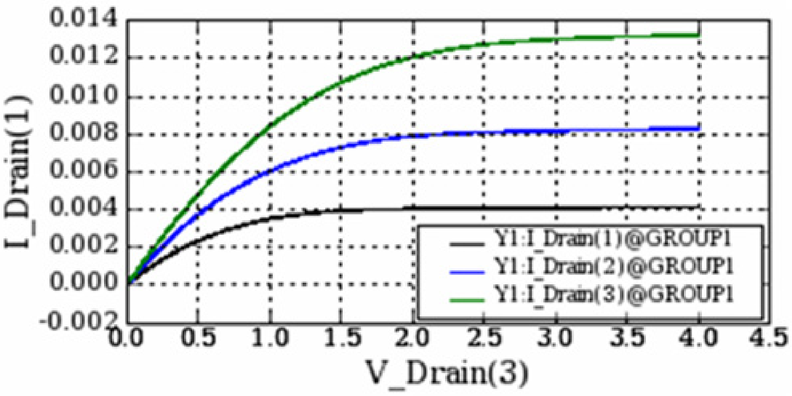
The next figure graphically illustrates the combined Step and Sweep forcing functions specified in the previous figure.

Figure 112: Example MOSFET step gate and sweep drain voltage



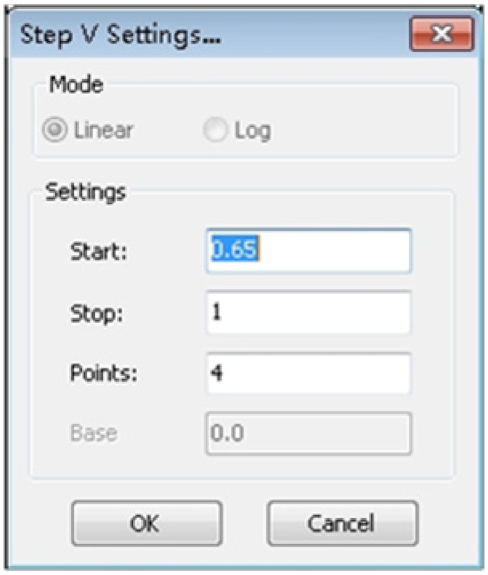
The next figure is a data plot of the graph from the typical forcing parameters figure.

Figure 113: Vds-Id graph



Double click the Source cell of SMU that set as Step V/Step I will pop up the step settings window.

Figure 114: Source settings for SMU step functions



● Linear ○ Log

Start: 0.65

Stop: 1

Points: 4

Base 0.0

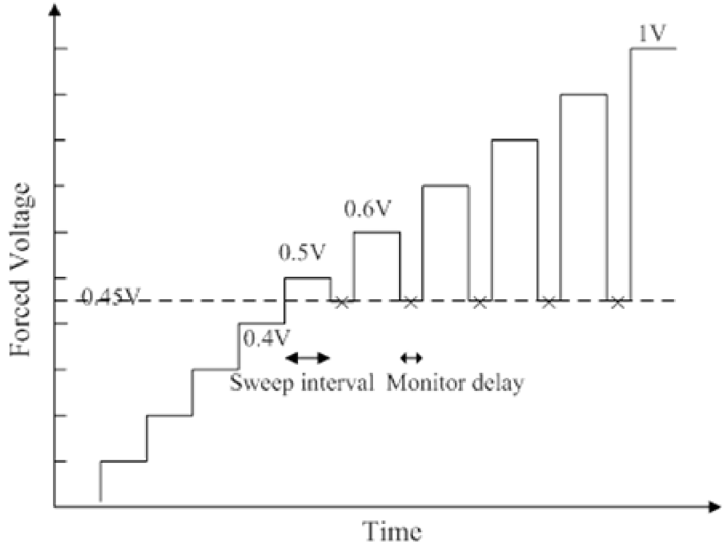
Settings:

- **Start**: Specifies a starting point for a step for any valid SMU voltage or current depending on the source function.
- **Stop**: Specifies a stopping point for a step for any valid SMU voltage or current depending on the source function.
- **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. More than two points muse be entered in the Data Points cell. For example: If Start = 0V, Stop = 4V, and Points = 5, Step value = (4V-0V)/(5-1)=1V Five values are forced 0V, 1V, 2V, 3V, 4V.
- **Base**: Set the Pulse base level when Pulse Available is selected.

Sweep2 V, Sweep2 I

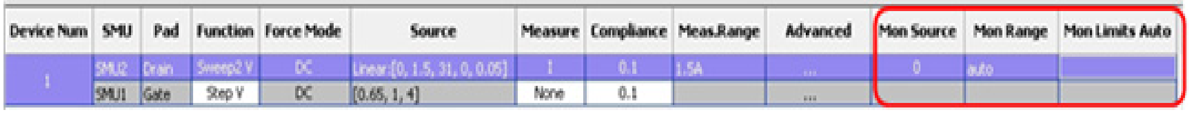
Sweep2 is a special sweep test when used in DC Only mode. When the forced value is greater than the desired value, a monitor test will add to each sweep step, thus the test can generate two test results: measured results and monitored results. More specifically, when each sweep interval increases, it will equal the value of the sweep delay and monitor delay (see next figure; the sweep 2V waveform has a 0.45V as the monitor source voltage).

Figure 115: Sweep 2V waveform



Click the Function cell and in the drop-down list choose Sweep2 V or Sweep2 I. The monitor items will append to the Force and Meas cell (see next figure)

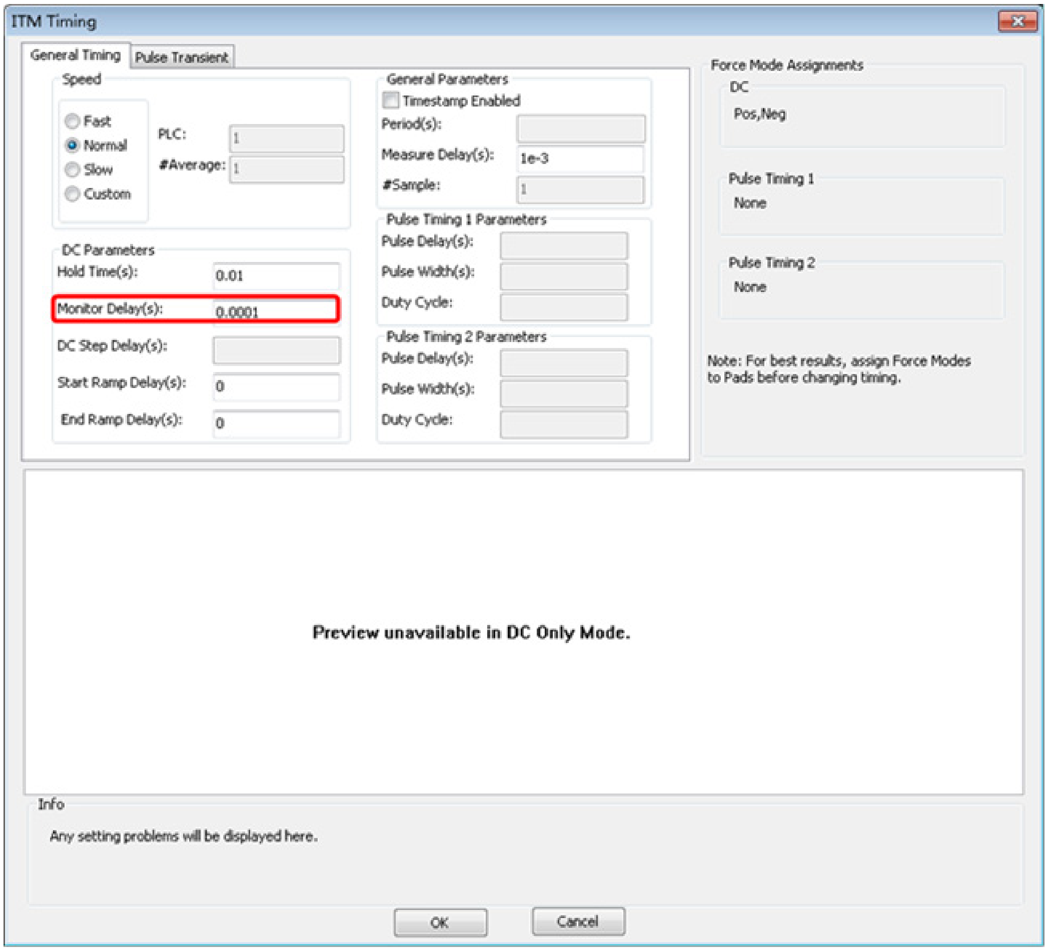
Figure 116: The SMU monitor items



Device Num	SMU	Pad	Function	Force Mode	Source	Measure	Compliance	Meas.Range	Advanced	Mon Source	Mon Range	Mon Limits Auto
1	SMU2	Drain	Sweep2 V	DC	Linear:[0, 1.5, 31, 0, 0.05]	I	0.1	1.5A	...	0	auto	
	SMU1	Gate	Step V	DC	[0.65, 1, 4]	None	0.1		...			

Also, the monitor delay in Time dialog will be enabled (see next figure).

Figure 117: Enable monitor delay



○ Fast
● Normal
○ Slow
○ Custom

PLC: 1

#Average: 1

Hold Time(s): 0.01

Monitor Delay(s): 0.0001

DC Step Delay(s):

Start Ramp Delay(s): 0

End Ramp Delay(s): 0

□ Timestamp Enabled

Measure Delay(s): 1e-3

#Sample: 1

DC: Pos,Neg

Pulse Timing 1: None

Pulse Timing 2: None

Note: For best results, assign Force Modes to Pads before changing timing.

Preview unavailable in DC Only Mode.

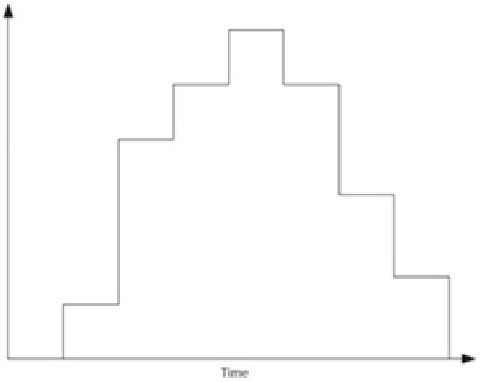
Any setting problems will be displayed here.

List V, List I

List V and List I are used to input a voltage and current list according to the test's requirements. The SMU will force the input values one by one like a sweep.

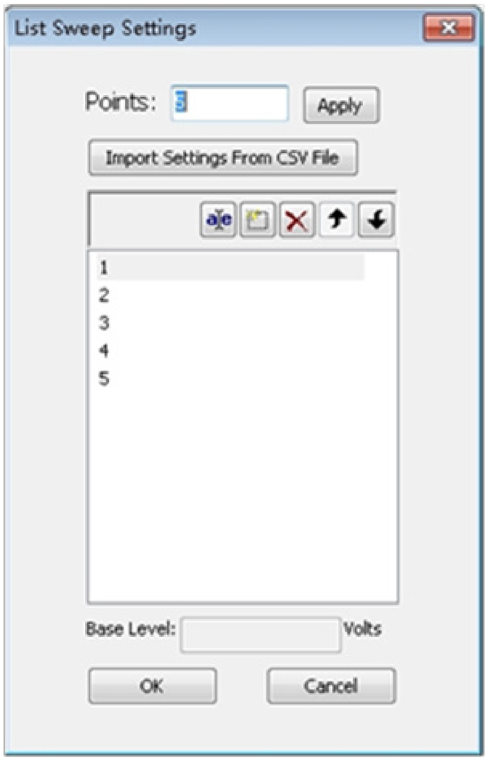
List Sweeps allow you to make measurements only at selected forced voltages and currents. For example, they allow you to skip measurement points or to synthesize a custom sweep that is based on a mathematical equation.

Figure 118: List sweep general illustration



Double click the Source cell of the SMU and the List Sweep Settings dialog box opens (see next figure).

Figure 119: List Sweep Settings dialog box



Points: 5

1
2
3
4
5

Base Level: Volts

Points: the number of points for the forcing values.

You can import (and then save) a .csv file by clicking the Import Setting From CSV File button (see previous figure).

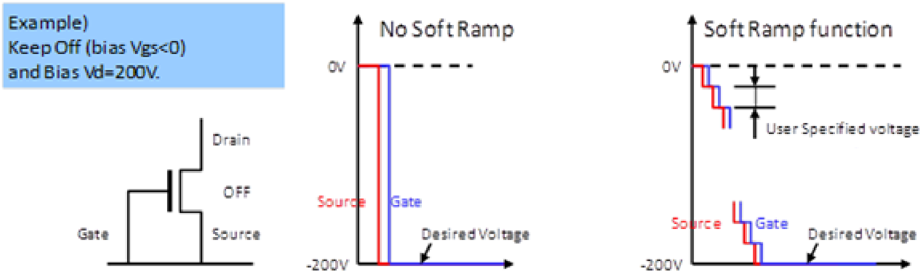
- **Edit item** : the default number can be changed to a different value.
- **New item** □: a new item is added when you click this icon.
- **Delete item** ☒: a selected item is deleted when you click this icon.
- **Move up** ↑: a selected item moves up when you click this icon.
- **Move down** ↓: a selected item moves down when you click this icon.
- **Base Level**: the Pulse base level when Pulse Available is selected.

Soft Ramp Bias

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

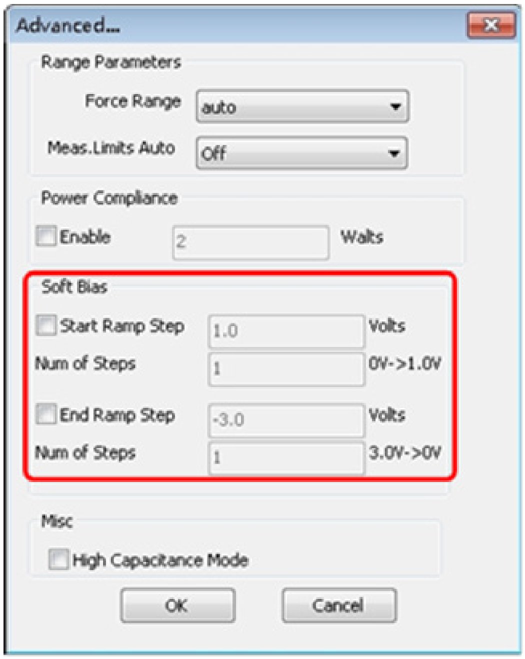
The Soft Ramp is from 0 to the desired value (bias value or first and last sweep/list value). If the bias or first or last sweep/list point is 0, no soft ramp is necessary. In the "No Soft Ramp" example, the gate breakdown occurs because of the 200V stress applied. However, with the "Soft Ramp function,"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

Figure 120: Comparison of Soft Ramp and without Soft Ramp



Soft Ramp is enabled in the Advanced settings in the SMU forcing and measuring settings table. The Soft Ramp timing is set in the timing settings window (see next figures).

Figure 121: Advanced Soft Bias settings



Advanced...

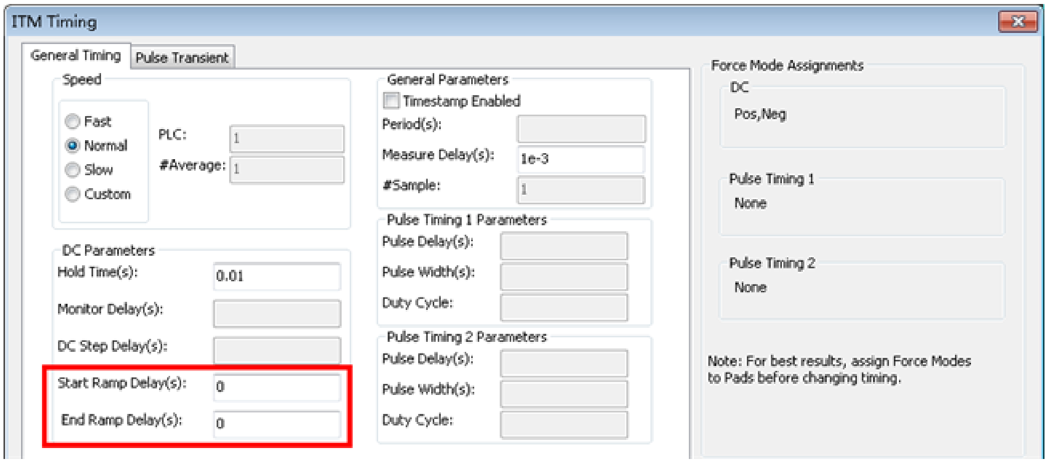
Range Parameters
Force Range: auto
Meas.Limits Auto: Off

Power Compliance
□ Enable 2 Watts

Soft Bias
□ Start Ramp Step 1.0 Volts
Num of Steps 1 0V->1.0V
□ End Ramp Step -3.0 Volts
Num of Steps 1 3.0V->0V

Misc
□ High Capacitance Mode

Figure 122: ITM General Timing tab settings



ITM Timing

Speed
○ Fast
● Normal
○ Slow
○ Custom
PLC: 1
#Average: 1

DC Parameters
Hold Time(s): 0.01
Monitor Delay(s):
DC Step Delay(s):
Start Ramp Delay(s): 0
End Ramp Delay(s): 0

General Parameters
□ Timestamp Enabled
Period(s):
Measure Delay(s): 1e-3
#Sample: 1

Pulse Timing 1 Parameters
Pulse Delay(s):
Pulse Width(s):
Duty Cycle:

Pulse Timing 2 Parameters
Pulse Delay(s):
Pulse Width(s):
Duty Cycle:

Force Mode Assignments
DC
Pos,Neg
Pulse Timing 1
None
Pulse Timing 2
None

Note: For best results, assign Force Modes to Pads before changing timing.

If the Start Ramp and End Ramp are selected, the test will apply a Soft Ramp before and after the Bias or Sweep. The Soft Ramp is from 0 to the desired value (Bias value or first and last Sweep, List value). If the Bias or first or last Sweep, List point is 0, no Soft Ramp is necessary.

In the Advanced settings dialog box, setting the the Ramp Step, the Num of Steps will be automatically calculated.

Start Ramp Step: The step value determines when to start the ramp.

Stop Ramp Step: The step value determines when to stop the ramp.

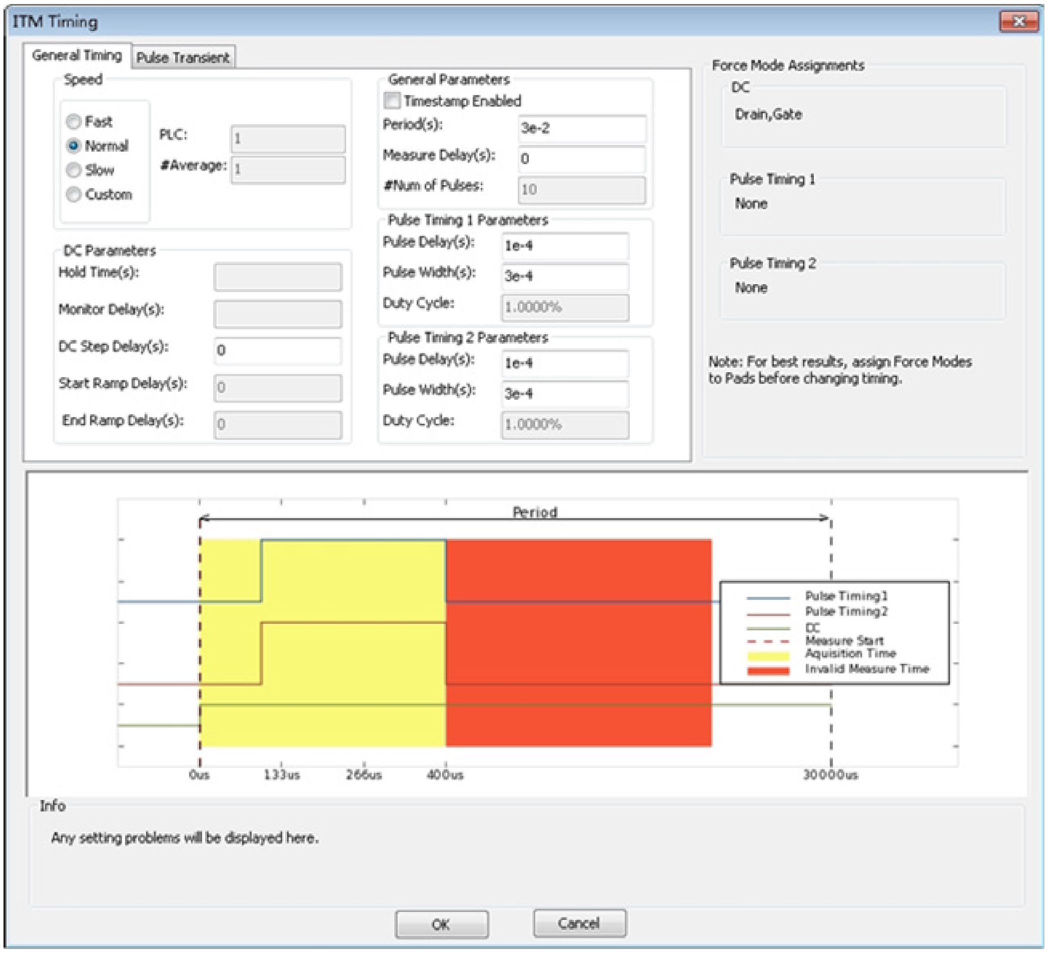
Num of Steps: The number of steps defined for all of the SMUs, however, the start and stop steps are defined separately. The default value for Num of steps is 1. It's value depends on the Force Value and the Start/Stop Ramp Steps.

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

ITM timing settings

When you click the ITM Timing icon in the ITM Definition tab, the ITM Timing GUI opens (see next figure). The ITM Timing GUI is used to configure ITM timing settings.

Figure 123: ITM Timing GUI



ITM Timing

General Timing | Pulse Transient

Speed

○ Fast
● Normal
○ Slow
○ Custom

PLC: 1

#Average: 1

DC Parameters

Hold Time(s):

Monitor Delay(s):

DC Step Delay(s): 0

Start Ramp Delay(s): 0

End Ramp Delay(s): 0

General Parameters

□ Timestamp Enabled

Period(s): 3e-2

Measure Delay(s): 0

#Num of Pulses: 10

Pulse Timing 1 Parameters

Pulse Delay(s): 1e-4

Pulse Width(s): 3e-4

Duty Cycle: 1.0000%

Pulse Timing 2 Parameters

Pulse Delay(s): 1e-4

Pulse Width(s): 3e-4

Duty Cycle: 1.0000%

Force Mode Assignments

DC

Drain,Gate

Pulse Timing 1

None

Pulse Timing 2

None

Note: For best results, assign Force Modes to Pads before changing timing.

Info

Any setting problems will be displayed here.

Speed

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

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

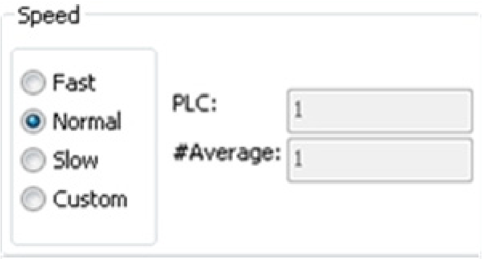
Fast, Normal, Quiet, and Custom modes are defined as follows:

- **Fast**: Optimizes speed at the expense of noise performance. It is a good choice for fast measurements where noise and settling time are not concerns.
- **Normal**: The default and most commonly used setting. It provides a good combination of speed and low noise, and is the best setting for most cases.
- **Slow**: Optimizes the low-noise measurements at the expense of speed. If speed is not a critical consideration, it is a good 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 edit 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 msec (1/60 of a second).

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

Figure 124: Speed settings for an ITM



Speed

○ Fast
● Normal
○ Slow
○ Custom

PLC: 1

#Average: 1

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 Quiet measurement Speed mode, ACS Basic sets the A/D Integration Time correspondingly. The next table summarizes the allowed A/D Integration Time settings for various measurement Speed modes.

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

General parameters

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

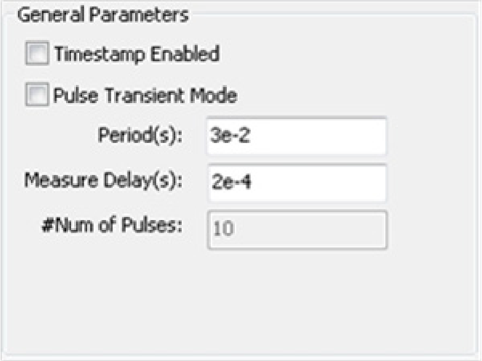
Pulse Transient Mode: if checked, enable transient measurement of pulse mode. And Fast ADC is selected, and PLC is automatically set to 0.001 and #Average to 1.

Period(s): pulse period, unit is second. All the two pulse timing have the same period.

Measure Delay(s): delay time before measurement.

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

Figure 125: General Parameters ITM timing settings



General Parameters

□ Timestamp Enabled

□ Pulse Transient Mode

Period(s): 3e-2

Measure Delay(s): 2e-4

#Num of Pulses: 10

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.

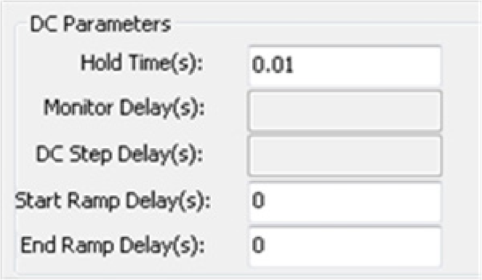
Monitor Delay(s): Sets the monitor delay time in the second sweep phase in of the Sweep2 function.

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 126: DC Parameters ITM timing settings



DC Parameters

Hold Time(s): 0.01

Monitor Delay(s):

DC Step Delay(s):

Start Ramp Delay(s): 0

End Ramp Delay(s): 0

Pulse Timing Parameters

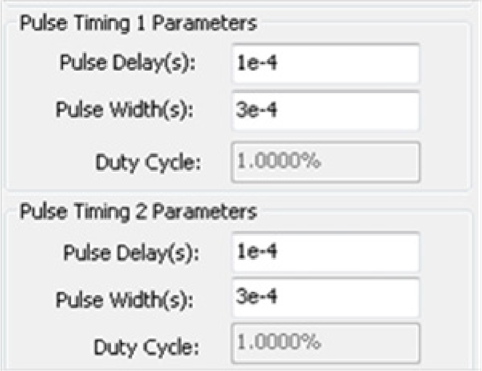
ACS Basic allows you to configure a total of two pulse timing configurations.

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 Model Pulse Timing 2 Parameters. Duty cycle is calculated by dividing the pulse width by the period. Duty cycle = Pulse width / period.

Figure 127: Pulse Timing Parameters for an ITM



Pulse Timing 1 Parameters

Pulse Delay(s): 1e-4

Pulse Width(s): 3e-4

Duty Cycle: 1.0000%

Pulse Timing 2 Parameters

Pulse Delay(s): 1e-4

Pulse Width(s): 3e-4

Duty Cycle: 1.0000%

Force Mode Assignments

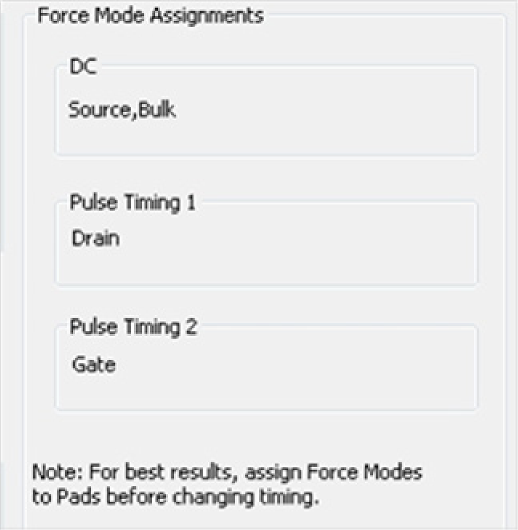
This part displays the force mode assignments for each pad (SMU). It is recommended to assign Force Mode to Pads (SMUs) in the SMU forcing and measuring settings table before changing timing settings.

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 128: Force Mode Assignments for an ITM



Force Mode Assignments

DC

Source,Bulk

Pulse Timing 1

Drain

Pulse Timing 2

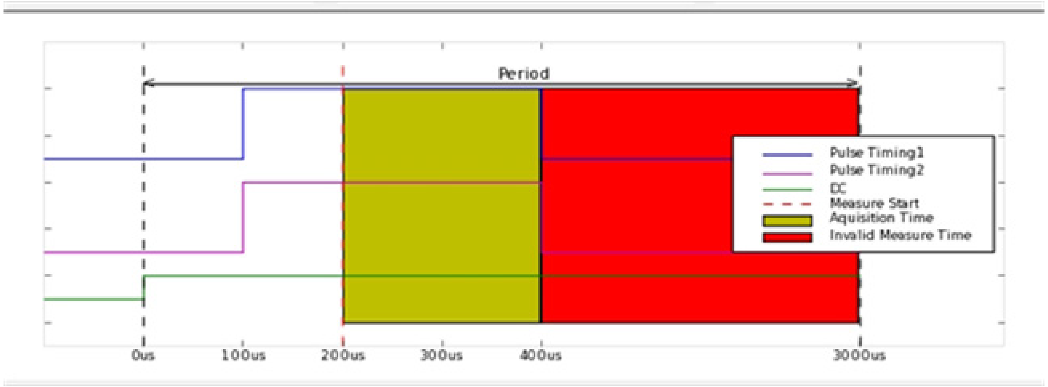
Gate

Note: For best results, assign Force Modes to Pads before changing timing.

Timing schematic diagram

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

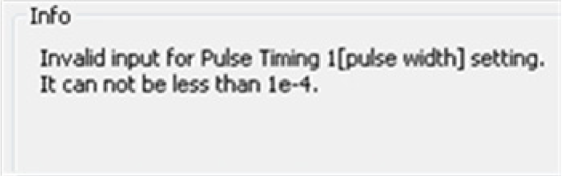
Figure 129: Timing schematic diagram for an ITM



Info

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

Figure 130: Info in the ITM timing settings



Info

Invalid input for Pulse Timing 1[pulse width] setting. It can not be less than 1e-4.

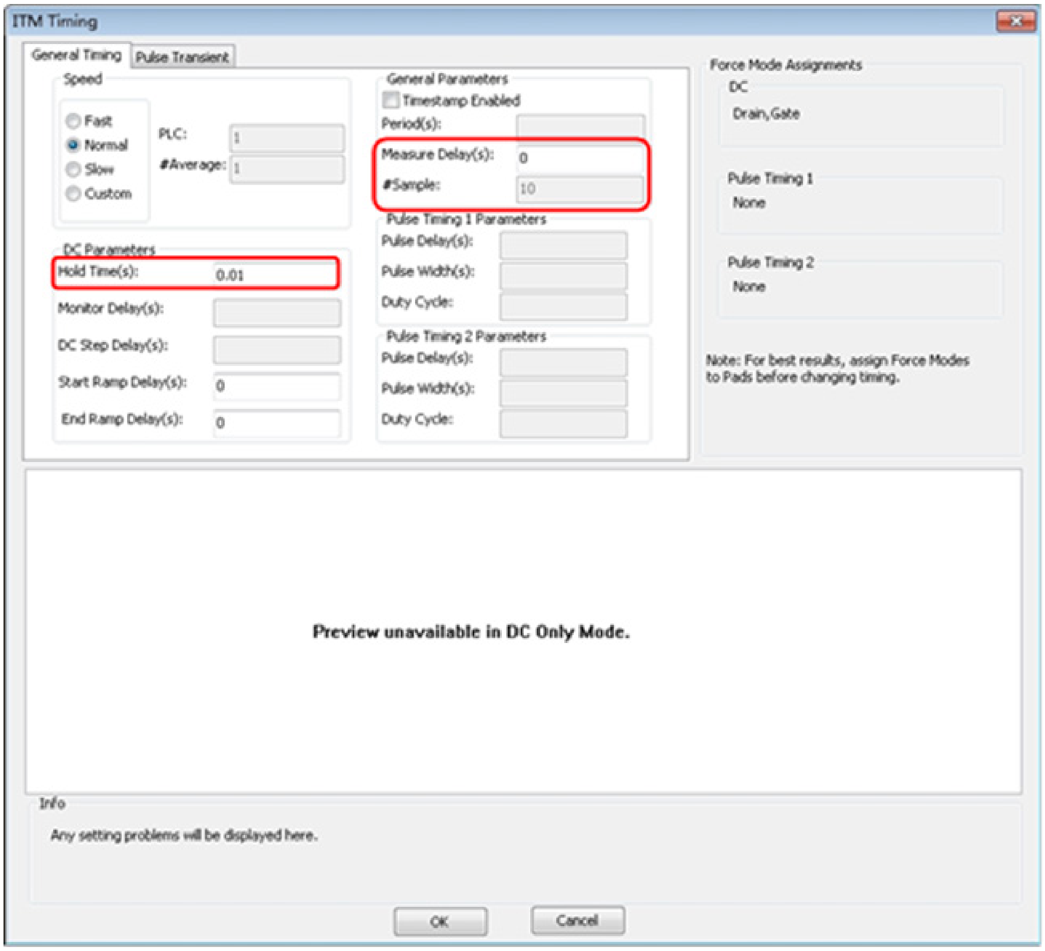
DC Only Timing settings

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

#Sample: when all the SMUs are set to Bias, sets the sample number of samples for the sampling measurement. # Samples is only available when all the SMUs are set to the Bias forcing function.

Sweep: 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.

Figure 131: ITM Timing for DC only



PLC: 1

#Average: 1

General Parameters

□ Timestamp Enabled

Period(s):

Measure Delay(s): 0

#Sample: 10

DC Parameters

Hold Time(s): 0.01

Monitor Delay(s):

DC Step Delay(s):

Start Ramp Delay(s): 0

End Ramp Delay(s): 0

Pulse Timing 1 Parameters

Pulse Delay(s):

Pulse Width(s):

Duty Cycle:

Pulse Timing 2 Parameters

Pulse Delay(s):

Pulse Width(s):

Duty Cycle:

Force Mode Assignments

DC

Drain,Gate

Pulse Timing 1

None

Pulse Timing 2

None

Note: For best results, assign Force Modes to Pads before changing timing.

Preview unavailable in DC Only Mode.

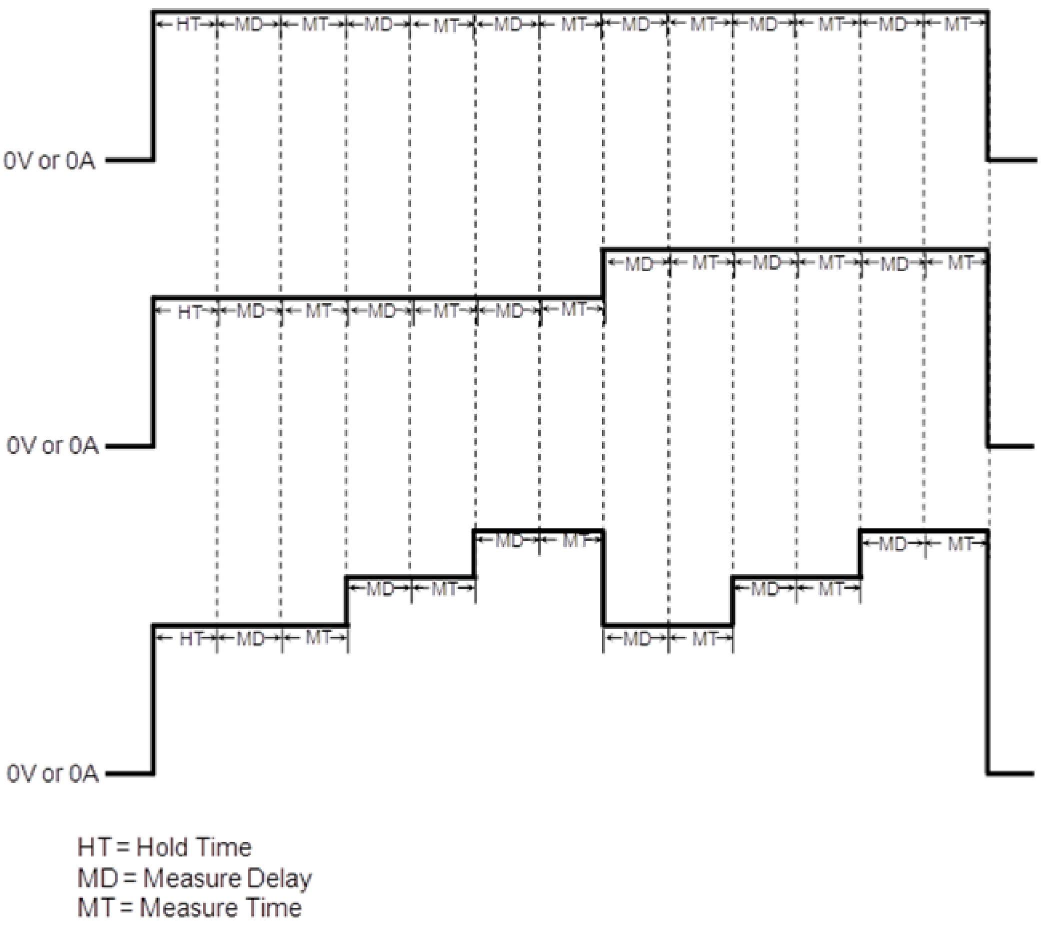
Info

Any setting problems will be displayed here.

Measure 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 edit box. You can specify a Measure Delay from 0 to 1000 seconds. The default Measure Delay is 0s.

Hold Time: The starting voltage or current of a sweep may be substantially larger than the voltage/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 of 0 to 1000 seconds. The default Hold Time is 0s.

Figure 132: Sweep timing diagram in DC only

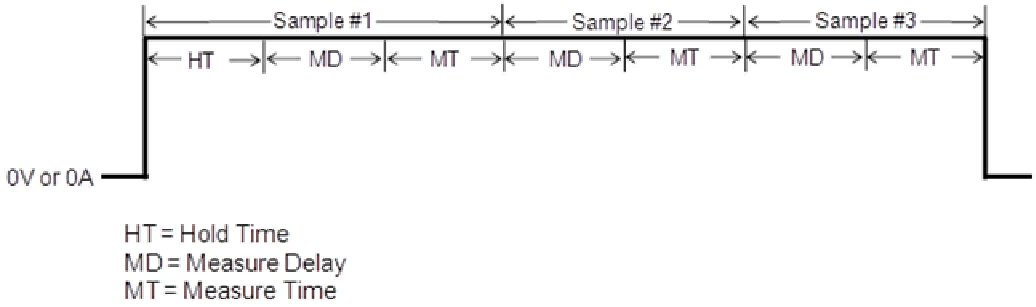


Sampling

The Sampling mode measures voltage and current as a function of time while forcing constant voltages/currents (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 SMU(s) apply the forced voltage or current (that is, t = 0 at the step change from 0.0V/0.0A to the applied voltage/current).

ACS Basic 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 is must in DC Only mode.

Figure 133: Sampling timing diagram in DC only



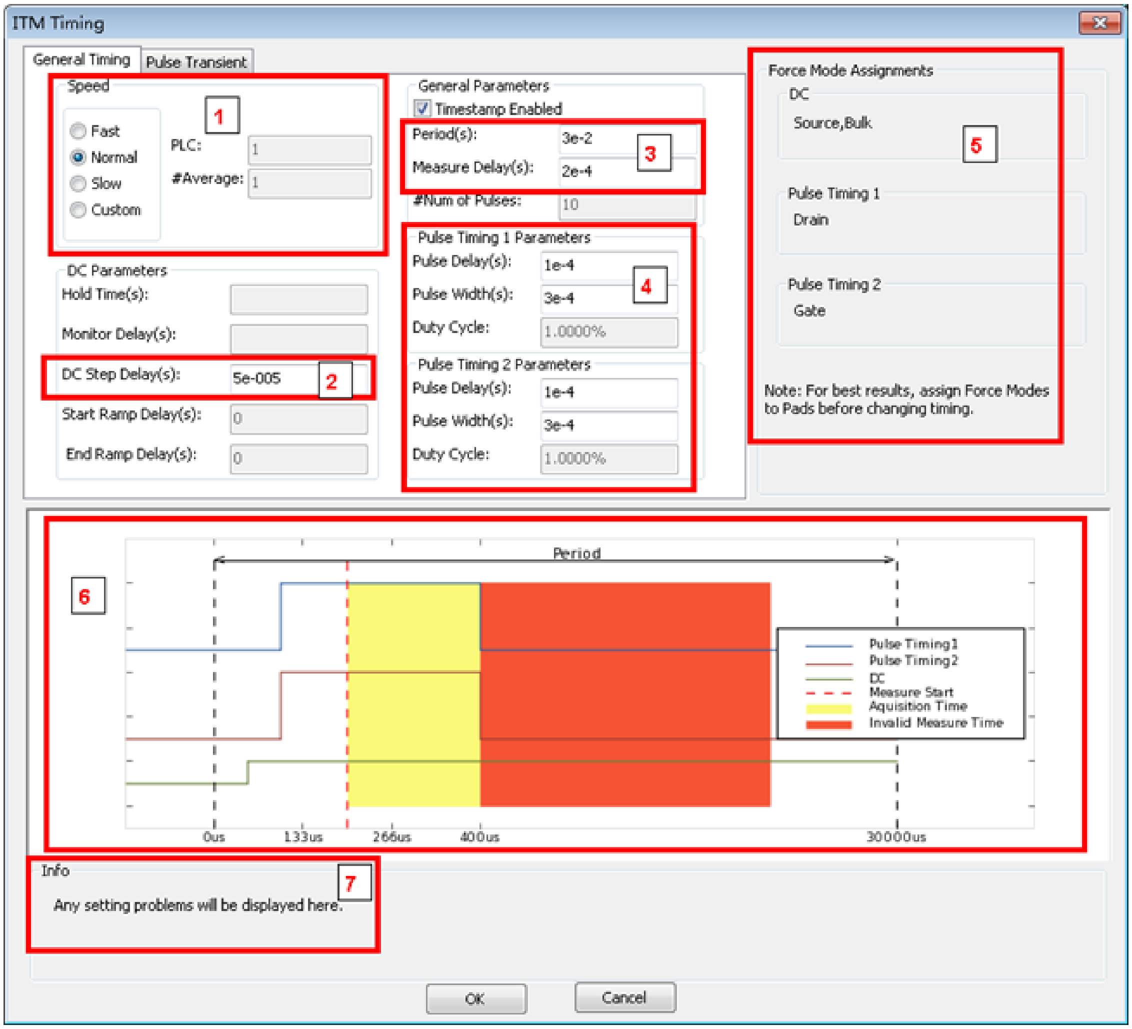
Pulse Available Timing settings

When the Pulse Available mode is selected, the ITM will allow DC timing and two pulse timing settings for the SMUs. All of 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 next figure indicates the settings for the pulse timing:

1. Measure speed: PLC and average number in acquisition time.
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 134: Pulse timing settings



ITM Timing

General Timing | Pulse Transient

Speed
○ Fast
● Normal
○ Slow
○ Custom
PLC: 1
#Average: 1

General Parameters
☑ Timestamp Enabled
Period(s): 3e-2
Measure Delay(s): 2e-4
#Num of Pulses: 10

Pulse Timing 1 Parameters
Pulse Delay(s): 1e-4
Pulse Width(s): 3e-4
Duty Cycle: 1.0000%

Pulse Timing 2 Parameters
Pulse Delay(s): 1e-4
Pulse Width(s): 3e-4
Duty Cycle: 1.0000%

DC Parameters
Hold Time(s):
Monitor Delay(s):
DC Step Delay(s): 5e-005
Start Ramp Delay(s): 0
End Ramp Delay(s): 0

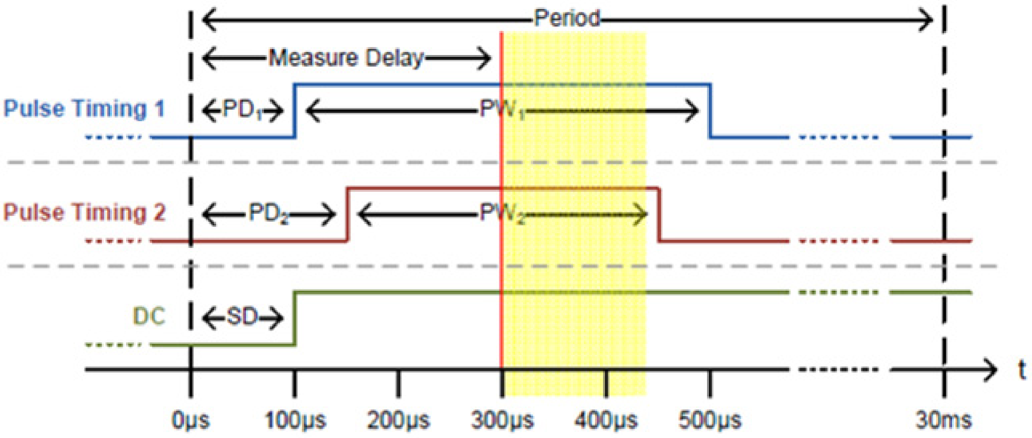
Force Mode Assignments
DC
Source,Bulk
Pulse Timing 1
Drain
Pulse Timing 2
Gate

Note: For best results, assign Force Modes to Pads before changing timing.

Info
Any setting problems will be displayed here.

OK Cancel

Figure 135: Pulse timing diagram

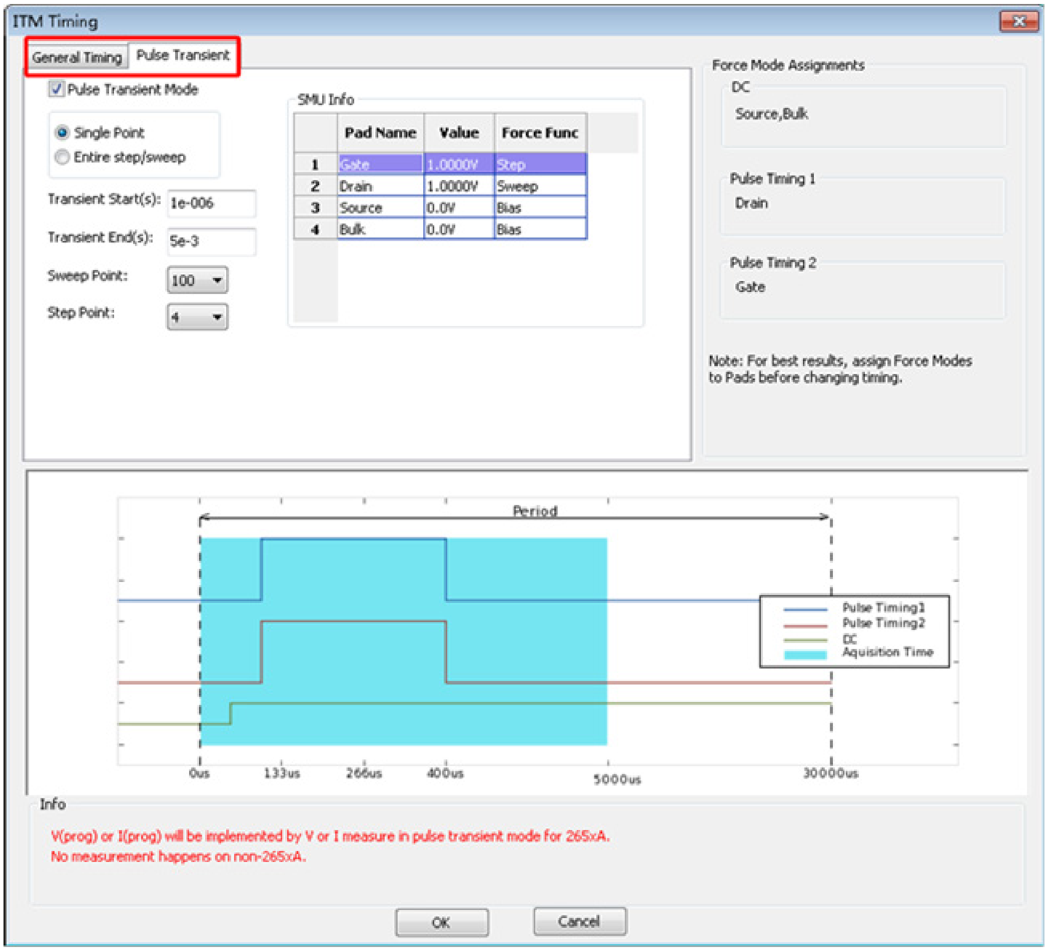


PD_1 – Pulse Delay for Timing 1
PW_1 – Pulse Width for Timing 1
PD_2 – Pulse Delay for Timing 2
PW_2 – Pulse Width for Timing 2
SD – DC Step Delay

Pulse Transient

Pulse Transient mode uses the Fast ADC that is available on certain SMU models (details about Fast ADC, please refer to the reference manual of 265xA SMU) to capture the waveform for a single point in the step/sweep sequence. It is useful for determining the proper measurement delay setting for the step/sweep sequence by capturing the settling time of the SMU outputs. It is also useful for debugging troublesome data points in the step/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 window on the Pulse Transient tab (see next figure).

Figure 136: Pulse Transient mode settings



☑ Pulse Transient Mode

● Single Point
○ Entire step/sweep

Transient Start(s): 1e-006

Transient End(s): 5e-3

Sweep Point: 100

Step Point: 4

SMU Info

	Pad Name	Value	Force Func
1	Gate	1.0000V	Step
2	Drain	1.0000V	Sweep
3	Source	0.0V	Bias
4	Bulk	0.0V	Bias

Force Mode Assignments

DC
Source,Bulk

Pulse Timing 1
Drain

Pulse Timing 2
Gate

Note: For best results, assign Force Modes to Pads before changing timing.

Info

V(prog) or I(prog) will be implemented by V or I measure in pulse transient mode for 265xA.
No measurement happens on non-265xA.

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

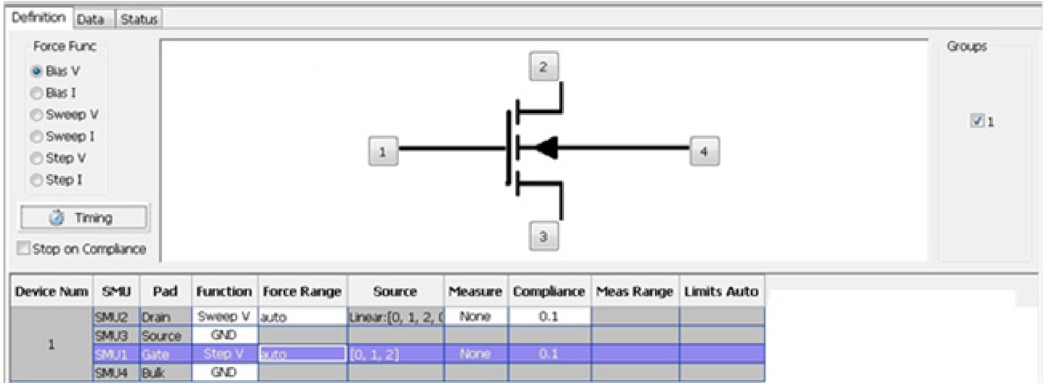
Step Point: Sets the step point that will be captured.

The Sweep Point and the Step Point together appoint a certain period for waveform capture. The SMU info 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 both DC and pulse measurement, however, it does not have pulse mode. 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 (see next figure).

Figure 137: Definition tab of the Model 42XX SMU ITM



Force Func
● Bias V
○ Bias I
○ Sweep V
○ Sweep I
○ Step V
○ Step I

□ Stop on Compliance

Groups
☑ 1

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						

Timing: This button opens the ITM timing dialog box where you can set detailed timing settings for pulse and synchronization between SMUs (refer to the Timing dialog box for more information).

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.

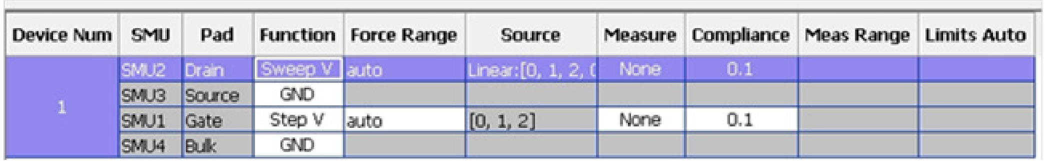
Groups: Select the group or groups where you want to run the test. The group is determined by the GPIB or Ethernet address. Make sure that all the groups you select have the same hardware configurations as your ITM.

SMU parameters: the forcing and measuring settings table is where you can set your parameters for testing.

Model 42XX SMU parameters

The following figure displays the items that you can use to set up your testing for forcing and measuring functions for each SMU.

Figure 138: 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; based on the hardware connection.

SMU: SMU number defined in the Device work area; based on the hardware connection.

Pad: Pad name defined in the Device work area; based on the hardware connection.

Function: Force function type.

Force Range: Specifies the SMU current range used to force the sweep current:

- 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 an 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 10mA 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.

Measure: Measure the current unit in amps and the voltage unit in volts. All options are:

- I: Actual measured current values.
- V: Actual measured voltage values.
- V(prog): The programmed voltage, as requested forced voltage values. For example, 2.5V is to be forced at a transistor drain terminal. The programmed value [V(prog)]) is exactly 2.5V, even if the measured value (V) is 2.4997V. Under the programmed option, the reported value is exactly 2.5V, even if the measured value is 2.4997V.
- I(prog): The programmed current, as requested forced current values, when the source function is voltage.
- I+V: Measures both I and V.
- I+V(prog): Measures I and returns V(prog).
- V+I(prog): Measures V and returns I(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 I is selected in the Measure cell, ACS Basic 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:

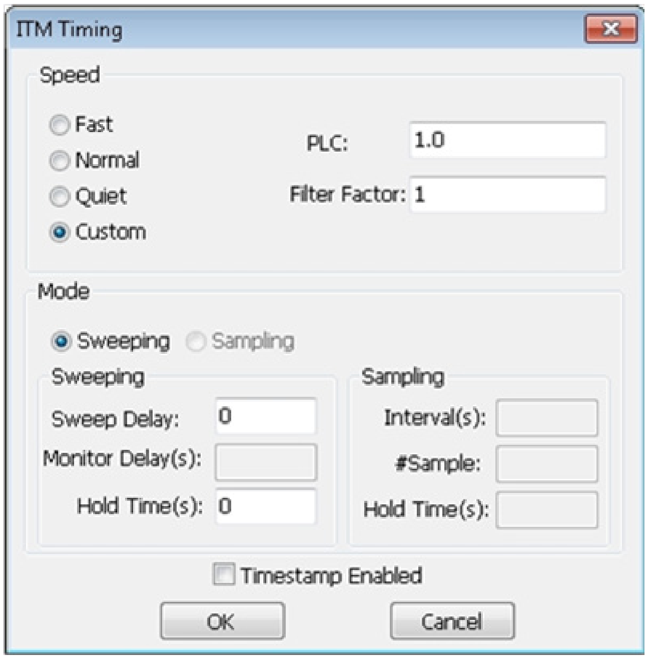
- Auto option commands the SMU to automatically optimize the current measurement range. This option provides the best resolution
- Limits Auto is only enabled when the force function is voltage and there is current measurement. This option is a compromise between the full Auto option and a fixed range 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 of 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.

Timing

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

Figure 139: 42XX SMU ITM Timing settings



ITM Timing

Speed

○ Fast
○ Normal
○ Quiet
● Custom

PLC: 1.0
Filter Factor: 1

Mode

● Sweeping ○ Sampling

Sweeping

Sweep Delay: 0
Monitor Delay(s):
Hold Time(s): 0

Sampling

Interval(s):
#Sample:
Hold Time(s):

□ Timestamp Enabled

Speed

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

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

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

- Fast: Optimizes speed at the expense of noise performance. It is a good choice for fast measurements where noise and settling time are not concerns.
- Normal: The default and most commonly used setting. It provides a good combination of speed and low noise, and is the best setting for most cases.
- Slow: Optimizes the low-noise measurements at the expense of speed. If speed is not a critical consideration, it is a good 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 edit 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 msec (1/60 of a second).

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

Mode area

There are two test modes you can select in the Mode area of the ITM Timing GUI: Sweeping and Sampling.

Sweeping: Applies to any ITM where one or more forced voltages/currents 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 is used to record a few static measurements, or to time-profile the charging voltage of a capacitor while forcing a constant current.

The Mode is used to observe and/or change the test mode as follows:

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:

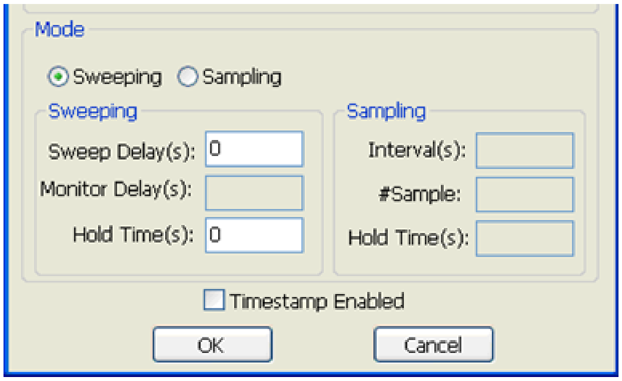
1. Only the static forcing functions are configurable in the Sampling mode.
2. Only mode-appropriate timing options are configurable.

For an existing library ITM 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 dynamic forcing functions.

Sweep 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 GUI (Sweep Delay and Hold Time). You can configure these settings for all measurement Speed modes: Fast, Normal, Quiet, and Custom.

Figure 140: ITM Sweep timing GUI



Mode

● Sweeping ○ Sampling

Sweeping

Sweep Delay(s): 0

Monitor Delay(s):

Hold Time(s): 0

Sampling

Interval(s):

#Sample:

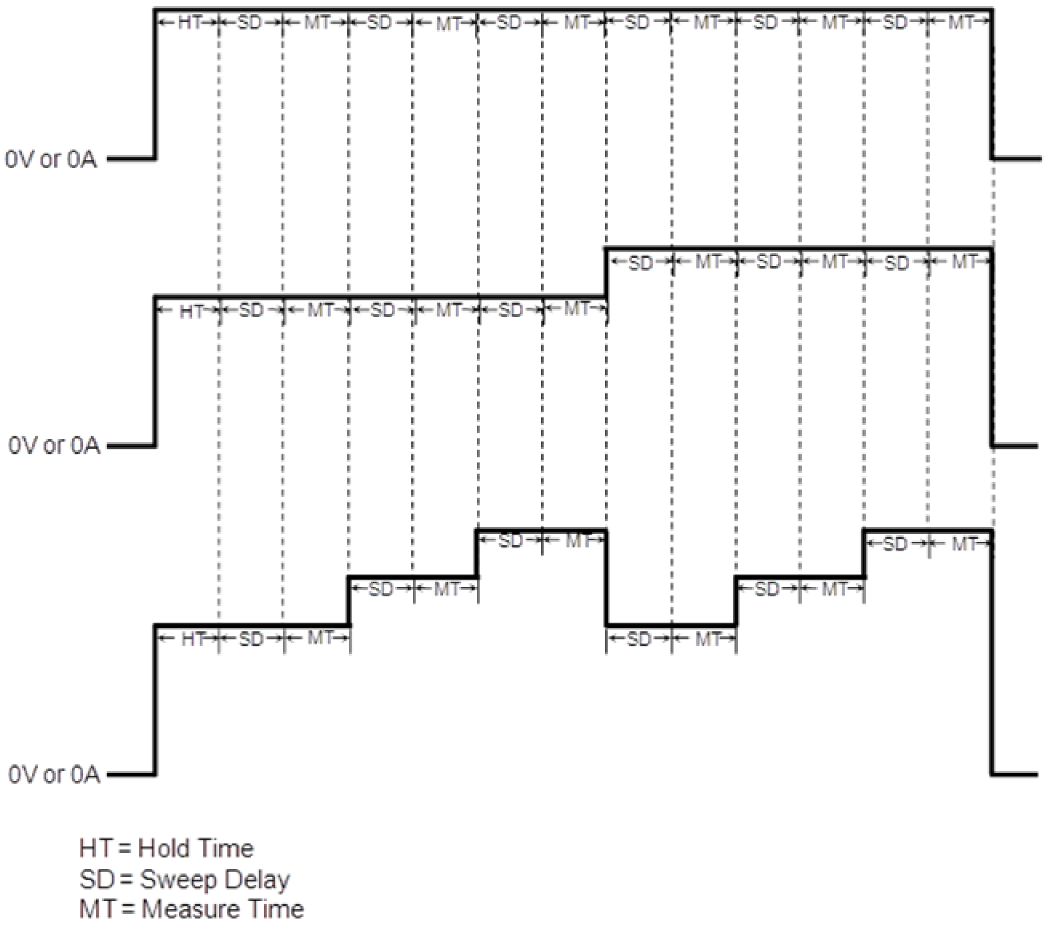
Hold Time(s):

□ Timestamp Enabled

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 edit box. You can specify a Sweep Delay from 0 to 1000 seconds. The default Sweep Delay is 0s.

Hold Time: The starting voltage or current of a sweep may be substantially larger than the voltage/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 0s (see next figure).

Figure 141: 42XX sweep mode timing diagram



Hold Time: When the test is started, hold time (HT) 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 applied at the start (only) of each sweep.

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.

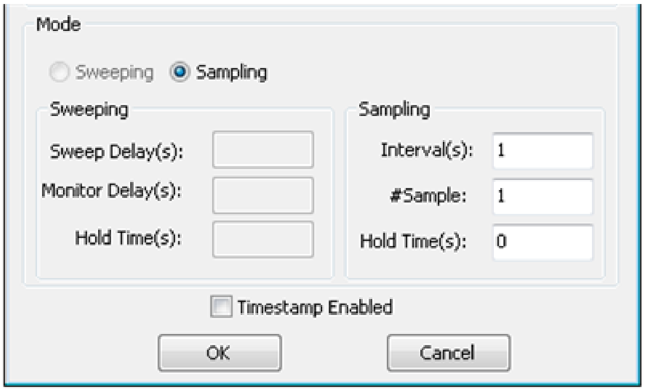
Measure Time: Determined by the # Average and the A/D Integration Time (NPLC). All SMUs in the test system are synchronized. The measure time for the SMU requiring the longest measure time is the same for all SMUs in the test system.

Sampling mode

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

ACS Basic 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 142: ITM sample timing settings



Mode

○ Sweeping ● Sampling

Sweeping

Sweep Delay(s):

Monitor Delay(s):

Hold Time(s):

Sampling

Interval(s): 1

#Sample: 1

Hold Time(s): 0

□ Timestamp Enabled

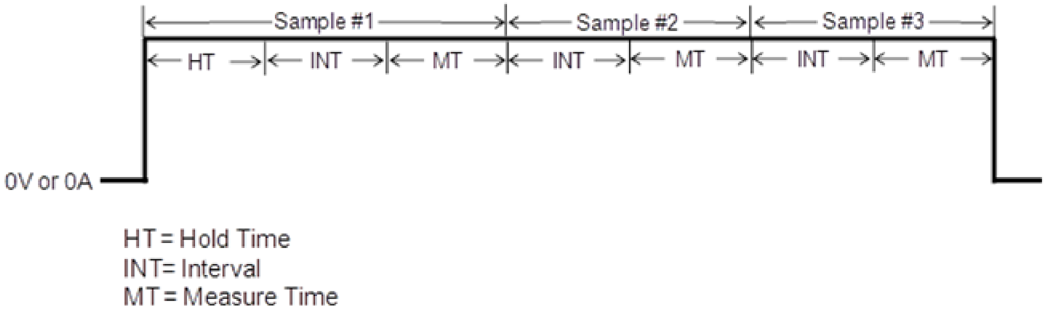
If an ITM is configured for the Sampling mode, you can configure three Sampling mode settings: Interval, #Sample, and Hold Time.

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 SMU(s), the source settling time(s) 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 (see next figure).

Figure 143: 42XX sampling mode timing diagram



Hold Time: If needed, 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.

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.

Configure a script test module (STM)

A STM is a test module that supports test script processor (TSP) files. A TSP file is written in test script language (TSL) for testing a device with a Series 2600B, Model 3706A, or Model 707B/708B in Instrument Control Library (ICL) mode. Several STM examples and tests are installed with ACS Basic. These scripts can be imported and modified in the STM work area.

TSP files can also have a GUI associated with them to simplify configuring tests. STM GUIs can either be the standard GUI built into ACS Basic or a user-created XML Resource (XRC) GUI.

The Test Script

A script is a collection of instrument control commands, and programming statements. The statements control script execution and provide facilities 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 Language (TSL). TSL is derived from the Lua scripting language. Descriptions of various commands and examples can be found in the ACS Basic Libraries Reference Manual.

STM Script examples

Scripts can be written in the STM or using the Test Script Builder (see Script Editor for more information). Scripts are loaded into the Series 2600B and can be saved in nonvolatile memory. Scripts not saved in non-volatile memory will be lost from instrument memory when the Series 2600B is turned off.

The script below is a script used to perform the resistance measurement.

Example:

```
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 function

TSL 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. TSL allows you to define functions. A function can take a predefined number of parameters and return multiple parameters if desired.

To define a function and call that function, see the example below:

Example:

```
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 to perform the resistance measurements.

Example:

```
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
```

Call the function as shown below:

```
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. This causes the following:

- In the Data tab of a test module in Test Setup, you will see the variable sequence is the same as in the OUTPUT list.
- This OUTPUT list is of the highest priority compared to "postdata." If you add the OUTPUT list, postdata will not be scanned or 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.

Add the following OUTPUT header:

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

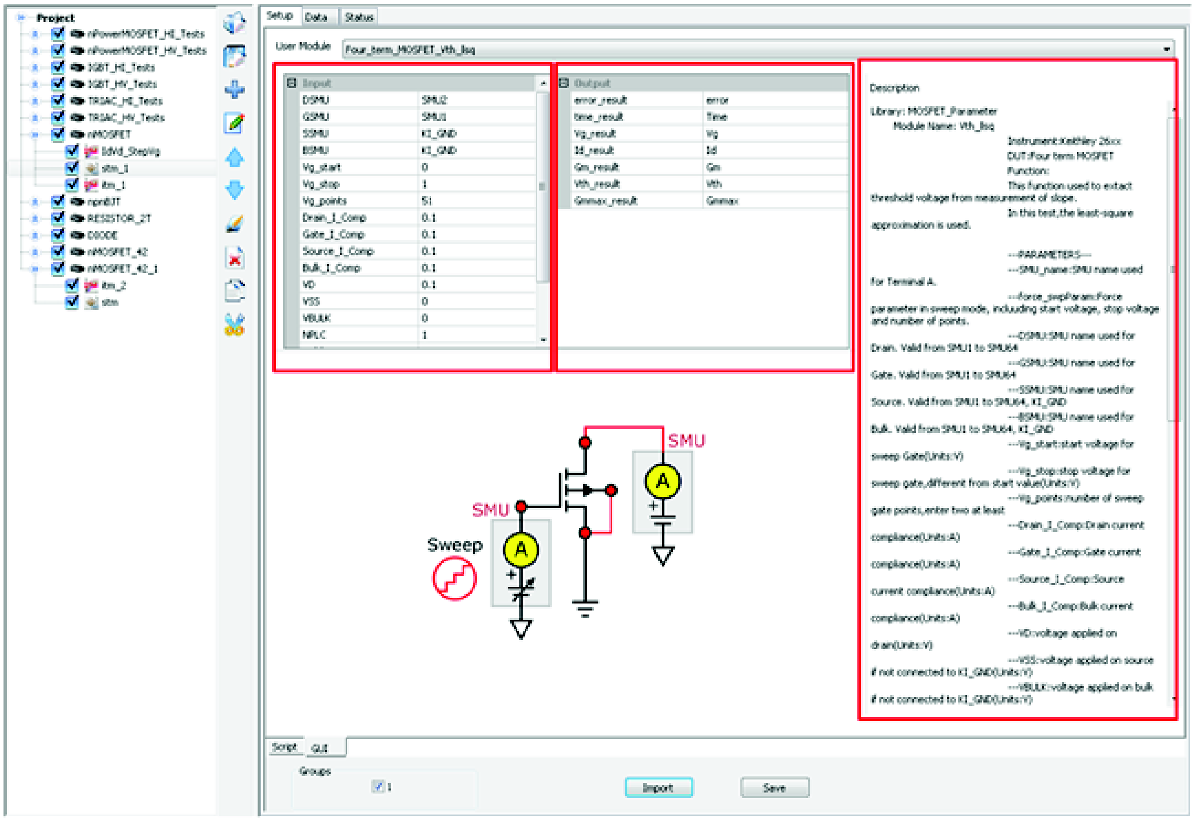
The GUI

STMs can include either the standard GUI or a custom built XRC GUI. Both GUIs provide an interface to change parameters in the script without editing the script itself. If a GUI is associated with your STM, you will be abel to view both the GUI and the script in the STM work area (see Setup tab for more information).

Standard GUI

A standard GUI can be created using the Script Editor tool. In the standard GUI, you can input the parameters directly without editing the script itself. There are three areas in the standard GUI Setup tab: Input, Output, and Description (see next figure).

Figure 144: Standard STM GUI



User Module: Four_term_MOSFET_Vth_lsq

Input	
DSMU	SMU2
GSMU	SMU1
SSMU	KI_GND
BSMU	KI_GND
Vg_start	0
Vg_stop	1
Vg_points	51
Drain_I_Comp	0.1
Gate_I_Comp	0.1
Source_I_Comp	0.1
Bulk_I_Comp	0.1
VD	0.1
VSS	0
VBULK	0
NPLC	1

Output	
error_result	error
time_result	Time
Vg_result	Vg
Id_result	Id
Gm_result	Gm
Vth_result	Vth
Gmmax_result	Gmmax

Description

Library: MOSFET_Parameter
Module Name: Vth_lsq
Instrument:Keithley 26xx
DUT:Four term MOSFET
Function:
This function used to extact threshold voltage from measurement of slope.
In this test,the least-square approximation is used.

---PARAMETERS---
---SMU_name:SMU name used for Terminal A.
---force_swpParam:Force parameter in sweep mode, incluuding start voltage, stop voltage and number of points.
---DSMU:SMU name used for Drain. Valid from SMU1 to SMU64
---GSMU:SMU name used for Gate. Valid from SMU1 to SMU64
---SSMU:SMU name used for Source. Valid from SMU1 to SMU64, KI_GND
---BSMU:SMU name used for Bulk. Valid from SMU1 to SMU64, KI_GND
---Vg_start:start voltage for sweep Gate(Units:V)
---Vg_stop:stop voltage for sweep gate,different from start value(Units:V)
---Vg_points:number of sweep gate points,enter two at least
---Drain_I_Comp:Drain current compliance(Units:A)
---Gate_I_Comp:Gate current compliance(Units:A)
---Source_I_Comp:Source current compliance(Units:A)
---Bulk_I_Comp:Bulk current compliance(Units:A)
---VD:voltage applied on drain(Units:V)
---VSS:voltage applied on source if not connected to KI_GND(Units:V)
---VBULK:voltage applied on bulk if not connected to KI_GND(Units:V)

XRC GUI

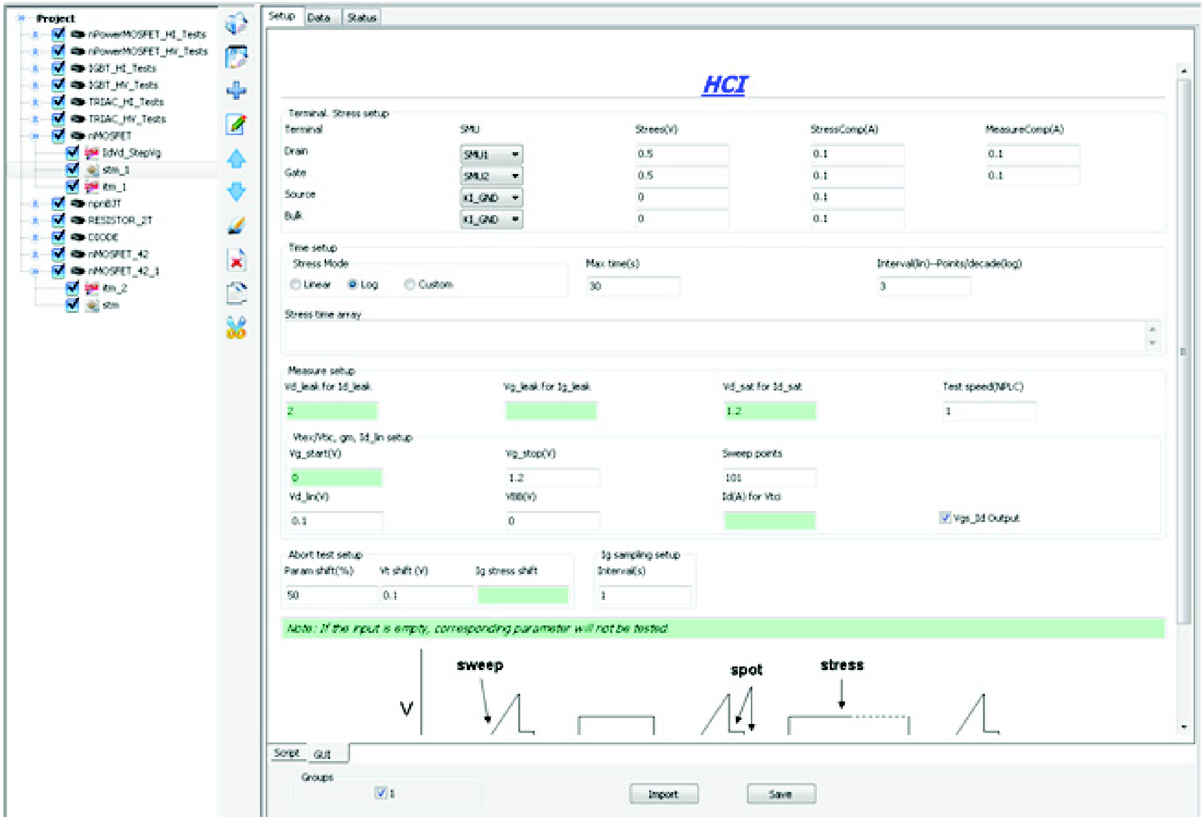
XRC GUIs are created using the XRCed tool and are saved as a file with the .xrc extension. If the imported TSP file has a corresponding .xrc GUI file, ACS Basic automatically loads and opens the GUI. You can set parameters on the GUI and you do not need to edit the parameters within the script itself.

XRCed can be accessed by clicking the ACS Basic Tools menu, then the Custom Test GUI designer command. For more information on how to generate the .xrc file, refer to Create a XRC GUI file topic.

An XRC GUI is completely customizable, and can include windows, menus, and a variety of input controls. The HCI STM in the next figure is one example of a STM that includes a XRC GUI.

You will have to include .xrc file in the header of the script to include it in the STM.

Figure 145: HCI module opened in STM work area



Terminal. Stress setup

Terminal	SMU	Stress(V)	StressComp(A)	MeasureComp(A)
Drain	SMU1	0.5	0.1	0.1
Gate	SMU2	0.5	0.1	0.1
Source	K1_GND	0	0.1	
Bulk	K1_GND	0	0.1	

Time setup

Stress Mode: ○ Linear ● Log ○ Custom

Max time(s): 30

Interval(lin)--Points/decade(log): 3

Stress time array:

Measure setup

Vd_leak for Id_leak: 2

Vg_leak for Ig_leak:

Vd_sat for Id_sat: 1.2

Test speed(NPLC): 1

Vtex/Vtic, gm, Id_lin setup

Vg_start(V): 0

Vg_stop(V): 1.2

Sweep points: 101

Vd_lin(V): 0.1

VBB(V): 0

Id(A) for Vtci:

☑ Vgs_Id Output

Abort test setup

Param shift(%): 50

Vt shift (V): 0.1

Ig stress shift:

Ig sampling setup

Interval(s): 1

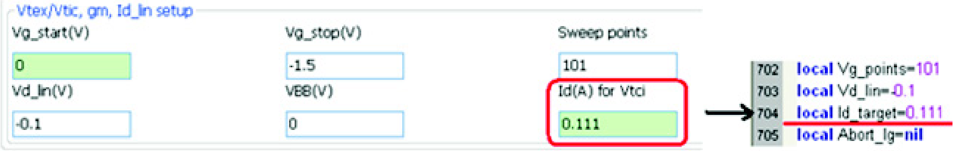
Note: If the input is empty, corresponding parameter will not be tested.

Groups: ☑ 1

All files in the WLR folder (the path is \\ACS\library\26Library\WLR) have a XRC GUI. The GUI file is installed in the XRC folder: \\ACS\library\26Library\xrc.

If an input parameter in the GUI is left empty, the value of this input parameter will show as "nil" in the function call of the script. After the correct input is entered, the function call area of the Setup tab of this test module will automatically update (see next figure).

Figure 146: Change of function call



Vtex/Vtic, gm, Id_lin setup

Vg_start(V): 0

Vg_stop(V): -1.5

Sweep points: 101

Vd_lin(V): -0.1

VBB(V): 0

Id(A) for Vtci: 0.111

```
local Vg_points=101
local Vd_lin=-0.1
local Id_target=0.111
local Abort_Ig=nil
```

Open a STM work area

The STM work area allows you to enter information that defines a given STM. The work area also allows you to view and analyze test data (both numerically and graphically) that is created by the STM, as well as check it's status. To open a blank STM work area in Multi-test mode:

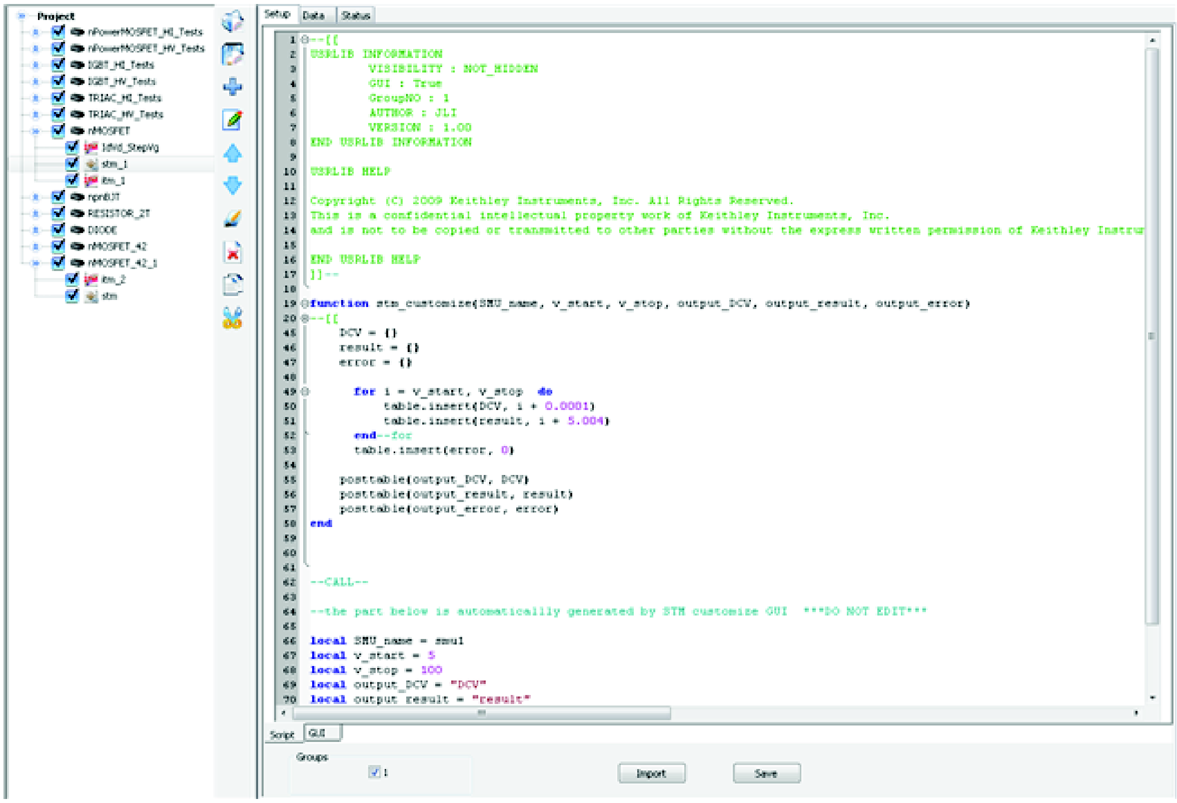
1. Select the device in the configuration navigator where you want to create the STM.
2. Click the **Add new test** icon ✚.
3. Select **STM** in the Specify the Test Type dialog box. Click **OK**. A new test named "stm" should be added to your device.
4. If you would like to run or modify an existing TSP file, Click the Import button located at the bottom of the STM work area in the setup tab. Refer to the ACS Basic Libraries Reference Manual for more information.

You can also access existing STMs from the test selection area of Multi-test or Single-test mode. Most of the available STMs in this area have a XRC GUI and are in the wafer level reliability (WLR) library.

As with all test modules, make sure to check that all of your SMU connections match what is written in the script or displayed in the GUI before you run the module. If you have written your own script or modified and existing file, you can save the configuration by clicking the **Save** icon 💾 located on the toolbar or by clicking the **Save** button at the bottom of the screen.

The STM work area contains three tabs: Setup, Data, and Status (see next figure).

Figure 147: The Setup tab of a XRC STM



```
--[[
USRLIB INFORMATION
        VISIBILITY : NOT_HIDDEN
        GUI : True
        GroupNO : 1
        AUTHOR : JLI
        VERSION : 1.00
END USRLIB INFORMATION

USRLIB HELP

Copyright (C) 2009 Keithley Instruments, Inc. All Rights Reserved.
This is a confidential intellectual property work of Keithley Instruments, Inc.
and is not to be copied or transmitted to other parties without the express written permission of Keithley Instru

END USRLIB HELP
]]--

function stm_customize(SMU_name, v_start, v_stop, output_DCV, output_result, output_error)
--[[
    DCV = {}
    result = {}
    error = {}

      for i = v_start, v_stop  do
          table.insert(DCV, i + 0.0001)
          table.insert(result, i + 5.004)
      end--for
      table.insert(error, 0)

    posttable(output_DCV, DCV)
    posttable(output_result, result)
    posttable(output_error, error)
end



--CALL--

--the part below is automaticallly generated by STM customize GUI   ***DO NOT EDIT***

local SMU_name = smu1
local v_start = 5
local v_stop = 100
local output_DCV = "DCV"
local output result = "result"
```

☑ 1

STM Setup tab

This tab allows you to specify a TSP-created user library and user module, and allows you to specify a limited number of test parameters, as well as import and save the STM. To configure or reconfigure an STM, you need to configure the features of the Setup tab, which is the default tab of the STM work area.The STM Setup tab area contains the following:

TSP Script area: displays a blank page for new STMs. It Is the main work area for viewing and editing a script test module (STM).

Groups information area: allows you to select the group(s) where you would like the current STM to be assigned.

STM Data tab

The Data tab displays the STM results in the data worksheet in a spreadsheet format. This spreadsheet allows you to perform data analysis. It displays the same output information as the Setup tab. It also allows you to create and export graphs of the test and test-analysis results. It also provides for flexible plot-data selection, formatting, annotation, and numerical coordinate display (using precision cursors).

STM Status tab

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.

Create a XRC GUI file

XRC GUIs are created using the XRC editor tool (XRCed) and saved as .xrc files. The XRC GUI can only be used to assign input values in the script. It cannot be used to display any outputs of the script.

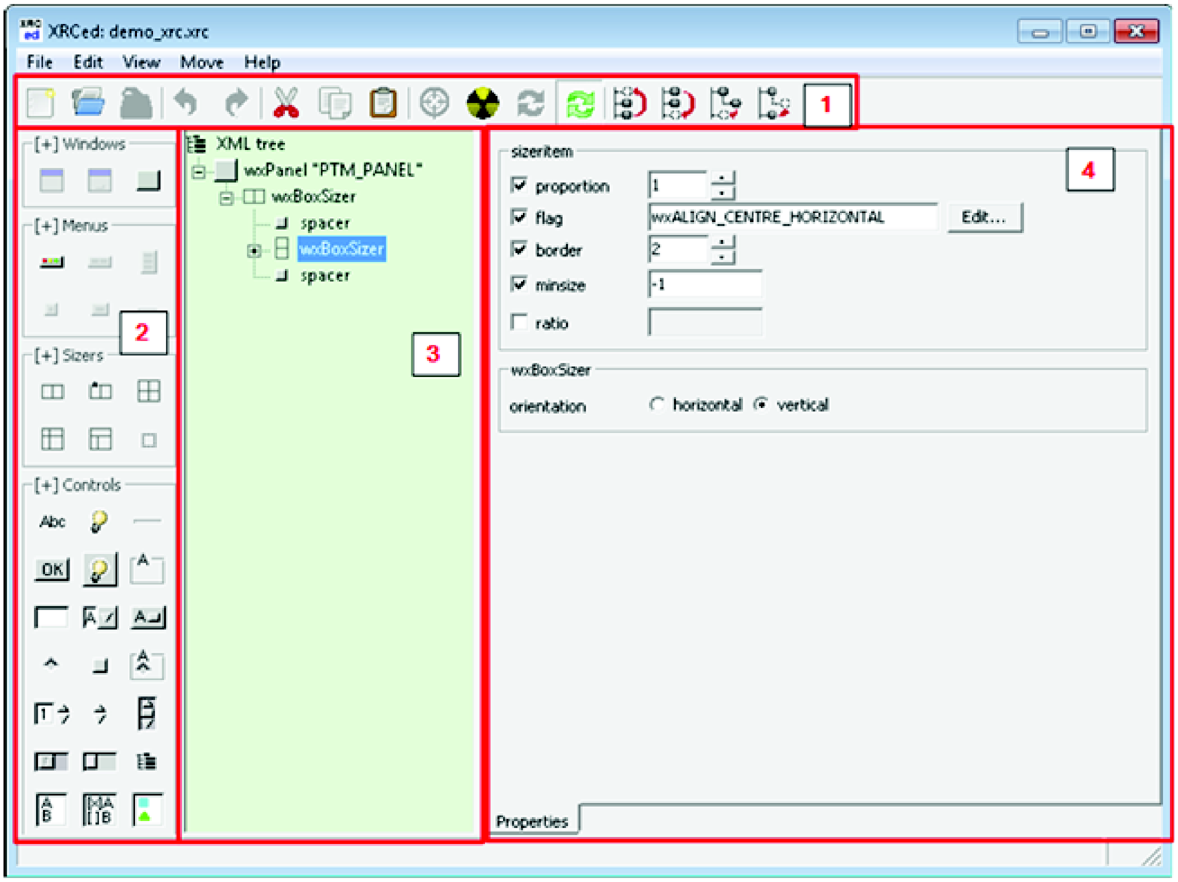
XRCed uses a tree format to create the GUI. As components are added, they must be placed as the child of a parent component within the XML tree, and their placement in the tree determines where they will be positioned in the GUI.

XRCed GUI

To start XRCed tool, click the ACS Basic Tools menu then select Custom Test GUI Designer (XRC). When the XRCed GUI opens, you will see four areas (see next figure):

1. Toolbar
2. Component selection area
3. XML tree
4. Work area

Figure 148: XRCed GUI



XML tree
- wxPanel "PTM_PANEL"
  - wxBoxSizer
    - spacer
    - wxBoxSizer
    - spacer

sizeritem

☑ proportion 1

☑ flag wxALIGN_CENTRE_HORIZONTAL Edit...

☑ border 2

☑ minsize -1

☐ ratio

wxBoxSizer

orientation ○ horizontal ● vertical

At the top of the XRCed, the toolbar allows you to execute several common commands as well as commands specific to the XRCed. The toolbar is grouped into four different sets of commands:

1. New, Open, and Save file
2. Undo and Redo
3. Cut, Copy, Paste
4. Locate, Test, Refresh, and Auto-refresh
   - **Locate**: finds a control in the XML tree. Click **Locate**, then click the control in the test window to view the control properties.
   - **Test**: creates a test window.
   - **Refresh**: updates the test window to show any changes you have made to the GUI. This button is only available when a test window is open.
   - **Auto-refresh**: updates the test window to show any changes whenever a new component is selected in the XML tree, if enabled.
5. Move up, Move down, Make sibling, Make child
   - **Move up**: moves the selected component to the next position in the tree.
   - **Move down**: moves the selected component to the position in the tree.
   - **Make sibling**: moves the selected component up to the next level in the tree hierarchy.
   - **Make child**: moves the selected component down to the next level in the tree hierarchy.

On the left hand side of the XRCed GUI is the the component selection area. These components are grouped into Windows, Menus, Sizers, and Controls:

- **Windows**: the largest component of the GUI; other components are grouped under windows:
  - **wxFrame** or **wxDialog** windows can only be created within the XML tree.
  - **wxPanel** is the only window that can be created within another component.
- **Menus**: create an area to group together controls.
- **Sizers**: help you organize the components in your window.
- **Controls**: make up the main content of the GUI. In addition to creating data input fields and descriptions for them, you can add more interactive ways to display the content.

The center column of the XRCed displays the XML tree. This is where you can navigate through different components of the GUI. To add a component to the tree, simply click the desired component in the component selection area. It will appear in the tree as a child to the component that was previously selected in the XML tree. Double click the component in the tree to change it's configuration.

The work area in the right column allows you to modify the components of the GUI. All components have a Properties tab in the work area that allows you to change settings specific to the component. Window and Control type components also have a Style tab that lets you change properties such as the font and layering settings.

Create a new XRC

The following instructions will give you a step by step guide on how to create an XRC GUI. For example, consider the following function within a script:

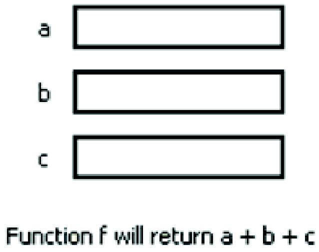
```
function f (a, b, c, output)
   output = a + b + c
   postdata ("output", output)
end
```

To create an XRC GUI for this function, use the codes that are in a TSP file. The TSP and XRC GUI file are linked by the .xrc file name that appears at the top of the TSP file, for example:

```
----<<xrc=example.xrc>>----
```

Before you start building your XRC GUI, you should have a draft of what you would like the GUI to look like. Consider what controls you would like to include, such as labels, static text, text boxes, and images. This will make the process faster and more efficient. The next figure is a draft of what the example GUI will look like:

Figure 149: An example GUI image

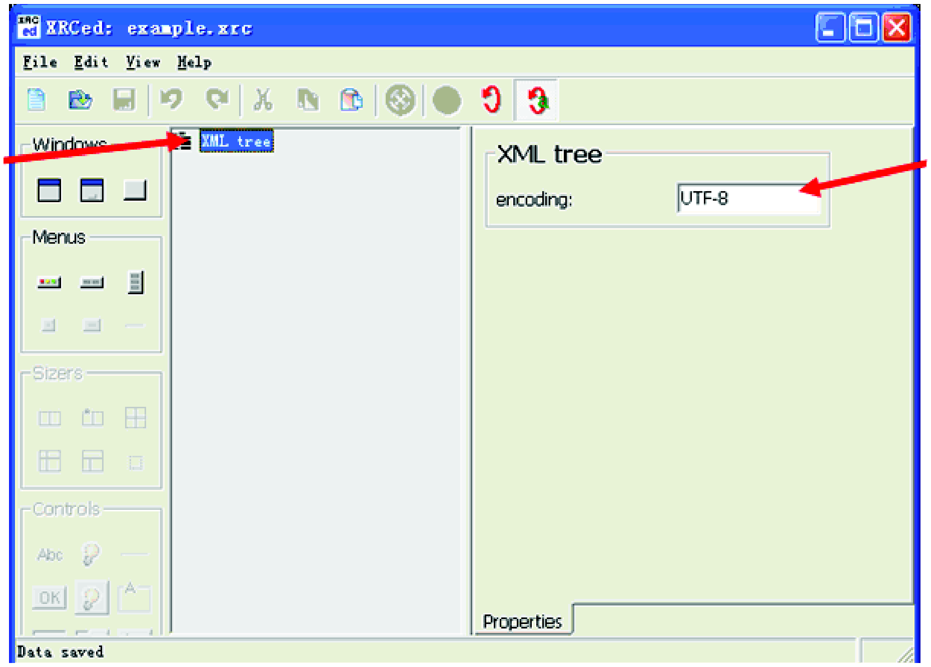


Once you have decided on a layout, open the XRCed program.

1. Click the ACS Basic **Tools** menu then select **Custom Test GUI Designer(XRC)**.
2. Click the **File** menu, then select **New** to open a new file.
3. Click the **XML tree** and change the encoding to UTF-8 in the encoding edit box (see next figure).

Click the **File** menu, then select **Save As** and save your .xrc file in the C:\ACS\library\26Library\xrc folder to set the directory path.

Figure 150: TSP GUI builder interface



XML tree

encoding: UTF-8

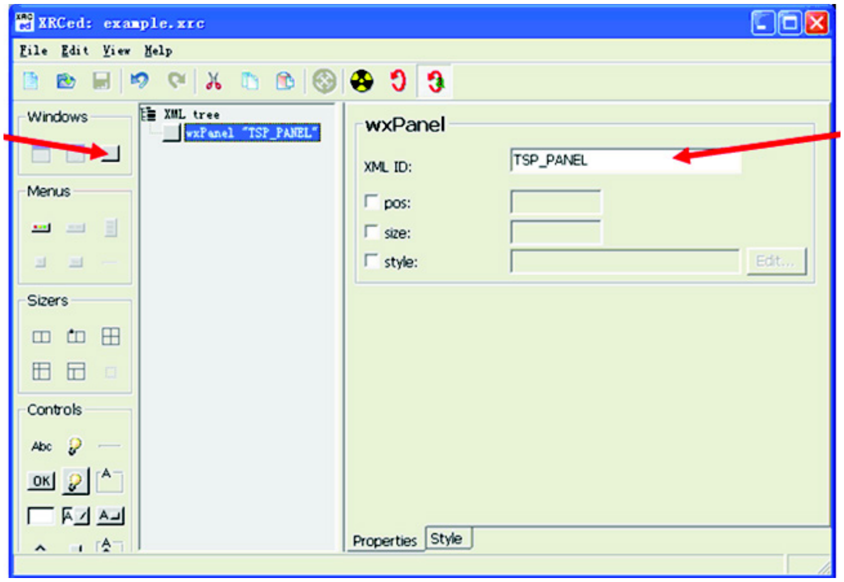
The first thing you can add to your GUI is a window, which is the outermost shell of the GUI. You must use a wxPanel window for the GUI to work in ACS Basic. For our example, add a panel under the XML tree:

1. Click the Add wxPanel icon on the left panel to add a panel under the XML tree (see next figure).
2. Select the wxPanel in the XML tree to bring up it’s workspace, then change the XML ID to "TSP_Panel" (see next figure).

NOTE

Naming the panel "TSP_PANEL" for the new object is essential for creating a GUI related to a TSP script. You should only use this name in the XML ID edit box. If you create an XRC GUI for a PTM, enter PTM_PANEL in the XML ID edit box (see the Configure an XRC PTM topic for more information).

Figure 151: Add panel for new object



XML ID: TSP_PANEL

□ pos:

□ size:

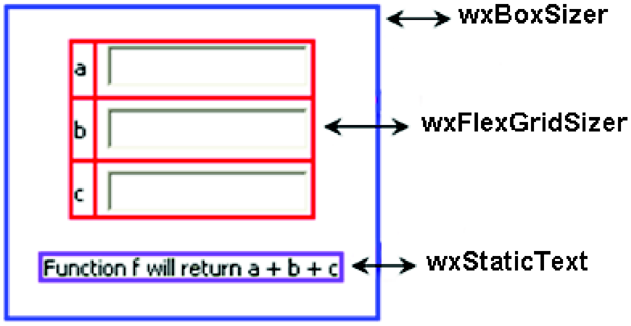
□ style:

After creating your panel, start adding Sizers to organize the layout of the GUI. Sizers generally do not create visuals that are seen in the GUI, they only create a structure to organize the components that they contain. Once you add a Sizer in the XML tree, you can then add components underneath it that will be arranged in the GUI based on the Sizers configuration. These components can either be Controls or other Sizers used to create further substructures. There are several types of sizers that can be added to an XRC GUI.

- **wxBoxSizer**: is the most basic Sizer. Components that are listed under a wxBoxSizer in the XML tree will be displayed in a horizontal row or vertical column, depending on the configuration.
- **wxStaticBoxSizer**: is similar to the wxBoxSizer, but it also creates a visual border around the entire Sizer in th GUI. You can also add text to embed in the upper right corner of the border.
- **wxGridSizer**: allows you to create multiple rows and columns. Components listed under this Sizer in the XML tree fill in the GUI grid from left to right, starting at the top row and proceeding to the one below after each row is full.
- **wxFlexGridSizer**: is similar to the wxGridSizer, but offers more flexibility in shaping the grid.
- **wxGridBagSizer**: is disabled in ACS Basic
- **wxSpacer**: can be used as a place holder to create a specifically shaped space in the GUI.

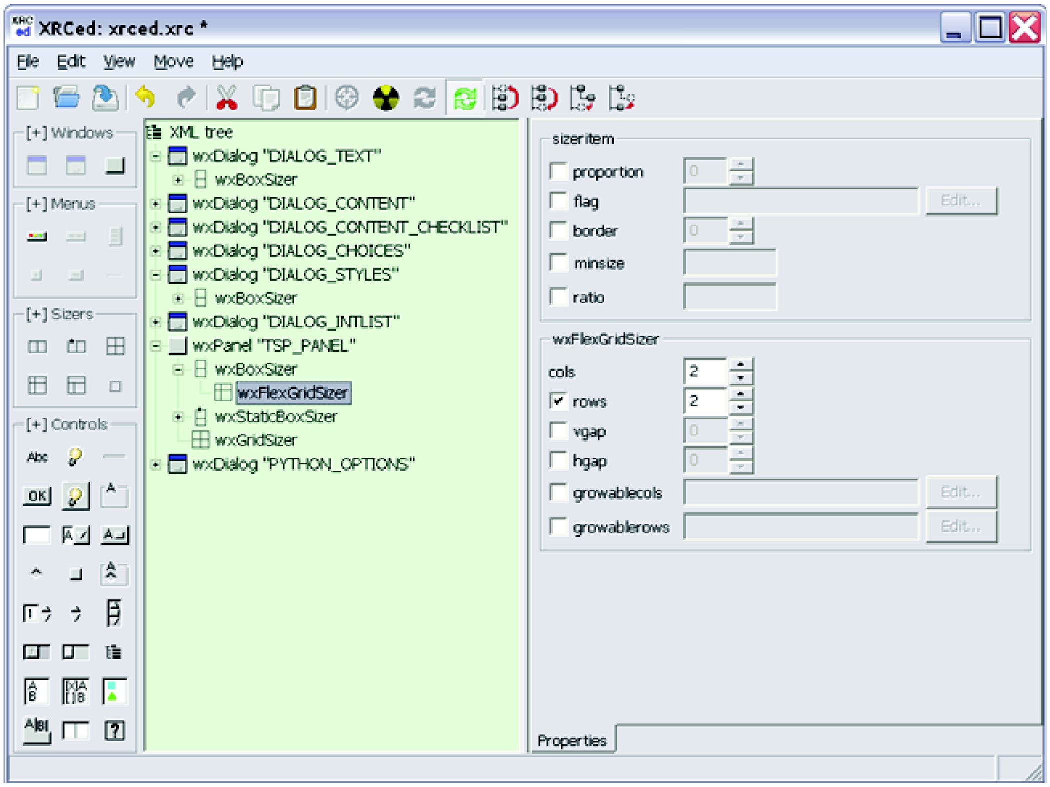
To continue our example, add Sizers to create the overall structure for the GUI (see next figure).

Figure 152: Design GUI structure



1. Click the **wxPanel** node of the XML tree.
2. Click the **wxBoxSizer** icon to add a wxBoxSizer under the wxPanel.
3. Select **wxBoxSizer** orientation: vertical or horizontal.
4. Double click the **wxBoxSizer** node to make sure it is selected in the XML tree. Click the **wxFlexGridSizer** icon to add a wxFlexGridSizer under wxBoxSizer.
5. Double click the **wxBoxSizer** node to make sure it is selected in the XML tree. Click the **wxStaticText** icon to add a wxStaticText under wxBoxSizer after the wxFlexGridSizer.

Figure 153: Add Sizers under the TSP_PANEL



XRCed: xrced.xrc *

XML tree
- wxDialog "DIALOG_TEXT"
  - wxBoxSizer
- wxDialog "DIALOG_CONTENT"
- wxDialog "DIALOG_CONTENT_CHECKLIST"
- wxDialog "DIALOG_CHOICES"
- wxDialog "DIALOG_STYLES"
  - wxBoxSizer
- wxDialog "DIALOG_INTLIST"
- wxPanel "TSP_PANEL"
  - wxBoxSizer
    - wxFlexGridSizer
  - wxStaticBoxSizer
  - wxGridSizer
- wxDialog "PYTHON_OPTIONS"

sizeritem

□ proportion	0
□ flag	Edit...
□ border	0
□ minsize	
□ ratio	

wxFlexGridSizer

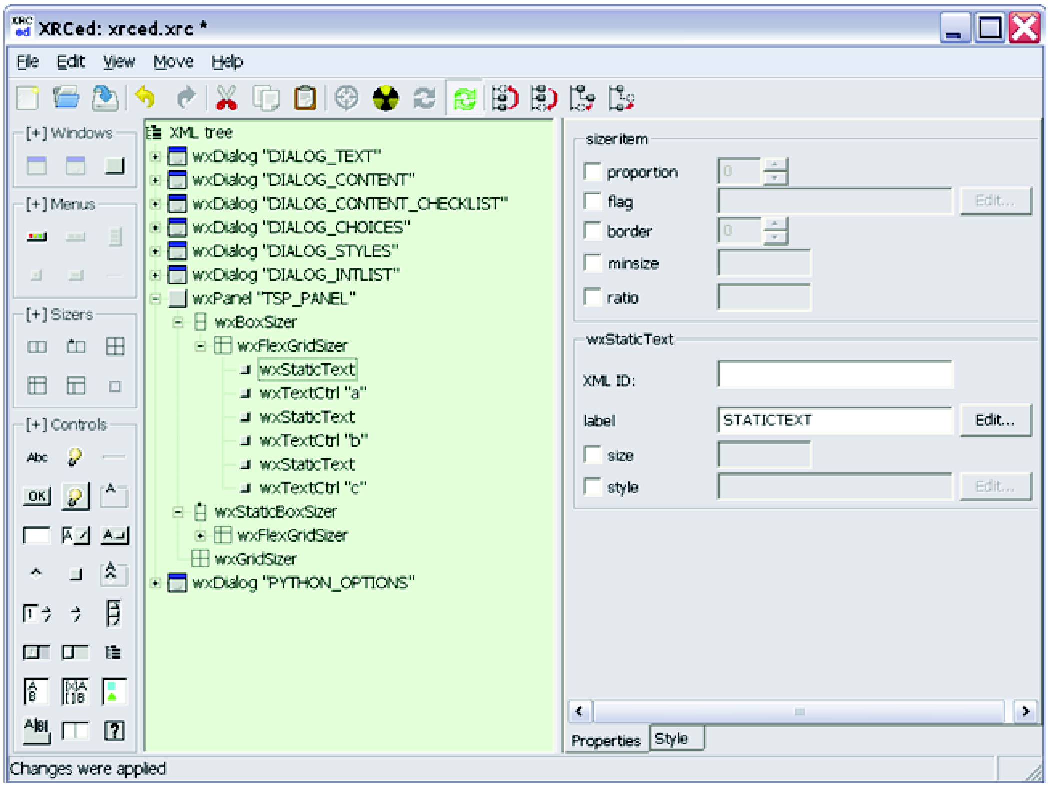
cols	2
☑ rows	2
□ vgap	0
□ hgap	0
□ growablecols	Edit...
□ growablerows	Edit...

Note that the example won't look like much; the only visible component is the wxStaticText. After more components are added, the structure created by the Sizers will become visible.

Once the structure is complete, add Controls to fill in the GUI. Add labels and text box under the wxFlexGridSizer, and then modify the information for each:

1. Click on the **wxFlexGridSizer** node of the XML tree.
2. Set cols (columns) and rows: item on the right of interface to match the number of rows in the GUI. For example, in the previous figure, the rows should be changed to three in order to match the GUI draft.
3. Click the **wxStaticText** icon and the **wxTextCtrl** icon in the Controls box to add a **wxStaticText** (label) and a **wxTextCtrl** (text box) under the wxFlexGridsizer. Repeat this step two times to complete all three rows.
4. Click on each **wxStaticText** component in the **wx-FlexGridSizer** and set the label in the work area. In this example, set the label edit box to a, b, and c respectively (see next figure).
5. Click on each **wxTextCtrl** component in the **wx-FlexGridSizer** and set the XML ID. For example, set the XML ID edit box to a, b, and c respectively. This step is important because it links the data input in these boxes to be used as variables in the script.
6. Click the **wxStaticText** component in the **wxBoxSizer** and set the label at the bottom of the GUI. For our example, enter the text "Function f will return a + b + c."

Figure 154: Modify information



```
XML tree
  wxDialog "DIALOG_TEXT"
  wxDialog "DIALOG_CONTENT"
  wxDialog "DIALOG_CONTENT_CHECKLIST"
  wxDialog "DIALOG_CHOICES"
  wxDialog "DIALOG_STYLES"
  wxDialog "DIALOG_INTLIST"
  wxPanel "TSP_PANEL"
    wxBoxSizer
      wxFlexGridSizer
        wxStaticText
        wxTextCtrl "a"
        wxStaticText
        wxTextCtrl "b"
        wxStaticText
        wxTextCtrl "c"
    wxStaticBoxSizer
      wxFlexGridSizer
    wxGridSizer
  wxDialog "PYTHON_OPTIONS"
```

sizeritem

- □ proportion 0
- □ flag
- □ border 0
- □ minsize
- □ ratio

wxStaticText

- XML ID:
- label: STATICTEXT
- □ size
- □ style

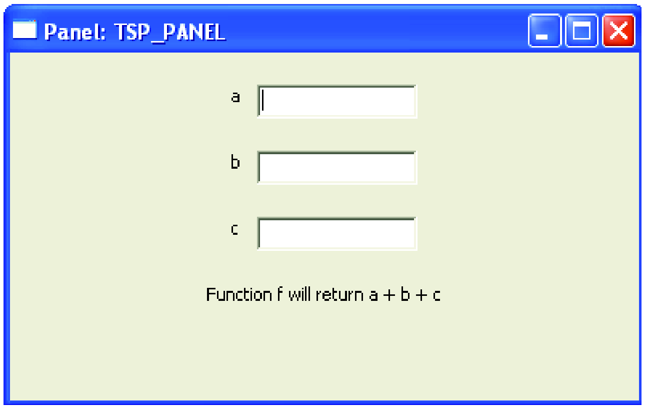
Changes were applied

Now all of the content has been added to your GUI. The last thing to do is add formatting to align and space out the components. Most of the formatting will be done in the "sizeritem" area of the work area for the wxFlexGridSizer and the wxStaticText at the end of the GUI. Since we are using a wxFlexGridSizer, the labels and text boxes in the grid will align nicely and do not need any extra formatting.

1. Click the **wxFlexGridSizer**. Check the border option in the sizeritem area of the work area and increase the value to 20. This border will add space around the grid so that it does not push against the edge of the window.
2. Check the flag option for the wxFlexGridSizer.
    a. Click the **Edit…** button to the right, which will bring up the Sizer item flags dialog box.
    b. Select **wxTOP** and **wxBOTTOM**. These will add the border space to the top and bottom edge of the grid. Alternately, you could select wxLEFT, wxRIGHT, or wxALL for other border configurations.
    c. Select **wxALIGN_CENTER_HORIZONTAL**. This will align the grid to the horizontal center of the window.
    d. Click **OK**.
3. In the **wxFlexGridSizer** work area, select the **vgap** (vertical gap) and **hgap** (horizontal gap) options. Increase the vertical gap to 20 and the horizontal gap to 10. This will add space between the cells of the column so that it is not so tightly packed.
4. Click the last sibling **wxStaticText** in the GUI. Select the flag option in the sizeritem area of the work area. Click the **Edit…** button, and select **wxALIGN_CENTER_HORIZONTAL** then click **OK**.

Your GUI is now complete, and should look like the next figure. There are many more options available to adjust formatting if you choose to do so. Every component will have a sizeritem area in it's work area, as well as settings specific to the type of component. Also, many components have a style tab in the work area where you can adjust settings like the font. Once you have finalized your GUI, make sure to click **Save** or **Save As** in the File menu to save the .xrc file.

Figure 155: Example GUI

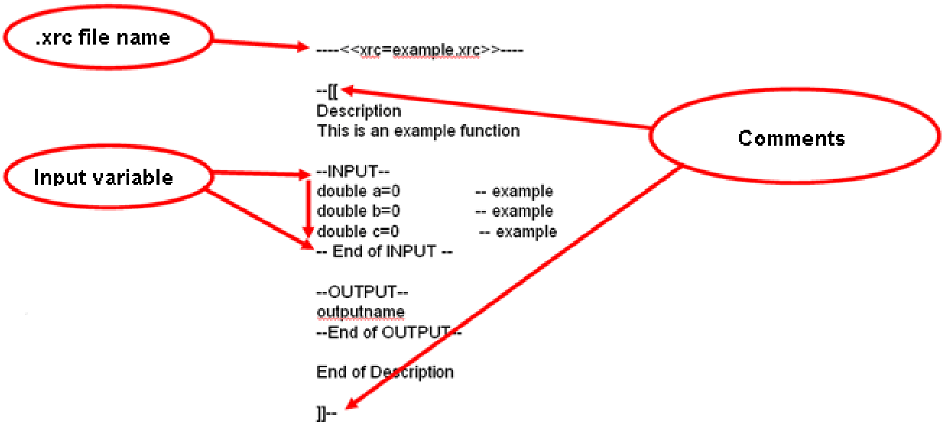


Function f will return a + b + c

Modify script for an imported GUI

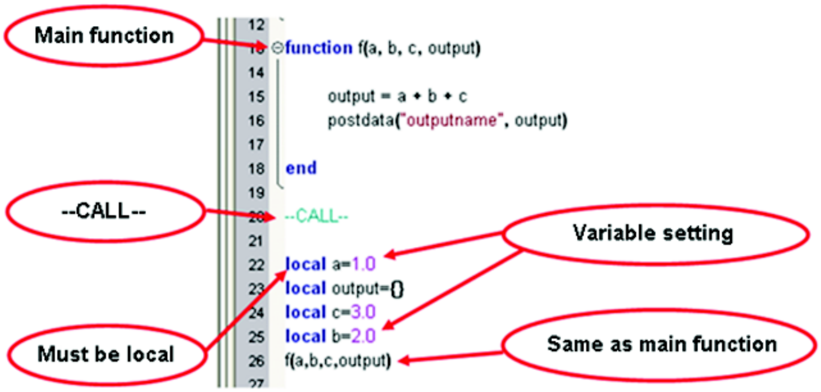
If you want to modify information for an imported GUI, you must modify the script. In the script, there must be four parts: .xrc file name, input variables, function, and function call. You can modify different parts depending on your requirements (see next two figures).

Figure 156: Modify Script for import GUI



```
----<<xrc=example.xrc>>----

--[[
Description
This is an example function

--INPUT--
double a=0          -- example
double b=0          -- example
double c=0          -- example
-- End of INPUT --

--OUTPUT--
outputname
--End of OUTPUT--

End of Description

]]--
```

Figure 157: Modify Script for import GUI_2



```
function f(a, b, c, output)

    output = a + b + c
    postdata("outputname", output)

end

--CALL--

local a=1.0
local output={}
local c=3.0
local b=2.0
f(a,b,c,output)
```

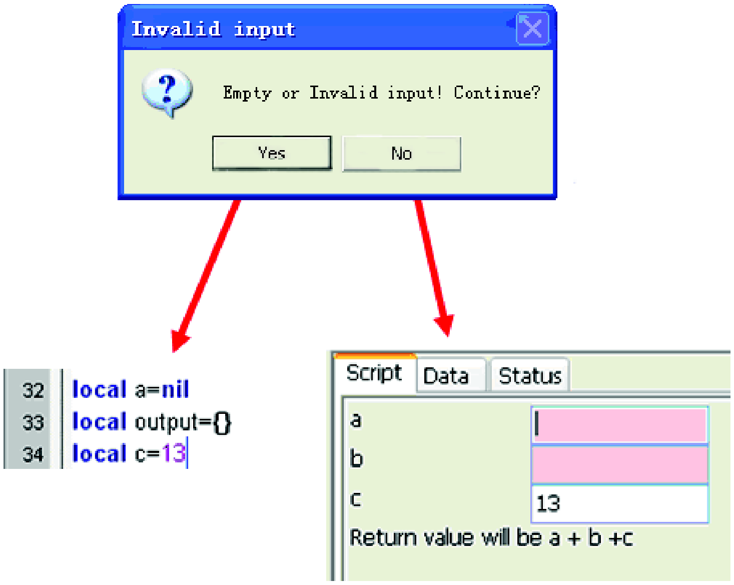
NOTE

In a TSP script, comments are preceded by "- -" on a single line or enclosed in "- - [[" and "]] - -" for a multi-line comment block. In a C++ program, comments are enclosed in "/*" and "*/". Input variable types can be integer, double, string, table, or instID. The function call area is proceeded by the text '--CALL-', and the function name must be the same as the main function's. The following '--CALL --' is auto-generated (see previous figure).

For input variables, you can limit the input value range. For example: double a=0.0 in [-40,40]; double b=0.0 in [-40,]; integer a=0 in [,40].

In the GUI, you can input values of your choice. If an input is left empty, the Invalid input dialog box will open when you click on the Script tab in the work area (see next figure). Clicking **Yes** sets the variable to "nil" in the script. Clicking **No** will change the invalid or empty input box to a different color.

Figure 158: Invalid input dialog box



Empty or Invalid input! Continue?

```
32  local a=nil
33  local output={}
34  local c=13
```

a |

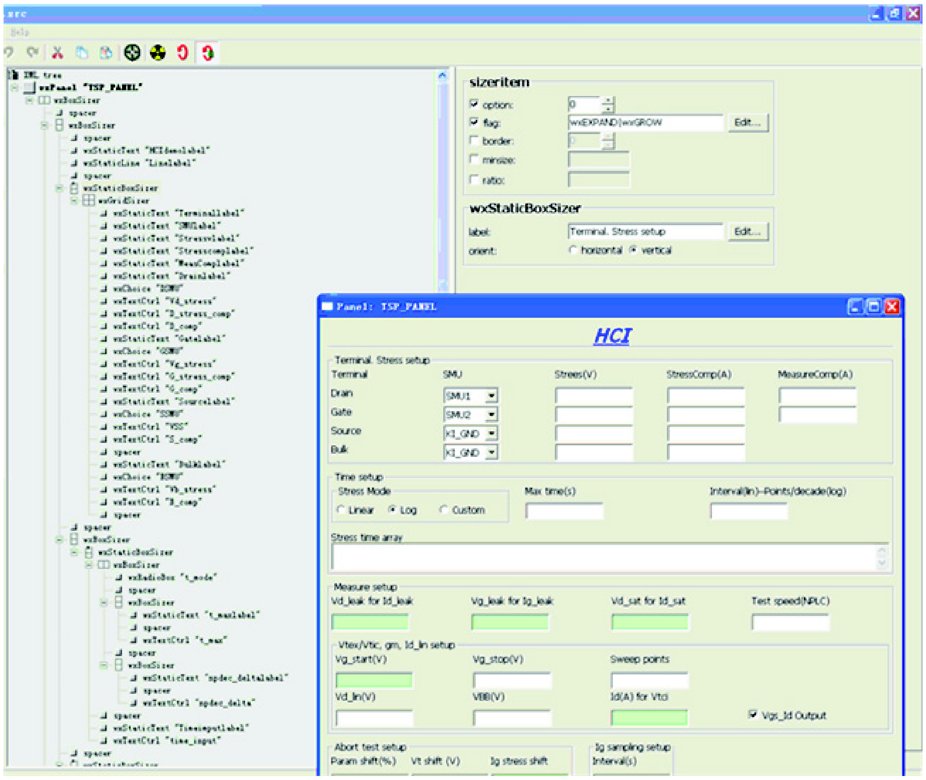
b

c 13

Return value will be a + b +c

The next figure shows the XRC GUI of the HCI and the XRC XML tree.

Figure 159: HCI .xrc GUI and it's XML tree



```
XML tree
wxPanel "TSP_PANEL"
  wxBoxSizer
    spacer
    wxBoxSizer
      spacer
      wxStaticText "HCIdemolabel"
      wxStaticLine "Linelabel"
      spacer
      wxStaticBoxSizer
        wxGridSizer
          wxStaticText "Terminallabel"
          wxStaticText "SMUlabel"
          wxStaticText "Stresslabel"
          wxStaticText "Stresscomplabel"
          wxStaticText "MeasComplabel"
          wxStaticText "Drainlabel"
          wxChoice "DSMU"
          wxTextCtrl "Vd_stress"
          wxTextCtrl "D_stress_comp"
          wxTextCtrl "D_comp"
          wxStaticText "Gatelabel"
          wxChoice "GSMU"
          wxTextCtrl "Vg_stress"
          wxTextCtrl "G_stress_comp"
          wxTextCtrl "G_comp"
          wxStaticText "Sourcelabel"
          wxChoice "SSMU"
          wxTextCtrl "VSS"
          wxTextCtrl "S_comp"
          spacer
          wxStaticText "Bulklabel"
          wxChoice "BSMU"
          wxTextCtrl "Vb_stress"
          wxTextCtrl "B_comp"
          spacer
      spacer
      wxBoxSizer
        wxStaticBoxSizer
          wxBoxSizer
            wxRadioBox "t_mode"
            spacer
            wxBoxSizer
              wxStaticText "t_maxlabel"
              spacer
              wxTextCtrl "t_max"
            spacer
            wxBoxSizer
              wxStaticText "npdec_deltalabel"
              spacer
              wxTextCtrl "npdec_delta"
          spacer
          wxStaticText "Timeinputlabel"
          wxTextCtrl "time_input"
      spacer
      wxStaticBoxSizer
```

sizeritem

- ☑ option: 0
- ☑ flag: wxEXPAND|wxGROW
- □ border: 0
- □ minsize:
- □ ratio:

wxStaticBoxSizer

label: Terminal. Stress setup

orient: ○ horizontal ● vertical

HCI

Terminal. Stress setup

Terminal	SMU	Stress(V)	StressComp(A)	MeasureComp(A)
Drain	SMU1			
Gate	SMU2			
Source	K1_GND			
Bulk	K1_GND			

Time setup

Stress Mode: ○ Linear ● Log ○ Custom

Max time(s)

Interval(lin)--Points/decade(log)

Stress time array

Measure setup

Vd_leak for Id_leak Vg_leak for Ig_leak Vd_sat for Id_sat Test speed(NPLC)

Vtex/Vtic, gm, Id_lin setup

Vg_start(V) Vg_stop(V) Sweep points

Vd_lin(V) VBB(V) Id(A) for Vtci ☑ Vgs_Id Output

Abort test setup

Param shift(%) Vt shift (V) Ig stress shift

Ig sampling setup

Interval(s)

Key points for creating TSP GUI

The XML ID for input components in the GUI must have the same name as the parameters in the script.

The XML ID of the wxPanel must be **TSP_PANEL** (For a PTM XRC GUI, the XML ID must be PTM_PANEL).

There must be a **--CALL--** area in the script.

The **--INPUT--** area and the definitions of the input variables should be clear.

Input a range for variables.

There must be a function call. The call function name has to be the same as the main function.

Configure a python test module (PTM)

A python language test module (PTM) is a module defined primarily by programming a python language user module. It can be created by using the script editor in ACS Basic.

ACS Basic supports three types of PTM: script, standard, and .xrc. All three types utilize a work area containing the same three tabs: Setup, Data, and Status. However, the Setup tab contains differences depending on the type of PTM.

Setup tab: allows you to specify a user library and user module, as well as a limited number of test parameters. To configure or reconfigure a PTM, you need to change the settings of the Setup tab, which is the default tab of the PTM work area.

Data tab: displays the test results in a spreadsheet and allows you to perform data analysis. The Data tab also allows you to create and export graphs of the test and test-analysis results. Graph options include flexible plot-data selection, formatting, annotation, and numerical coordinates display using precision cursors.

Status tab: monitors the current configuration status of the PTM, reporting its readiness for use and recommending additional preparations, if necessary. A test-ready report for a PTM is the same as for an ITM. To view example Status tab reports for ITMs, refer to the ITM Status tab topic.

Configure a PTM script

A PTM script refers to a file or program that can be created using python. In ACS Basic, the script module shows a script format with no GUI.

For information about creating a script using the Script Editor, refer to the Script editor tutorials.

After a new PTM is inserted in a project plan, it must be configured to meet the test requirements. This section describes how to add a script PTM and how to work with python code.

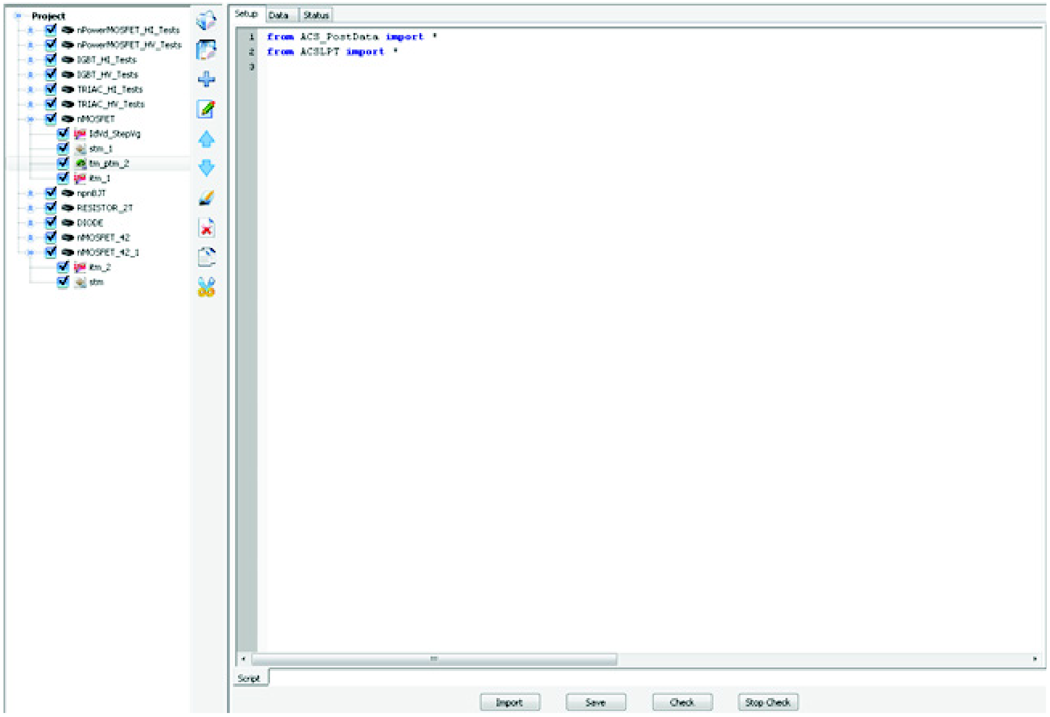
Insert a new PTM

1. Click the **Add new test** icon ✚, and then click **PTM** in the Specify the Test Type dialog box.

The PTM **Setup** tab is the default view and it contains the Python Script work area that is a blank module and allows you to import one into the user library or into your own library. It also contains buttons at the bottom of the GUI (see next figure).

- **Import button**: imports a PTM. You can import a script, standard, or a .xrc PTM.
- **Save button**: saves the PTM.
- **Check button**: checks the python code for basic errors by checking it in the interpreter (only available in Demo mode).
- **Stop Check button**: ends the code checking action (only available in Demo mode).

Figure 160: A blank PTM



```
from ACS_PostData import *
from ACSLPT import *
```

1. Import a PTM, or enter python program code in the Setup tab. If you choose to import, click the **Import** button. Scripts located in the pyLibrary are described in the Device library section.
2. Verify that the SMU settings match the physical connections; change inputs, if necessary.
3. If needed, configure or change the script. Then click the **Check** button to check the script for errors.
4. If needed, rename this module by clicking the Rename icon on the vertical toolbar.
5. Click the **Save** icon located on the toolbar (or you can select the **Save** button at the bottom of the GUI).

A python script

A script is a collection of instrument control commands and programming statements. PTM scripts are written as programming code in python, in a structure that is framed by commands. Program statements lie within these commands and control script execution, providing facilities such as variables, functions, branching, and loop control.

The table in the next topic explains Python code, displays the lines of code, and corresponding information.

For additional information regarding python code, go to the following websites:

- http://www.python.org/download/
- http://www.activestate.com/Products/ActivePython/
- http://www.python.org/doc/

PTM Script example

Below is an example script used to perform a diode measurement:

```
import ACSLPT as lpt
import ACS_PostData as post
def Diode_DynamicZ_I1I2(PSMU, NSMU, I1, I2, RangeV, ComplianceV, nPLC, Holdtime):
    """
    VISIBILITY: NOT_HIDDEN
    TYPE: script
    INPUTLIST
        #Name,  Type,  Default Value,  Min Value, Max Value
        PSMU,enum,'SMU2',('SMU1','SMU2','SMU3','SMU4')
        NSMU,enum,'SMU1',('SMU1','SMU2','SMU3','SMU4','GNDU')
        I1, float,  0.01,  0.1,0.1
        I2, float,  0.02,  0.1,0.1
        RangeV, int,  1,  1,5
        ComplianceV,  float,  20,  200,200
        nPLC,   float,  1.0,   0.01,10
        Holdtime, float,  0.01,  0,100
    END INPUTLIST
    OUTPUTLIST
        #Name,       Type
        error,       int
        DynamicZ,    float
    END OUTPUTLIST

    DESCRIPTION
        Library: S4200Parlib
        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.

        *********
        Input
        *********
        PSMU(enum):     SMU name of P terminal. Valid from SMU1 to SMU8.
        NSMU(enum):     SMU name of N terminal. Valid from SMU1 to SMU8, GNDU.
        I1(float):      P terminal forced current. Valid from -0.1 to 0.1. [A]
        I2(float):      P terminal forced current. Valid from -0.1 to 0.1. [A]
        RangeV(int):          Range of voltage measurement. Valid from 1 to 5.
                                Input            Range
                               ======================
                               1           Full Auto
                               2           Limited Auto 0.2V
                               3           Limited Auto 2V
                               4           Limited Auto 20V
                               5           Limited Auto 200V

        ComplianceV(float):   Compliance of current force. Valid from -200 to 200.
   [V]
```

```
     nPLC (float):  Number of power line cycles for integration.  Valid input
from 0.01 to 10.
     Holdtime (float):     Holdtime before measurement. Valid input from 0 to
100.  [s]

     *********
     Output
     *********
     error(int):
                     0     (OK) normal working condition;
                     -1       Output file open error
                     -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.
     DynamicZ(float):             Dynamic impedance of diode.
     END DESCRIPTION
 """

 error = 0
 DynamicZ = 0.0
 lpt.tstse()

 #Some input checking is needed
 if  abs(I1) > 0.1 or DrainIbd == 0:
     error = -10100
     post.ACSPostDataInt("error",error)
     return error, DynamicZ
 if  abs(I2) > 0.1 or DrainIbd == 0:
     error = -10100
     post.ACSPostDataInt("error",error)
     return error, DynamicZ
 if  RangeV < 1 or RangeV > 5:
     error = -10100
     post.ACSPostDataInt("error",error)
     return error, DynamicZ
 if  abs(ComplianceV) > 200:
     error = -10100
     post.ACSPostDataInt("error",error)
     return error, DynamicZ
 if  nPLC < 0.01 or nPLC > 10:
     error = -10100
     post.ACSPostDataInt("error",error)
     return error, DynamicZ
 if  Holdtime < 0 or Holdtime > 100:
     error = -10100
     post.ACSPostDataInt("error",error)
     return error, DynamicZ

 #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
```

Python code explanation

Code lines	Element	Description
1 through 2	Import declaration	The keyword `Import` starts the import declaration.
3 through 4	Declaring functions	The keyword `def` starts the function declaration, followed by the function name and the arguments in parentheses. Multiple arguments are separated with commas.
5 through 75	Function's document string (help information)	The function's document string is enclosed within triple quotes. 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. White space is significant and must be consistent. In this example, the function code (including the `doc string`) is indented four spaces. It does not need to be four spaces; it needs to be consistent. The first line that is not indented is outside the function. Refer to the indentation error examples below this table
77 through 169	Functional code	The functional program code that runs the test.
170 through 176	Post-test data	The post-test data posting. This section sends variable data to the test module data tab.

The following example shows various 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
```

Post data function in PTM

The following functions are used for a PTM to post data to the data tab and plot of the PTM:

- 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 Basic, you must import ACS_PostData at the beginning of the python source code.

Configure a standard PTM

A standard PTM refers to a script with a standard PTM GUI that can be created by using the Script Editor tool of ACS Basic. A standard GUI allows you to enter organized parameters and values before running a test.

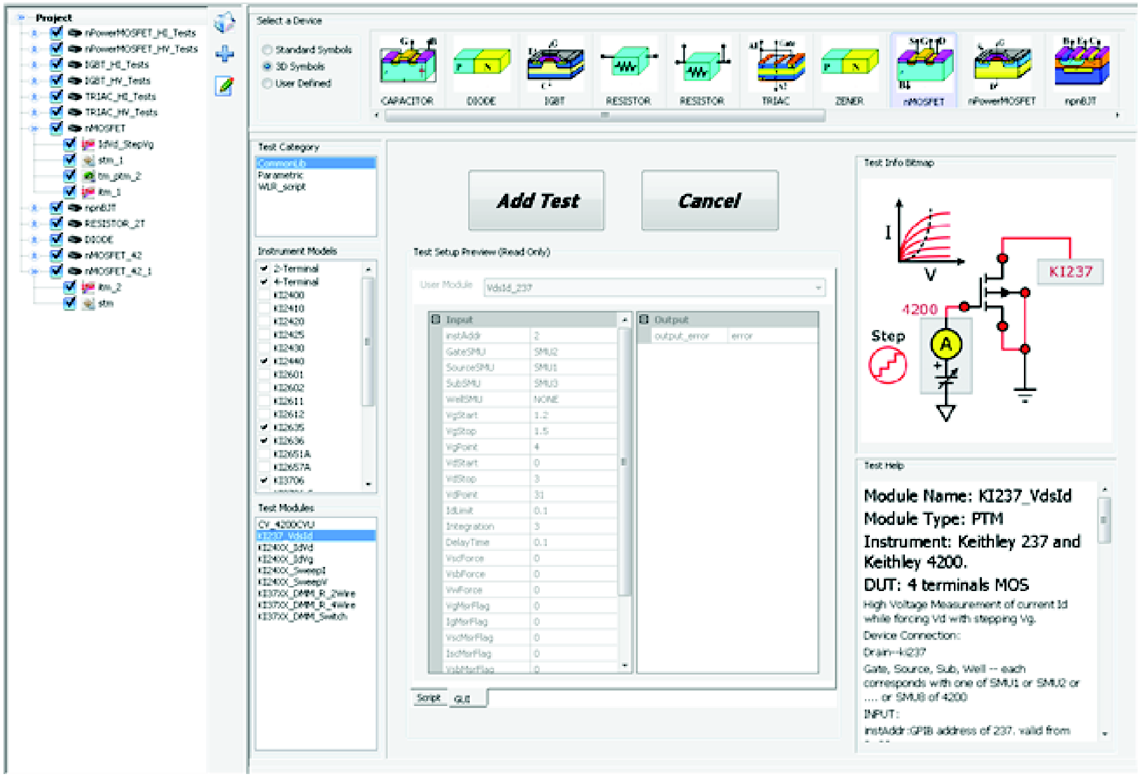
For more information on how to create a standard PTM using Script Editor, you can refer to **Tutorial 2: Create and run a new PTM.**

After a new standard PTM is inserted to a project plan, it must be configured to meet the test requirements. This section describes the configuration of standard PTMs.

Configure a standard PTM

Import and open a standard PTM. The PTM Setup tab GUI opens (see next two figures).

Figure 161: Import a standard PTM



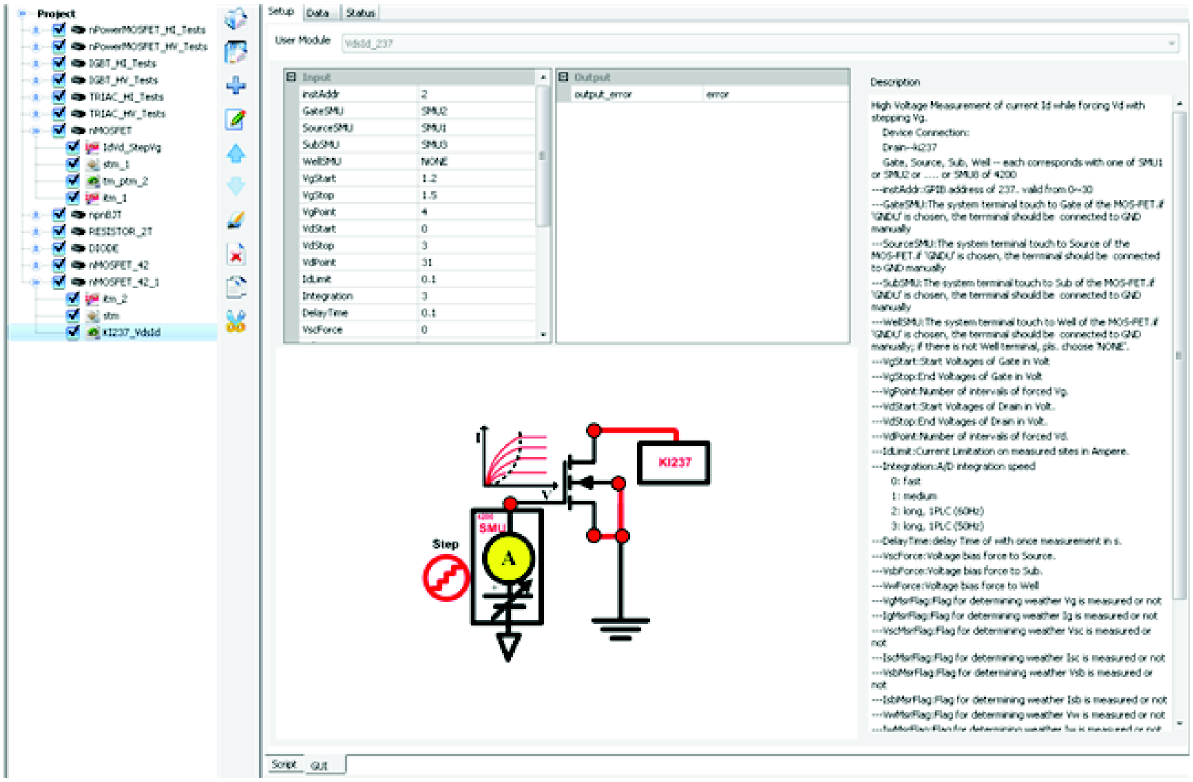
User Module: VdsId_237

Input	
instAddr	2
GateSMU	SMU2
SourceSMU	SMU1
SubSMU	SMU3
WellSMU	NONE
VgStart	1.2
VgStop	1.5
VgPoint	4
VdStart	0
VdStop	3
VdPoint	31
IdLimit	0.1
Integration	3
DelayTime	0.1
VsdForce	0
VsbForce	0
VwForce	0
VgMsrFlag	0
IgMsrFlag	0
VsdMsrFlag	0
IsdMsrFlag	0
VsbMsrFlag	0

Output	
output_error	error

Module Name: KI237_VdsId
Module Type: PTM
Instrument: Keithley 237 and Keithley 4200.
DUT: 4 terminals MOS
High Voltage Measurement of current Id while forcing Vd with stepping Vg.
Device Connection:
Drain--KI237
Gate, Source, Sub, Well -- each corresponds with one of SMU1 or SMU2 or .... or SMU8 of 4200
INPUT:
instAddr :GPIB address of 237, valid from

Figure 162: Open a standard PTM



User Module Vdsld_237

Input	
instAddr	2
GateSMU	SMU2
SourceSMU	SMU1
SubSMU	SMU3
WellSMU	NONE
VgStart	1.2
VgStop	1.5
VgPoint	4
VdStart	0
VdStop	3
VdPoint	31
IdLimit	0.1
Integration	3
DelayTime	0.1
VscForce	0

Output	
output_error	error

Description

High Voltage Measurement of current Id while forcing Vd with stepping Vg.

Device Connection:

Drain--Ki237

Gate, Source, Sub, Well -- each corresponds with one of SMU1 or SMU2 or ..... or SMU8 of 4200

---instAddr:GPIB address of 237. valid from 0~30

---GateSMU:The system terminal touch to Gate of the MOS-FET.if 'GNDU' is chosen, the terminal should be connected to GND manually

---SourceSMU:The system terminal touch to Source of the MOS-FET.if 'GNDU' is chosen, the terminal should be connected to GND manually

---SubSMU:The system terminal touch to Sub of the MOS-FET.if 'GNDU' is chosen, the terminal should be connected to GND manually

---WellSMU:The system terminal touch to Well of the MOS-FET.if 'GNDU' is chosen, the terminal should be connected to GND manually; if there is not Well terminal, pls. choose 'NONE'.

---VgStart:Start Voltages of Gate in Volt

---VgStop:End Voltages of Gate in Volt

---VgPoint:Number of intervals of forced Vg.

---VdStart:Start Voltages of Drain in Volt.

---VdStop:End Voltages of Drain in Volt.

---VdPoint:Number of intervals of forced Vd.

---IdLimit:Current Limitation on measured sites in Ampere.

---Integration:A/D integration speed

0: fast

1: medium

2: long, 1PLC (60Hz)

3: long, 1PLC (50Hz)

---DelayTime:delay Time of with once measurement in s.

---VscForce:Voltage bias force to Source.

---VsbForce:Voltage bias force to Sub.

---VwForce:Voltage bias force to Well

---VgMsrFlag:Flag for determining weather Vg is measured or not

---IgMsrFlag:Flag for determining weather Ig is measured or not

---VscMsrFlag:Flag for determining weather Vsc is measured or not

---IscMsrFlag:Flag for determining weather Isc is measured or not

---VsbMsrFlag:Flag for determining weather Vsb is measured or not

---IsbMsrFlag:Flag for determining weather Isb is measured or not

---VwMsrFlag:Flag for determining weather Vw is measured or not

---IwMsrFlag:Flag for determining weather Iw is measured or not

You can see in the previous figure that there are three areas in the Setup tab of a standard PTM: Input, Output, and Description.

Set the Input parameters. Refer to the description in the right panel. If the input parameter cell is an edit box, enter a value. If the input parameter cell is a drop-down list, select the desired item.

Set the output parameter names. Enter a string and it will appear on the Data tab as an output. Save the configuration.

Configure a XRC PTM

The XRCed is a tool for creating a GUI. The PTM can include a .xrc file with a test module that will be the user interface for setting input values.

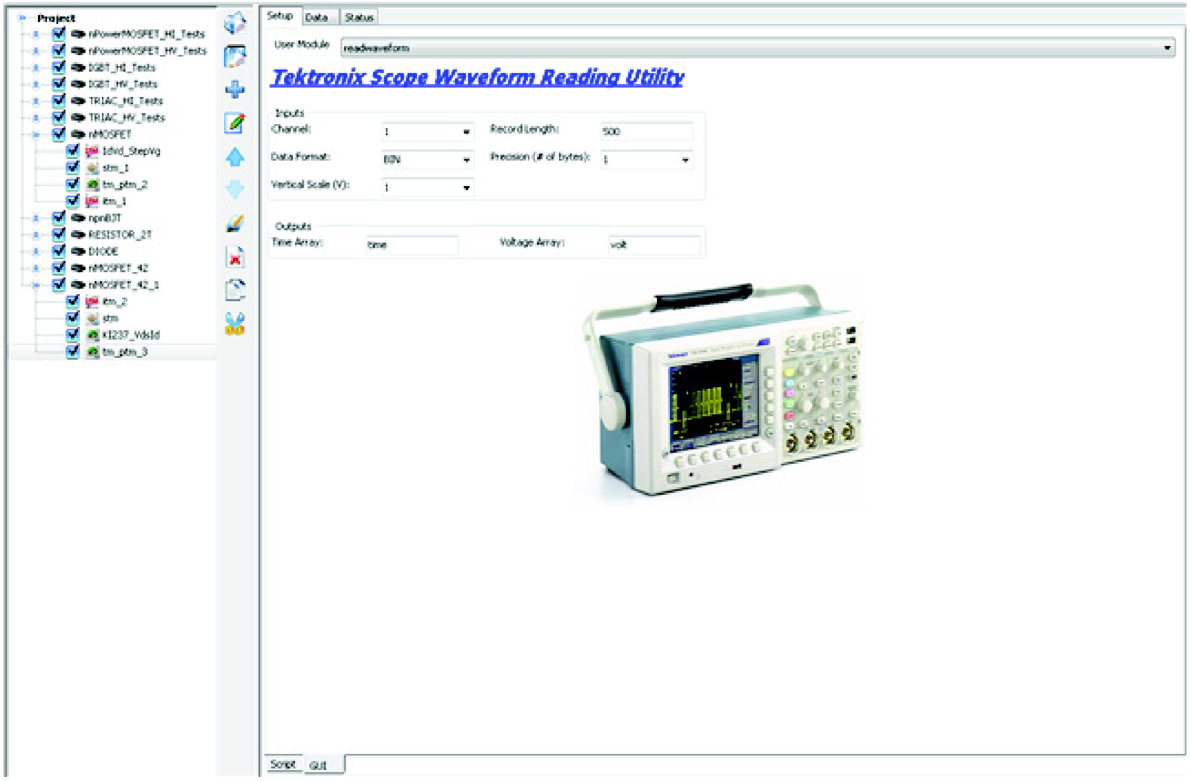
If the imported PTM file has a corresponding .xrc GUI file, ACS Basic automatically loads and opens the GUI. You can set parameters on the GUI rather than editing parameters on the Setup tab.

For more information on how you can generate the .xrc file, refer to Create a XRC GUI file topic.

Configure a PTM with XRC:

After the XRC PTM is imported, click the test name in the configuration navigator. The module will open and the GUI is displayed by default (see next figure). The script code is in the Setup tab.

Figure 163: Import a XRC PTM



User Module: readwaveform

Tektronix Scope Waveform Reading Utility

Inputs

Channel: 1

Record Length: 500

Data Format: BIN

Precision (# of bytes): 1

Vertical Scale (V): 1

Outputs

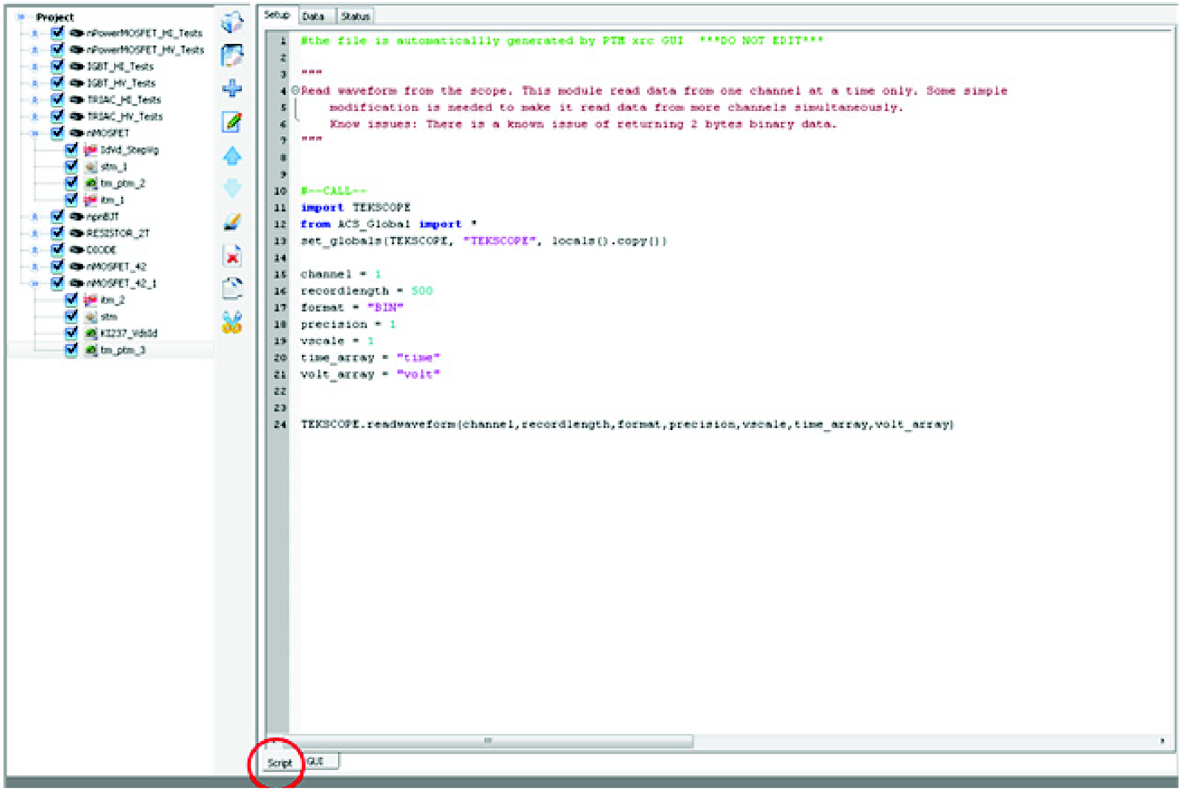
Time Array: time

Voltage Array: volt

1. Set the Input parameters. Refer to the Common library topic for more information.
2. Save the configuration.

The inputs in the GUI are related to the function call in the Setup tab. To open, click the Script tab within the Setup tab (see next figure).

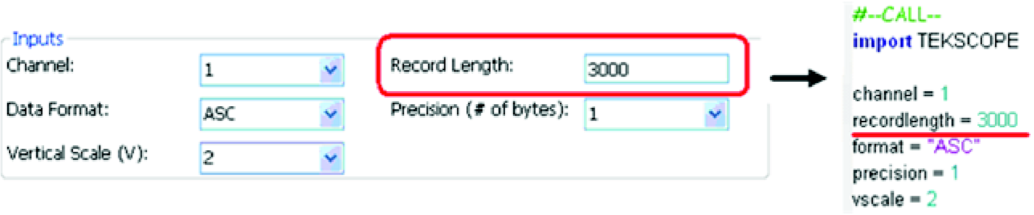
Figure 164: The Setup tab of a XRC PTM



```
#the file is automaticallly generated by PTM xrc GUI  ***DO NOT EDIT***

"""
Read waveform from the scope. This module read data from one channel at a time only. Some simple
    modification is needed to make it read data from more channels simultaneously.
    Know issues: There is a known issue of returning 2 bytes binary data.
"""


#--CALL--
import TEKSCOPE
from ACS_Global import *
set_globals(TEKSCOPE, "TEKSCOPE", locals().copy())

channel = 1
recordlength = 500
format = "BIN"
precision = 1
vscale = 1
time_array = "time"
volt_array = "volt"


TEKSCOPE.readwaveform(channel,recordlength,format,precision,vscale,time_array,volt_array)
```

After the correct input is entered in the GUI, the function call area of the Setup tab for this test module will automatically update (see next figure).

Figure 165: The function call automatic update


Inputs

Channel: 1

Data Format: ASC

Vertical Scale (V): 2

Record Length: 3000

Precision (# of bytes): 1

```
#--CALL--
import TEKSCOPE

channel = 1
recordlength = 3000
format = "ASC"
precision = 1
vscale = 2
```

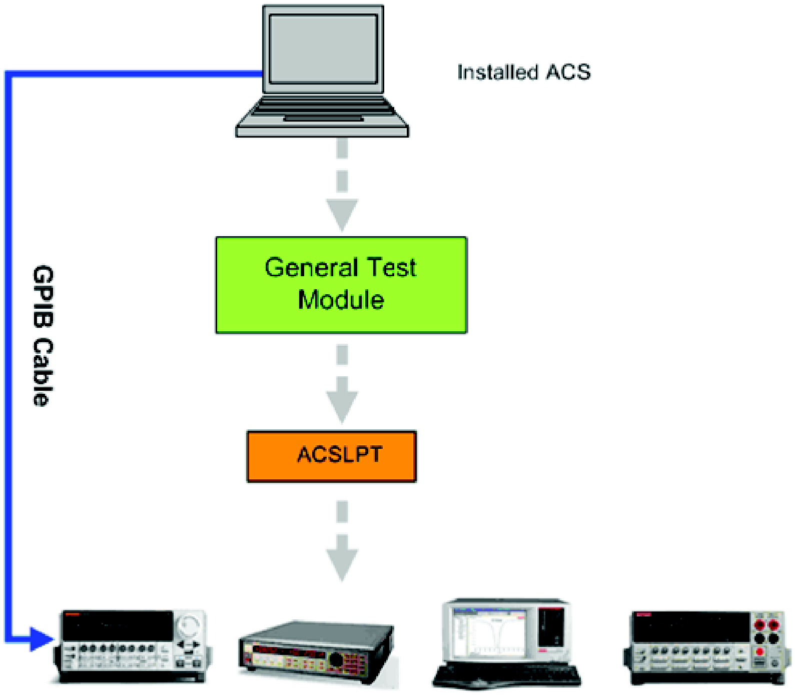
Configure a general purpose PTM

The general purpose PTM is used to perform measurements with SMUs from different Keithley product lines. Its GUI provides an interface similar to the standard interactive test modules (ITM) for the Series 2600 and 2600B, therefore, no programming is required.

You can see in the next figure that the general purpose test module can be used with any combination of the following SMUs: Series 2400, Series 2600, Series 2600B, Model 4200-SCS, and Model 237.

For more information on how to configure different GPIB instruments, refer to the Hardware configuration topic.

Figure 166: General purpose module with multiple SMUs



The general purpose module in ACS Basic 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 different types of SMUs, it can also be used in other applications, such as solar cell and LED testing, among others.

Due to hardware platform differences in the module-compatible SMUs, the general purpose module interface only supports the following forcing functions: BiasV/I, SweepV/I, and StepV/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 auto ranging.

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 Basic hardware configuration.

The other Common Settings are similar to 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 2600/2600B SourceMeter instruments for a particular test, it is recommend that you use either an ITM or STM for optimum SMU performance.

Open the general purpose PTM

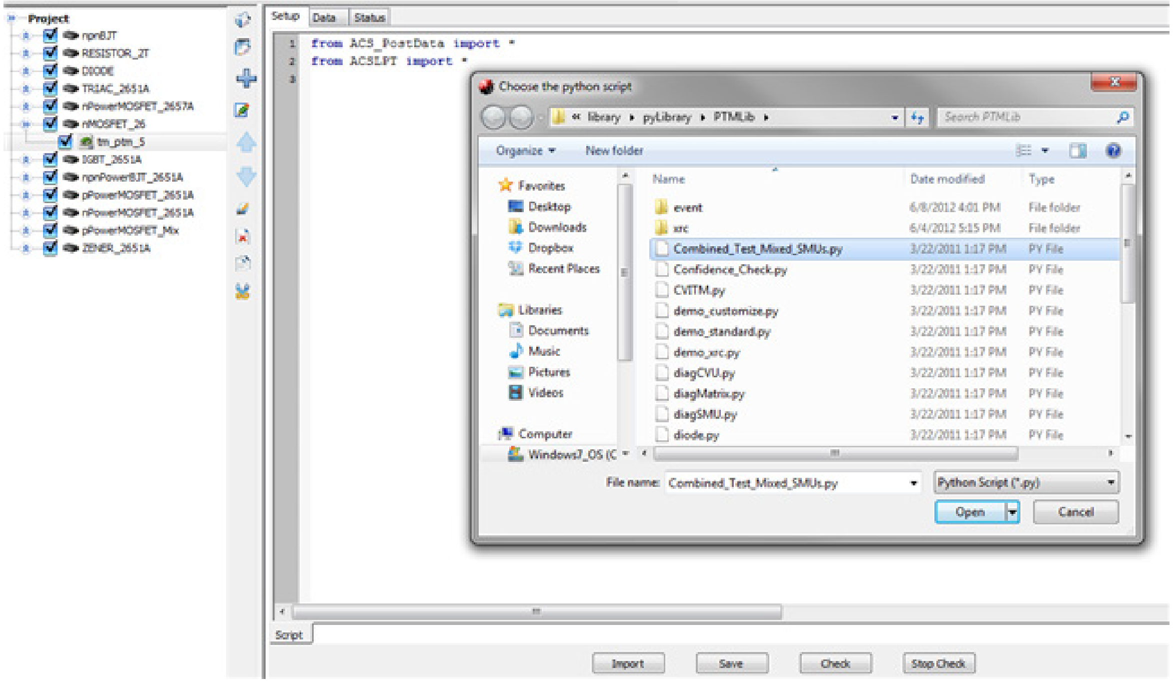
Configure the hardware system

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 Basic, refer to the Hardware configuration topic.

Add a new general purpose module

Click the **Add new test** icon ✚, then click **PTM** in the Specify the Test dialog box. Open the PTM, and click the **Import** function. In the dialog box that opens, select the Combined_Test_Mixed_SMUs.py script (see next figure):

Figure 167: Import the General PTM



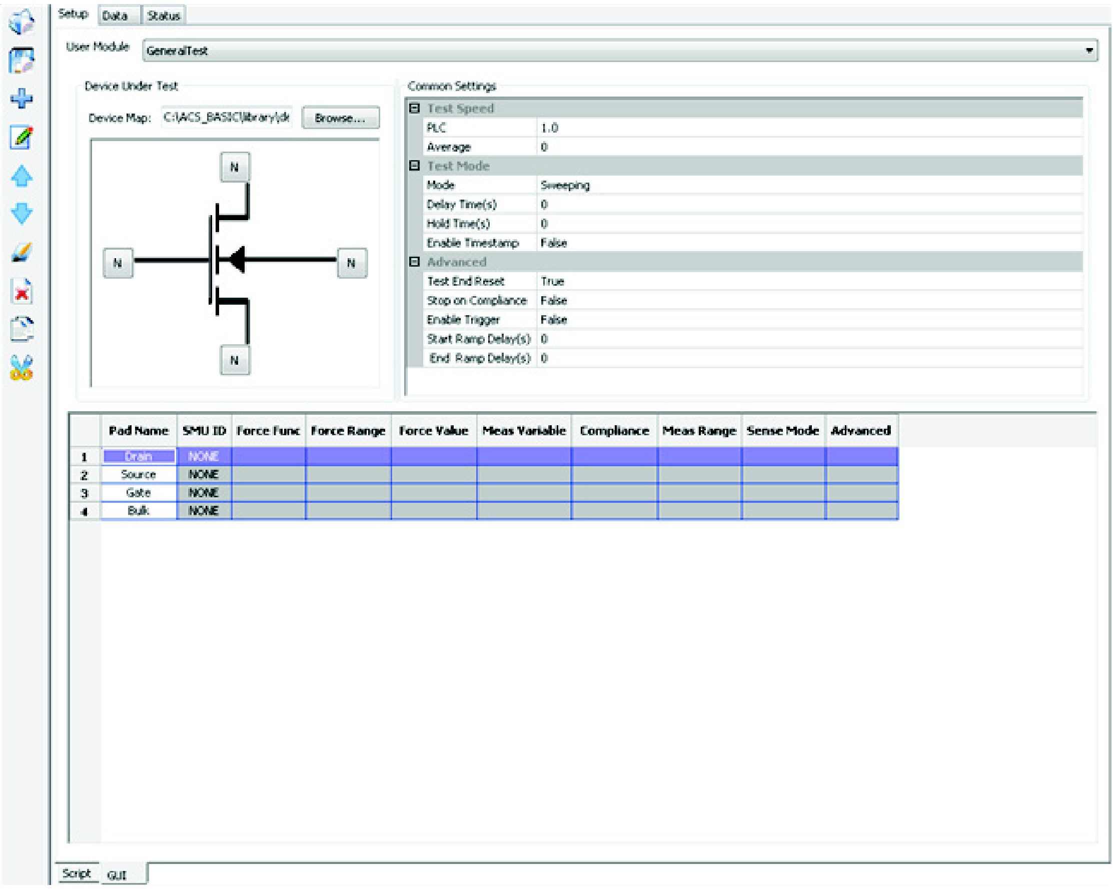
```
from ACS_PostData import *
from ACSLPT import *
```

Name	Date modified	Type
event	6/8/2012 4:01 PM	File folder
xrc	6/4/2012 5:15 PM	File folder
Combined_Test_Mixed_SMUs.py	3/22/2011 1:17 PM	PY File
Confidence_Check.py	3/22/2011 1:17 PM	PY File
CVITM.py	3/22/2011 1:17 PM	PY File
demo_customize.py	3/22/2011 1:17 PM	PY File
demo_standard.py	3/22/2011 1:17 PM	PY File
demo_xrc.py	3/22/2011 1:17 PM	PY File
diagCVU.py	3/22/2011 1:17 PM	PY File
diagMatrix.py	3/22/2011 1:17 PM	PY File
diagSMU.py	3/22/2011 1:17 PM	PY File
diode.py	3/22/2011 1:17 PM	PY File

File name: Combined_Test_Mixed_SMUs.py

The general purpose PTM GUI opens on the ACS Basic GUI (see next figure).

Figure 168: General purpose PTM GUI



User Module: GeneralTest

Device Under Test

Device Map: C:\ACS_BASIC\library\dk

Common Settings

Test Speed	
PLC	1.0
Average	0
Test Mode	
Mode	Sweeping
Delay Time(s)	0
Hold Time(s)	0
Enable Timestamp	False
Advanced	
Test End Reset	True
Stop on Compliance	False
Enable Trigger	False
Start Ramp Delay(s)	0
End Ramp Delay(s)	0

	Pad Name	SMU ID	Force Func	Force Range	Force Value	Meas Variable	Compliance	Meas Range	Sense Mode	Advanced
1	Drain	NONE								
2	Source	NONE								
3	Gate	NONE								
4	Bulk	NONE								

Configure the general purpose PTM

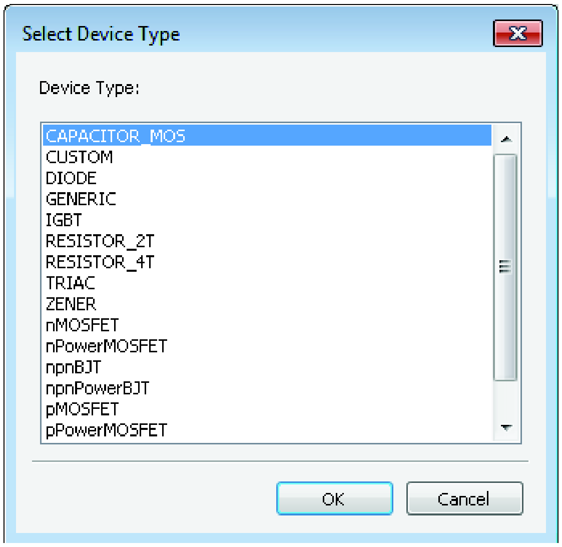
After inserting a general purpose PTM into your project, you must configure it to meet your test requirements.

Click the Browse function, and the Select Device Type dialog box will show various Device Types for you to choose (see next figure).

NOTE

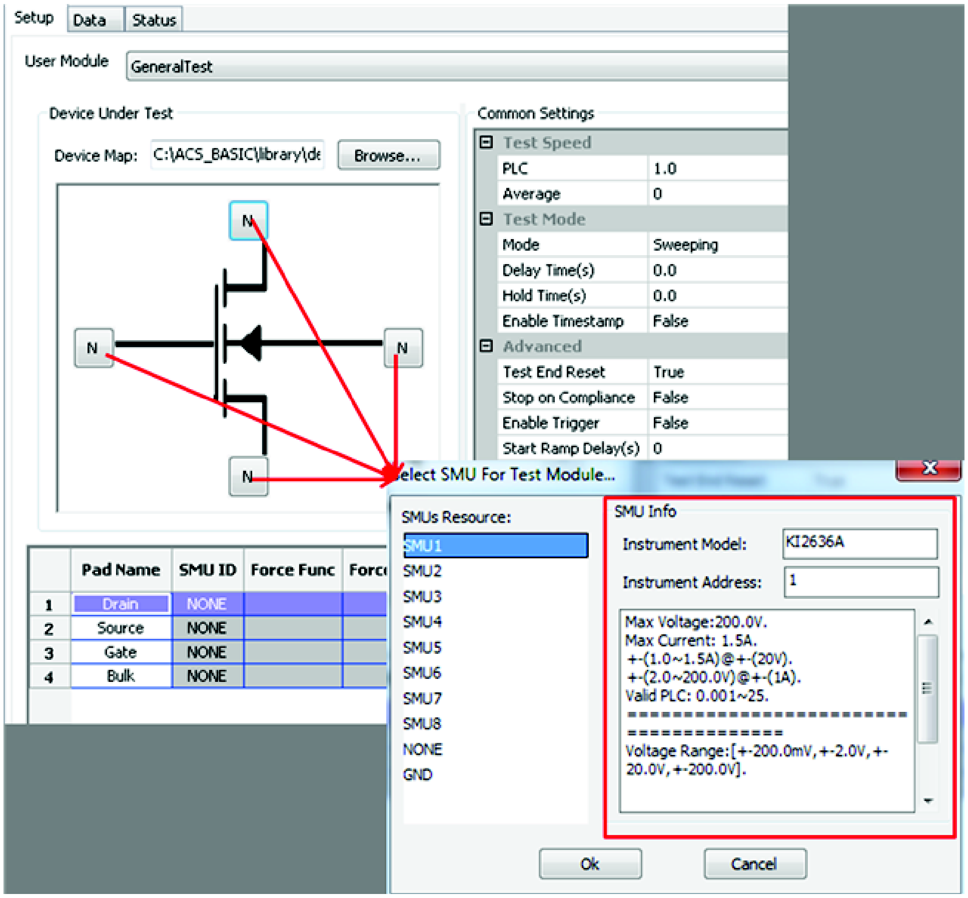
The device you select in the general purpose PTM GUI is independent of the device in the configuration navigator. You must select a device in the module GUI to configure the general purpose PTM. For consistency with device definitions, you should select the same device for the general purpose module 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 169: Select Device type dialog box



Next, specify which SMU is connected to each device terminal. Click on a device terminal and the Select SMU For Test Module dialog box opens. Select the appropriate SMU and the device terminal will reflect your selection once it is made (see next figure).

Figure 170: Map SMU and device terminal



User Module: GeneralTest

Device Map: C:\ACS_BASIC\library\de

⊟ Test Speed	
PLC	1.0
Average	0
⊟ Test Mode	
Mode	Sweeping
Delay Time(s)	0.0
Hold Time(s)	0.0
Enable Timestamp	False
⊟ Advanced	
Test End Reset	True
Stop on Compliance	False
Enable Trigger	False
Start Ramp Delay(s)	0

	Pad Name	SMU ID	Force Func	Forc
1	Drain	NONE		
2	Source	NONE		
3	Gate	NONE		
4	Bulk	NONE		

SMUs Resource:

- SMU1
- SMU2
- SMU3
- SMU4
- SMU5
- SMU6
- SMU7
- SMU8
- NONE
- GND

Instrument Model: KI2636A

Instrument Address: 1

```
Max Voltage:200.0V.
Max Current: 1.5A.
+-(1.0~1.5A)@+-(20V).
+-(2.0~200.0V)@+-(1A).
Valid PLC: 0.001~25.
=========================
==============
Voltage Range:[+-200.0mV,+-2.0V,+-
20.0V,+-200.0V].
```

NOTE

The Model 4200-SMU cannot use general purpose PTMs if ACS Basic is installed on a PC,. This means that the Model 4200-SMU will not be displayed in the Select SMU For Test Module dialog box. Only when ACS Basic is installed on the Model 4200-SCS will the Model 4200-SMU be available for general purpose PTMs.

NOTE

It is not a good practice to mix different SMU model numbers on a device. ACS Basic will provide you a warning if you try to mix SMUs inadvertently.

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 GUI. 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, or 2602A/2602B.

A Model 2651A SMU cannot be mixed with low-current SMUs.

For two-terminal devices, a Model 2651A cannot be mixed 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, and a low-current SMU is connected to the gate, the low-current SMU cannot be connected to the source, if the Model 2651A SMU is connected to the drain.

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.

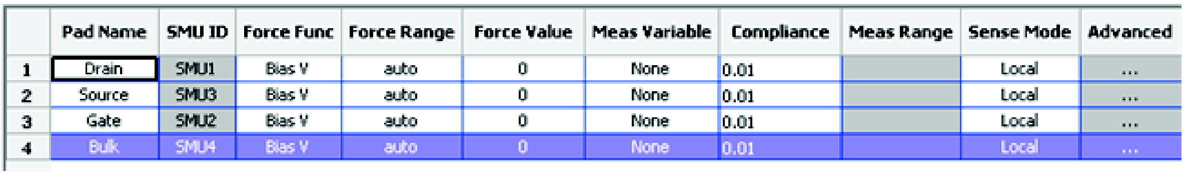
You must also set the Common Settings information in the PTM GUI. This includes the Test Speed, Test Mode, and Advanced information.

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 are used to configure the forcing function and measurements implemented by the instrument.

The following figure indicates all of the parameters that need configured for forcing and measuring.

Figure 171: Force and Measure parameters

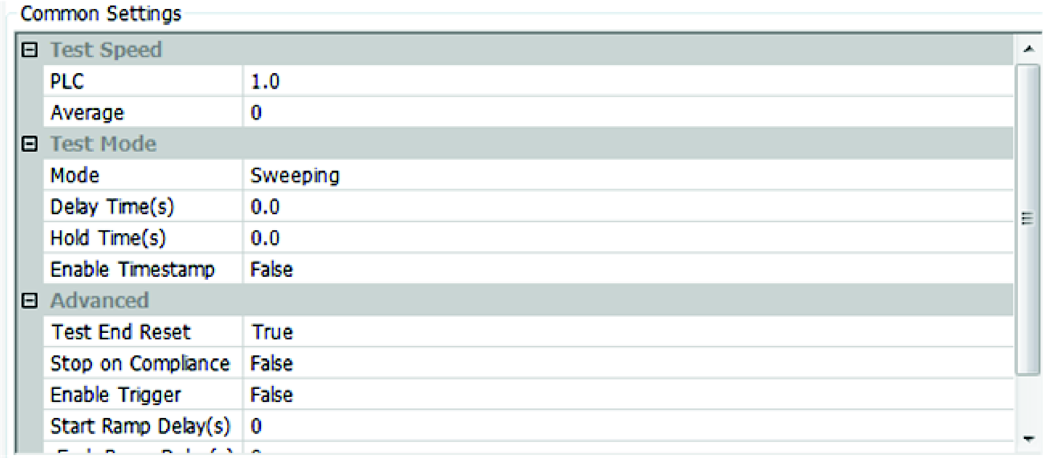


	Pad Name	SMU ID	Force Func	Force Range	Force Value	Meas Variable	Compliance	Meas Range	Sense Mode	Advanced
1	Drain	SMU1	Bias V	auto	0	None	0.01		Local	...
2	Source	SMU3	Bias V	auto	0	None	0.01		Local	...
3	Gate	SMU2	Bias V	auto	0	None	0.01		Local	...
4	Bulk	SMU4	Bias V	auto	0	None	0.01		Local	...

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 (see next figure).

Figure 172: General purpose test module Common Settings



Common Settings

⊟ Test Speed	
PLC	1.0
Average	0
⊟ Test Mode	
Mode	Sweeping
Delay Time(s)	0.0
Hold Time(s)	0.0
Enable Timestamp	False
⊟ Advanced	
Test End Reset	True
Stop on Compliance	False
Enable Trigger	False
Start Ramp Delay(s)	0

Test Speed: Controls the A/D (analog-to-digital) converter integration time used to measure a signal. Each measured reading by an 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, however it will take longer, which decreases the speed.

- **PLC** (power line cycle): For optimum immunity to line cycle noise, use an integer number for the PLC (for example, 1, 2, 3, etc.). The integration time setting is based on the number of power line cycles (NPLCs):

  For 60Hz line power, 1.0 PLO = 16.67msec (1/60)

  For 50Hz line power, 1.0 PLO = 20msec (1/50)
- **Average**: The Average edit box 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] is used to disable the average number setup

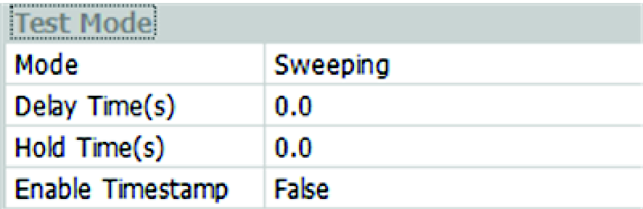
The next table shows the range of allowed values for a NPLC and the Average field setting.

Model	PLC	Average
KI24XX	[0.01 ~ 10]	[0 ~ 100]
KI23X	[0.01 ~ 1]	[0, 1, 2, 4, 8, 16, 32]
KI26XX	[0.001, 25]	ignored
KI4200 (on the 4200)	[0.01 ~ 10]	ignored

Test Mode: The Mode edit box lets you choose one of two possible test modes: Sweep or Sampling (see next figure).

- **Sweep Mode**: Applies to any general test module in which one or more forced voltages/currents vary with time.

Figure 173: Sweep Test Mode settings



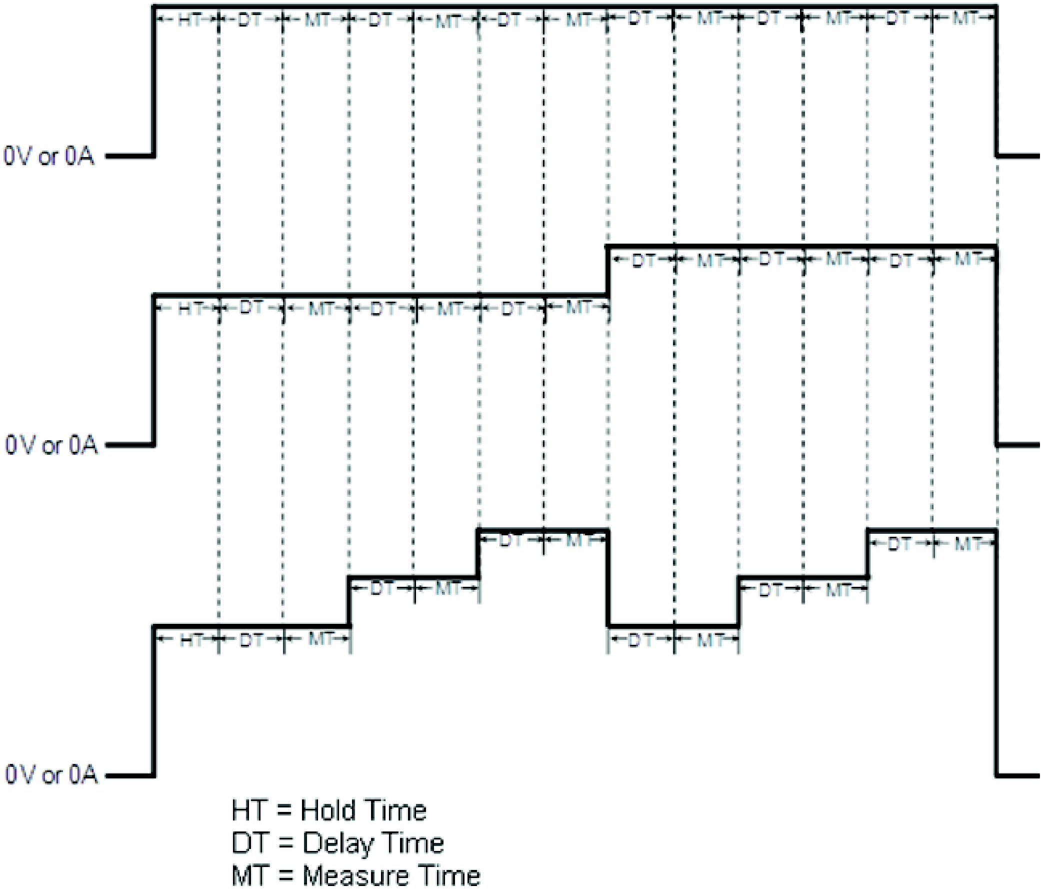
Test Mode

Mode	Sweeping
Delay Time(s)	0.0
Hold Time(s)	0.0
Enable Timestamp	False

- **Hold Time(s)**: sets the time delay before the first measurement is taken. The starting voltage or current in a sweep may be substantially larger than the subsequent voltage or current increments of the sweep. Accordingly, the source settling time required for the first point of the sweep may be substantially larger than the settling times required for the subsequent increments of the sweep. To compensate, you can specify a Hold Time delay at the beginning of each sweep. You can specify a Hold Time delay of 0 to 1000 seconds. The default Hold Time is 0s.

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.

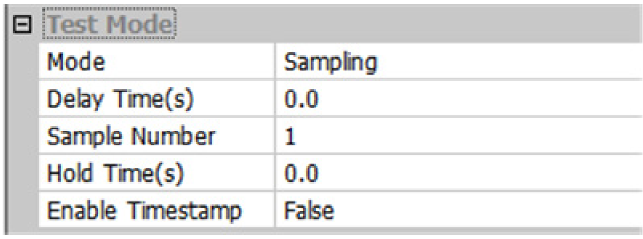
Figure 174: Sweep mode timing diagram



- **Hold Time(s)**: Specifies the time delay before the first measurement is taken. After the initial application of voltage or current by the SMU(s), 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 of 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 of 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 of the forced voltages or currents are constant (bias I or bias V), measurements made at timed intervals. For example, sampling mode is used to record one or more static measurements, or to time-profile the charging voltage of a capacitor while forcing a constant current.

ACS Basic enables the Sampling mode option only when all of 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 (see next figure).

Figure 175: Sampling Test Mode settings



⊟ Test Mode

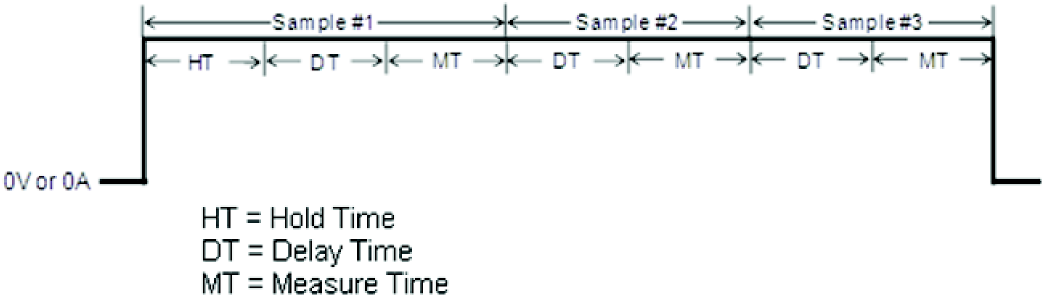
Mode	Sampling
Delay Time(s)	0.0
Sample Number	1
Hold Time(s)	0.0
Enable Timestamp	False

The Sampling mode measures voltage or current as a function of time while forcing constant voltage or current: voltage bias, or current bias. For example, the sampling mode is used to measure voltage while forcing a constant 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.The Delay Time interval, Sample Number, and Hold Time settings control the Sampling mode.

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 (see next figure).

Figure 176: Sampling mode timing diagram

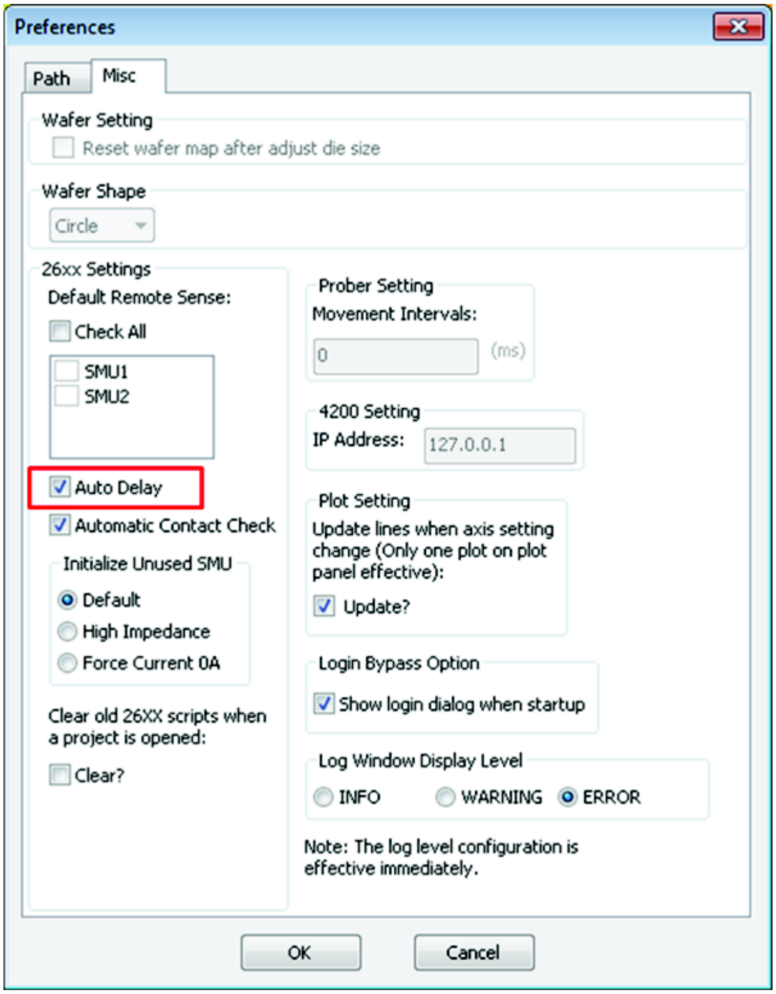


- **Hold Time**: Specifies the time delay before the first measurement is taken. After the initial application of voltage or current by the SMU(s), the source settling time can be substantial. To allow for settling, you can specify an extra Hold Time delay to be applied before making the first measurement. You can specify a Hold Time from 0 to 1000 seconds. Hold Time is a global setting, and is therefore the same for all of 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 seconds.
- **Sample Number**: Specifies the number of data points to be acquired. The Sample Number can be set from 1 to 4096.

In the previous two figures, there is a range-dependent delay (D). A range-dependent delay is automatically applied by an SMU before each measurement to allow for source settling. All of the SMUs in the test system are synchronized, therefore the range-dependant delay time applied by the most delayed SMU is the time applied to all of the SMUs. If the Auto Delay box in the Misc tab of the Preferences dialog box is selected (go to the Tools drop-down to find Preferences), the range-dependant Delay time will be applied to the Models 2635, 2636, 2635A, and 2636A, but not for other 2600 or 2600B models (see next figure).

The SMU will delay the Measure Time (MT) before taking a measurement after forcing.

Figure 177: Auto Delay on Preferences Misc tab



Preferences

Path | Misc

Wafer Setting
□ Reset wafer map after adjust die size

Wafer Shape
Circle

26xx Settings
Default Remote Sense:
□ Check All
□ SMU1
□ SMU2

☑ Auto Delay
☑ Automatic Contact Check

Initialize Unused SMU
● Default
○ High Impedance
○ Force Current 0A

Clear old 26XX scripts when a project is opened:
□ Clear?

Prober Setting
Movement Intervals: 0 (ms)

4200 Setting
IP Address: 127.0.0.1

Plot Setting
Update lines when axis setting change (Only one plot on plot panel effective):
☑ Update?

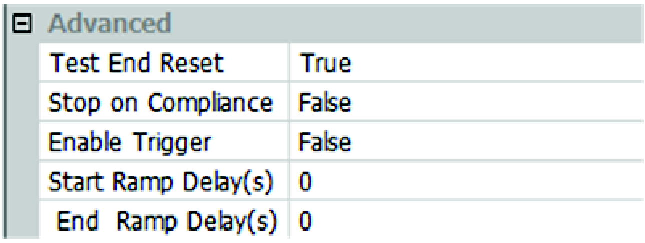
Login Bypass Option
☑ Show login dialog when startup

Log Window Display Level
○ INFO ○ WARNING ● ERROR

Note: The log level configuration is effective immediately.

Advanced: In the Advanced area there are five items for you to configure for your testing requirements (see next figure).

Figure 178: Advanced area settings



⊟ Advanced

Test End Reset	True
Stop on Compliance	False
Enable Trigger	False
Start Ramp Delay(s)	0
End Ramp Delay(s)	0

Test End Reset: There are two choices:

1. False: your test results will not be reset in the instruments when testing has completed.
2. True: after testing is complete, all of the test results will be reset in the instruments.

Stop on Compliance: There are two choices:

1. False: the test will not exit once it reaches the setting compliance (limit) value.
2. True: the test will exit once it reaches the setting compliance (limit) value.

Enable Trigger: Used for the Model 237 with the Model 2361 Trigger controller. There are two choices:

1. False: do not capture the measurement trigger value.
2. 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.

Start Ramp Delay: Determines when to start the ramp delay. The default value is 0 seconds.

End Ramp Delay: Determines when to end the ramp delay. The default value is 0 seconds.

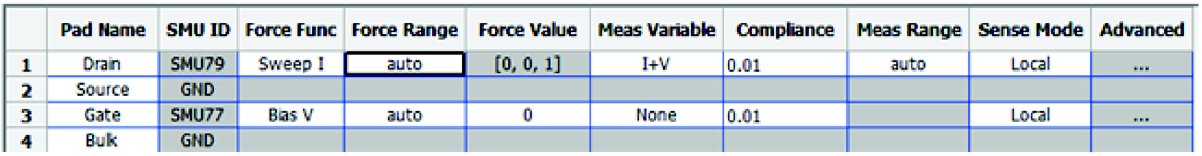
NOTE

For most SMUs, you can choose False for the Enable Trigger setting.

Configure the Force and Measure functions/variables

The Force and Measure settings are associated with an instrument object that is assigned to a device terminal. These settings are used to configure the forcing function and measurements implemented by the instrument (see next figure).

Figure 179: 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:

1. Voltage bias (Bias V)
2. Current bias (Bias I)
3. Open
4. Voltage sweep (Sweep V)
5. Current sweep (Sweep I)
6. Voltage step (Step V)
7. Current step (Step I)

For more information on how to set the voltage sweep and current sweep, refer to Configure force functions.

Sense Mode: There are two choices.

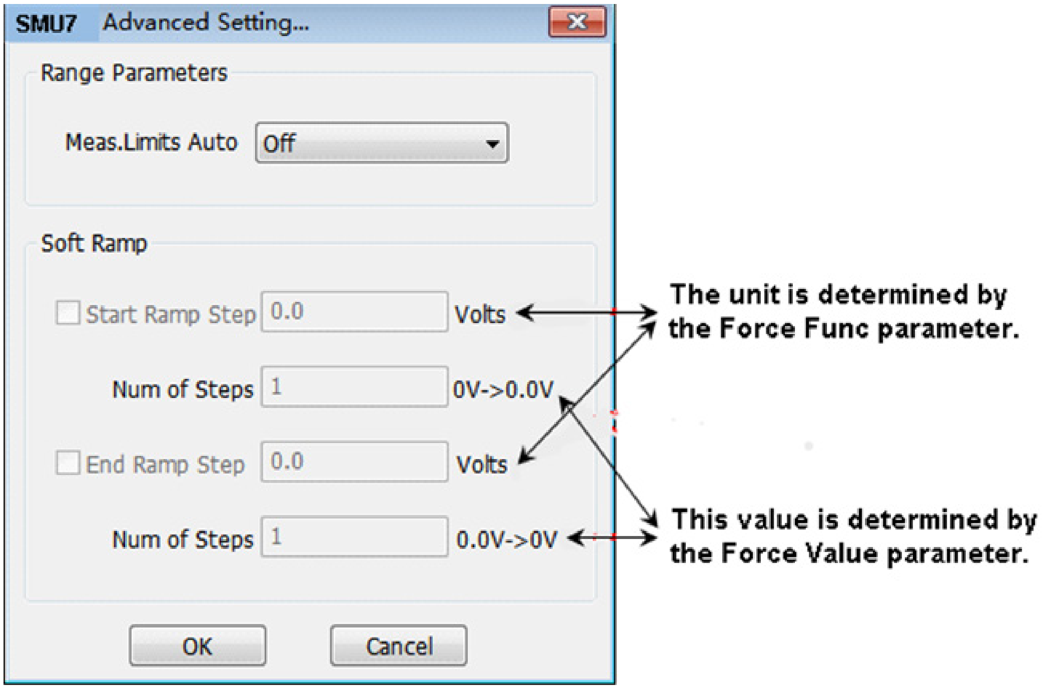
1. Local: two-wire sensing
2. Remote: four-wire sensing

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 the item, then open the Advanced Setting dialog (see next figure).

Figure 180: SMU Advanced Setting parameters



SMU7 Advanced Setting...

Range Parameters

Meas.Limits Auto: Off

Soft Ramp

□ Start Ramp Step: 0.0 Volts

Num of Steps: 1 0V->0.0V

□ End Ramp Step: 0.0 Volts

Num of Steps: 1 0.0V->0V

The unit is determined by the Force Func parameter.

This value is determined by the Force Value parameter.

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 is used to specify 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/off). If the Meas Range parameter is not set to auto, the Meas. Limits Auto function is disabled/off.

The next table identifies conditions when the Meas. Limits Auto parameter is enabled:

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: enables you to apply incremental test readings at the beginning of the bias or first sweep/list point tests, and at the end of the bias or last sweep/list point tests. This function allows the forced voltage or current to achieve the desired value after a certain number of steps.

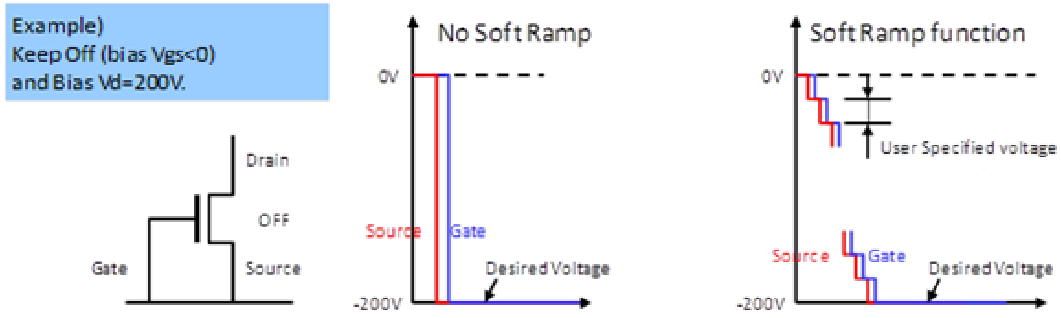
Start Ramp Step: The step value determines when to start the ramp.

Stop Ramp Step: The step value determines when to stop the ramp.

Num of Steps: The number of steps defined for all of the SMUs, however, the start and stop steps are defined separately. The default value for Num of steps is 1. It's value depends on the Force Value and the Start/Stop Ramp Steps.

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

Figure 181: Comparison of Soft Ramp and without Soft Ramp



Section 7

ACS Basic script editor tool

In this section:

Introduction

The Keithley 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 test script python programming language subroutines (or functions) or the TSP script language. User libraries are created to control instrumentation, analyze data, or perform any other system automation task programmatically. Once a user library has been created using the script editor, its user modules will run using ACS Basic software.

The script editor manages user libraries in a structured manner. It also provides a graphical user interface (GUI) that helps you to effectively enter code, compile a user module, and build a user library. It provides user-library management features, such as open, delete, copy module and library, check, edit and browse, save and export functions. You can create your own user libraries.

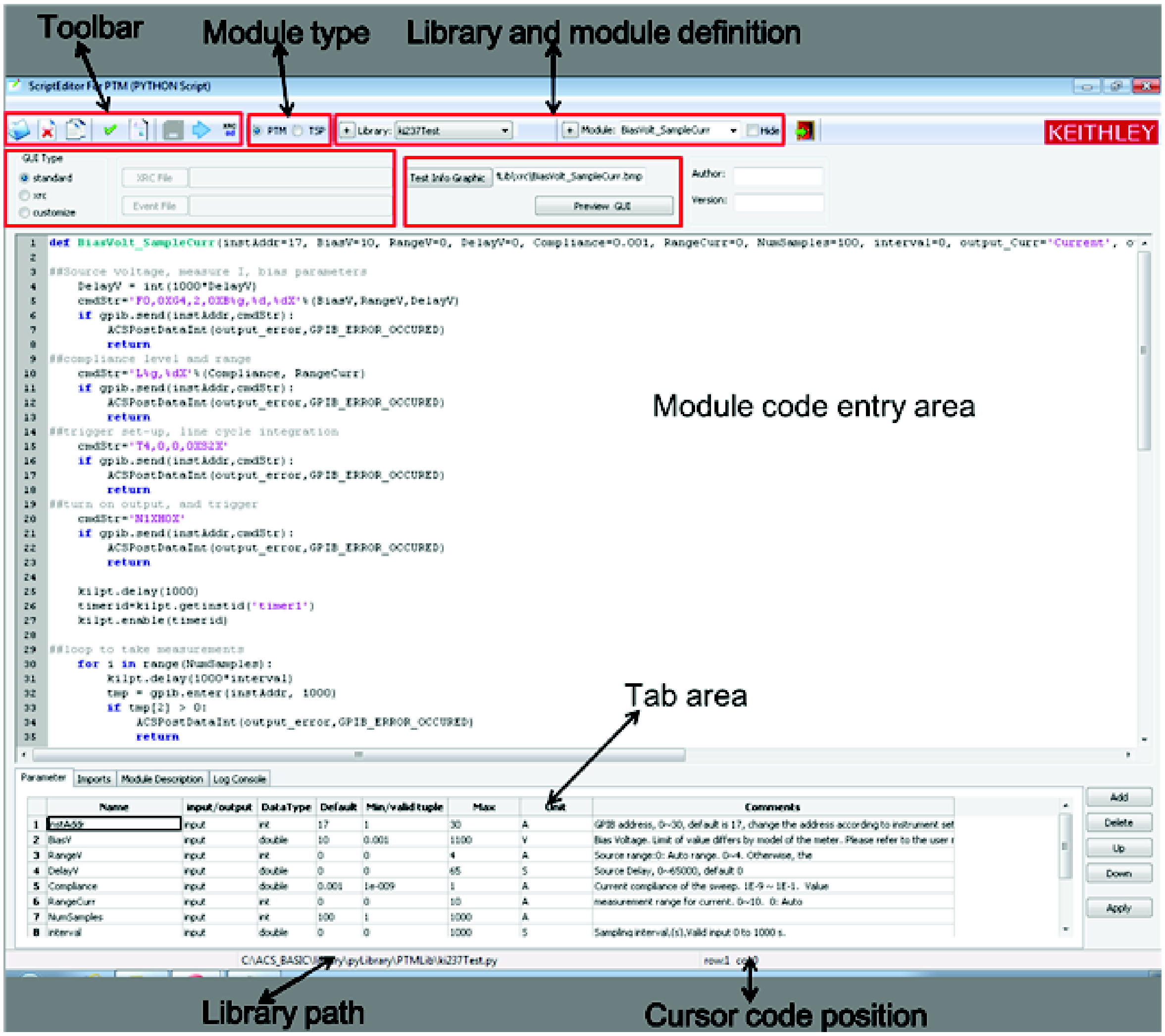
To run script editor user modules in ACS Basic, you must import or create a test module as a TSP or PTM and connect it to the user module. Once the user module is connected to the test module, the following occurs each time ACS Basic runs the STM or PTM:

- Dynamically loads the user module and the appropriate user library.
- Passes the user-module parameters (stored in the STM or PTM) to the user module.
- Generated data by the user module is returned to the STM or PTM for interactive analysis.

Script Editor GUI

The Script Editor GUI provides all the icons, 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 (see next figure).

Figure 182: Script Editor GUI



● PTM ○ TSP

Library: ki237Test

Module: BiasVolt_SampleCurr □ Hide

GUI Type

● standard

○ xrc

○ customize

Test Info Graphic: %lib\xrc\BiasVolt_SampleCurr.bmp

Author:

Version:

```
def BiasVolt_SampleCurr(instAddr=17, BiasV=10, RangeV=0, DelayV=0, Compliance=0.001, RangeCurr=0, NumSamples=100, interval=0, output_Curr='Current', o

##Source voltage, measure I, bias parameters
    DelayV = int(1000*DelayV)
    cmdStr='F0,0X04,2,0XB%g,%d,%dX'%(BiasV,RangeV,DelayV)
    if gpib.send(instAddr,cmdStr):
        ACSPostDataInt(output_error,GPIB_ERROR_OCCURED)
        return
##compliance level and range
    cmdStr='L%g,%dX'%(Compliance, RangeCurr)
    if gpib.send(instAddr,cmdStr):
        ACSPostDataInt(output_error,GPIB_ERROR_OCCURED)
        return
##trigger set-up, line cycle integration
    cmdStr='T4,0,0,0XS2X'
    if gpib.send(instAddr,cmdStr):
        ACSPostDataInt(output_error,GPIB_ERROR_OCCURED)
        return
##turn on output, and trigger
    cmdStr='N1XH0X'
    if gpib.send(instAddr,cmdStr):
        ACSPostDataInt(output_error,GPIB_ERROR_OCCURED)
        return

    kilpt.delay(1000)
    timerid=kilpt.getinstid('timer1')
    kilpt.enable(timerid)

##loop to take measurements
    for i in range(NumSamples):
        kilpt.delay(1000*interval)
        tmp = gpib.enter(instAddr, 1000)
        if tmp[2] > 0:
            ACSPostDataInt(output_error,GPIB_ERROR_OCCURED)
            return
```

	Name	input/output	DataType	Default	Min/valid tuple	Max	Unit	Comments
1	instAddr	input	int	17	1	30	A	GPIB address, 0~30, default is 17, change the address according to instrument set
2	BiasV	input	double	10	0.001	1100	V	Bias Voltage. Limit of value differs by model of the meter. Please refer to the user r
3	RangeV	input	int	0	0	4	A	Source range:0: Auto range. 0~4. Otherwise, the
4	DelayV	input	double	0	0	65	S	Source Delay, 0~65000, default 0
5	Compliance	input	double	0.001	1e-009	1	A	Current compliance of the sweep. 1E-9 ~ 1E-1. Value
6	RangeCurr	input	int	0	0	10	A	measurement range for current. 0~10. 0: Auto
7	NumSamples	input	int	100	1	1000	A	
8	interval	input	double	0	0	1000	S	Sampling interval,(s),Valid input 0 to 1000 s.

C:\ACS_BASIC\library\pyLibrary\PTMLib\ki237Test.py

row1 col0

The module identification area

The module type, library, and module name are located next to the toolbar. The components of this area are used as follows (see previous figure).

- **Module Type**: displays the type of the active (currently open) user module. There are two types of supported modules: STM and PTM. For more information on STMs and PTMs, refer to Configure a script test module (STM) and Configure a python test module (PTM).
- **Library Name**: displays the name of the active (currently open) user library. Use the Import Library icon to open a specific user library. Use the Delete icon to delete a library. Use the Copy icon to copy a library. Use the Add a new library icon ⊞ Library: to add a new library (refer to the The toolbar topic for more information).

> NOTE
>
> When naming a user module, remember to conform to the case-sensitive python or TSP programming language naming conventions. Do not duplicate names of existing user modules or user libraries.

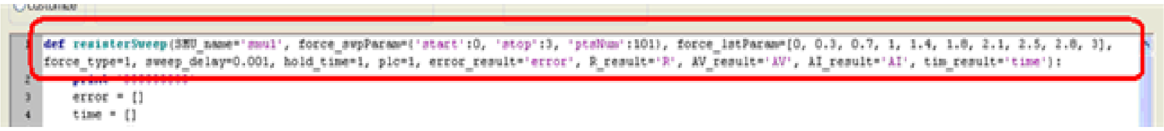
- **Module Name**: displays the name of the active (currently open) user test module.
- **Hide (Library and Module)**: hides the Library and Module information so that it is not displayed in the GUI. If you want the Library and Module information to display in the GUI make sure the Hide option is deselected.
- **GUI type**: displays the GUI type of the active (currently open) user module. There are three types of GUIs available: Standard, XRC, Custom. For more information about a STM with a standard GUI, refer to Configure a script test module (STM). For more information about a STM with a XRC GUI, refer to the Create a XRC GUI file topic.
- **Version info**: Displays creator of the user module and the version information.

The user module code-entry area

The user module code-entry area is where you enter, edit, or view the user module code (see previous figure). The scroll bars located to the right and below the module code-entry area let you move through the code.

The beginning of the module code contains the function definition area. In this area you will find the language function prototype for the user module that reflect the parameters that are specified in the Parameter tab area (see next figures).

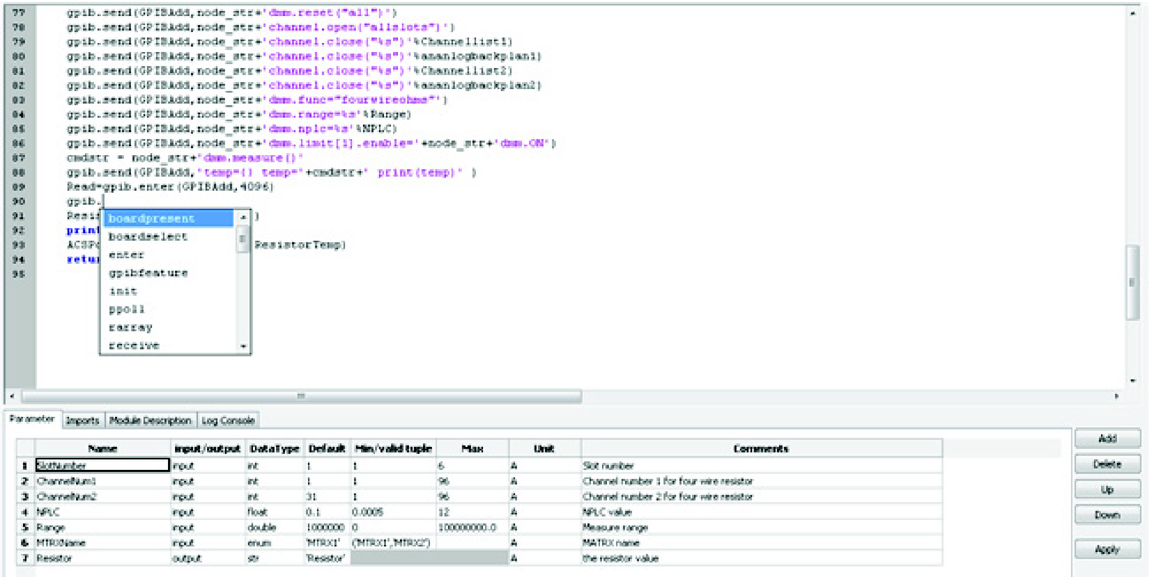
Figure 183: Function definition area



```
def resisterSweep(SMU_name='smu1', force_swpParam={'start':0, 'stop':3, 'ptsNum':101}, force_lstParam=[0, 0.3, 0.7, 1, 1.4, 1.8, 2.1, 2.5, 2.8, 3],
force_type=1, sweep_delay=0.001, hold_time=1, plc=1, error_result='error', R_result='R', AV_result='AV', AI_result='AI', tim_result='time'):
    print '*********'
    error = []
    time = []
```

The parameters that the module depends on to operate are configured in the Parameter tab area. After you add the parameters (by selecting the Add function) and entering the configurations, select the Apply function so that the updates to the module take affect immediately (see next figure).

Figure 184: Script editor enter Python code



```
    gpib.send(GPIBAdd,node_str+'dmm.reset("all")')
    gpib.send(GPIBAdd,node_str+'channel.open("allslots")')
    gpib.send(GPIBAdd,node_str+'channel.close("%s")'%Channellist1)
    gpib.send(GPIBAdd,node_str+'channel.close("%s")'%ananlogbackplan1)
    gpib.send(GPIBAdd,node_str+'channel.close("%s")'%Channellist2)
    gpib.send(GPIBAdd,node_str+'channel.close("%s")'%ananlogbackplan2)
    gpib.send(GPIBAdd,node_str+'dmm.func="fourwireohms"')
    gpib.send(GPIBAdd,node_str+'dmm.range=%s'%Range)
    gpib.send(GPIBAdd,node_str+'dmm.nplc=%s'%NPLC)
    gpib.send(GPIBAdd,node_str+'dmm.limit[1].enable='+node_str+'dmm.ON')
    cmdstr = node_str+'dmm.measure()'
    gpib.send(GPIBAdd,'temp={} temp='+cmdstr+' print(temp)' )
    Read=gpib.enter(GPIBAdd,4096)
    gpib.
    Resis                    )
    print
    ACSPo                    ResistorTemp)
    retu
```

	Name	input/output	DataType	Default	Min/valid tuple	Max	Unit	Comments
1	SlotNumber	input	int	1	1	6	A	Slot number
2	ChannelNum1	input	int	1	1	96	A	Channel number 1 for four wire resistor
3	ChannelNum2	input	int	31	1	96	A	Channel number 2 for four wire resistor
4	NPLC	input	float	0.1	0.0005	12	A	NPLC value
5	Range	input	double	1000000	0	100000000.0	A	Measure range
6	MTRXName	input	enum	'MTRX1'	('MTRX1','MTRX2')		A	MATRX name
7	Resistor	output	str	'Resistor'			A	the resistor value

NOTE

Do not enter the following python code items in the module code-entry area (Script Editor enters these at special locations based on information in other places in the Script Editor GUI):

Import and global data

The function prototype

Do not enter the following TSP code items in the module code-entry area (Script Editor enters these at special locations based on information in other places in the Script Editor GUI):

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 Basic Libraries Reference Manual for more information.

The tab areas

Four tab areas exist:

1. Parameter
2. Imports
3. Module Description
4. Log Console.

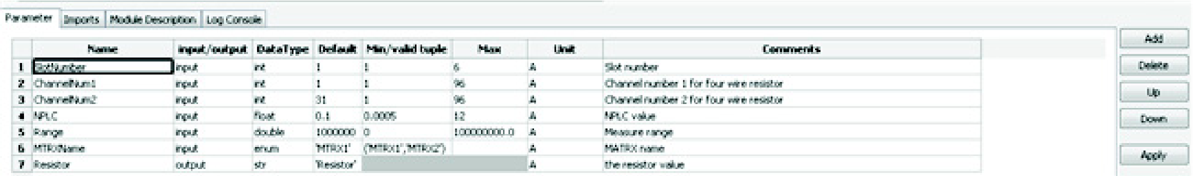
The Imports tab is for PTMs only.

Parameter tab

The Parameter tab area is used to configure and display the following for each parameter that is included in the user module (see next figure).

- Name
- Input/output
- Data Type
- Default
- Min/valid tuple
- Max
- Unit
- Comments

Figure 185: Parameter tab area with examples

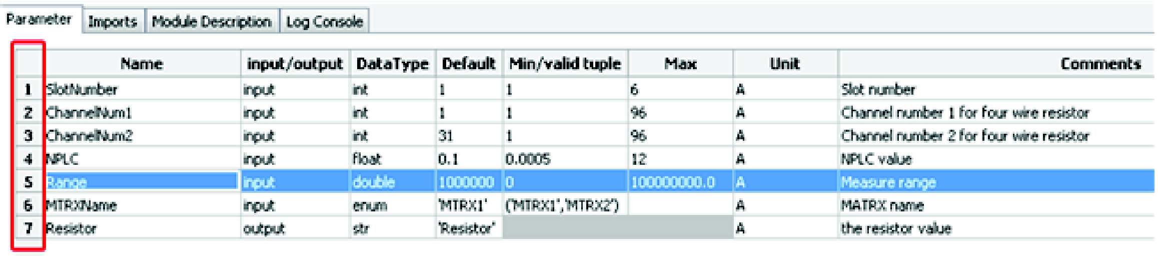


	Name	input/output	DataType	Default	Min/valid tuple	Max	Unit	Comments
1	SlotNumber	input	int	1	1	6	A	Slot number
2	ChannelNum1	input	int	1	1	96	A	Channel number 1 for four wire resistor
3	ChannelNum2	input	int	31	1	96	A	Channel number 2 for four wire resistor
4	NPLC	input	float	0.1	0.0005	12	A	NPLC value
5	Range	input	double	1000000	0	100000000.0	A	Measure range
6	MTRXName	input	enum	'MTRX1'	('MTRX1','MTRX2')		A	MATRX name
7	Resistor	output	str	'Resistor'			A	the resistor value

Add, delete, or apply a parameter:

- Click the **Add** function and enter the information as indicated in the field descriptions.
- Click the parameter name or any of the adjacent fields and click the **Delete** function.
- Select the row you want to move by clicking the row number, then click the **Up** or **Down** function (see next figure).
- Click the **Apply** function to apply changes made in the Parameter tab area.

Figure 186: Select a row of parameters to move



	Name	input/output	DataType	Default	Min/valid tuple	Max	Unit	Comments
1	SlotNumber	input	int	1	1	6	A	Slot number
2	ChannelNum1	input	int	1	1	96	A	Channel number 1 for four wire resistor
3	ChannelNum2	input	int	31	1	96	A	Channel number 2 for four wire resistor
4	NPLC	input	float	0.1	0.0005	12	A	NPLC value
5	Range	input	double	1000000	0	100000000.0	A	Measure range
6	MTRXName	input	enum	'MTRX1'	('MTRX1','MTRX2')		A	MATRX name
7	Resistor	output	str	'Resistor'			A	the resistor value

- **Name**: identifies the parameters that are passed to the user module.
- **input/output**: determines the direction of the data.

NOTE

The string (str) data type can only be used as output.

- **Data Type**: specifies the parameter data type:

  For PTM and TSP:

  - **enum** - enumerable

NOTE

When the data type is enumerable (enum) data, the value of Min/valid tuple field in the parameter tab is a valid tuple. You must set the Min/valid tuple first so that the Default device can be selected in the drop-down list (see next figure).

  - **str** - string
  - **float** - single precision, floating data point
  - **double** - double precision
  - **int** - integer
  - **list** - list
  - **dict** - dictionary

  For TSP only:

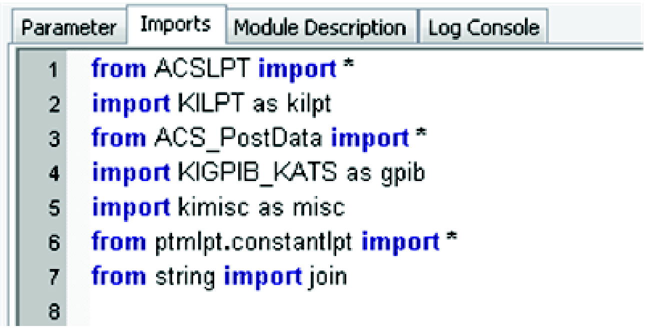
  - **table** - table type

- **Default**: specifies the default value for the input parameter
- **Min/valid tuple**: specifies the minimum recommended value for the input parameter (except list, dict, table, and str type). When the user module is used in ACS Basic, configuration of the PTM/STM with a parameter value smaller than the minimum value causes ACS Basic to display an out-of-range message.
- **Max**: specifies the maximum recommended value for the input parameter (except list, dict, table, and str type). When the user module is used in ACS Basic, configuration of the PTM/STM with a parameter value larger than the maximum value causes ACS Basic to display an out-of-range message.
- **Unit**: area to enter a label for the default, minimum, and maximum values.
- **Comment**: location where you can save important information about the input or output.

Imports tab

The Imports tab area only displays for a PTM. It lists the imported files used within the python user module. This area can be used to add import statements to the active (currently open) user module (see next figure).The code displayed in the figure (from ACS_PostData import*) is essential to the ACS Basic post data commands. It allows the PTM to return data to the module's Data tab and should be included in every test module. This tab can also be used to import the global statement.

Figure 187: Imports tab

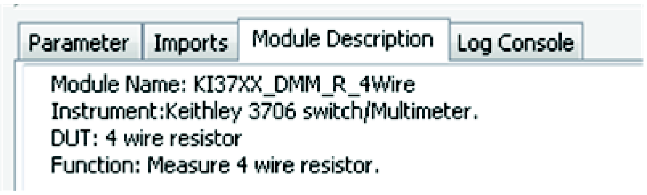


```
from ACSLPT import *
import KILPT as kilpt
from ACS_PostData import *
import KIGPIB_KATS as gpib
import kimisc as misc
from ptmlpt.constantlpt import *
from string import join
```

Module Description tab

The Module Description tab area allows you to enter descriptive information for the user module (see next figure).

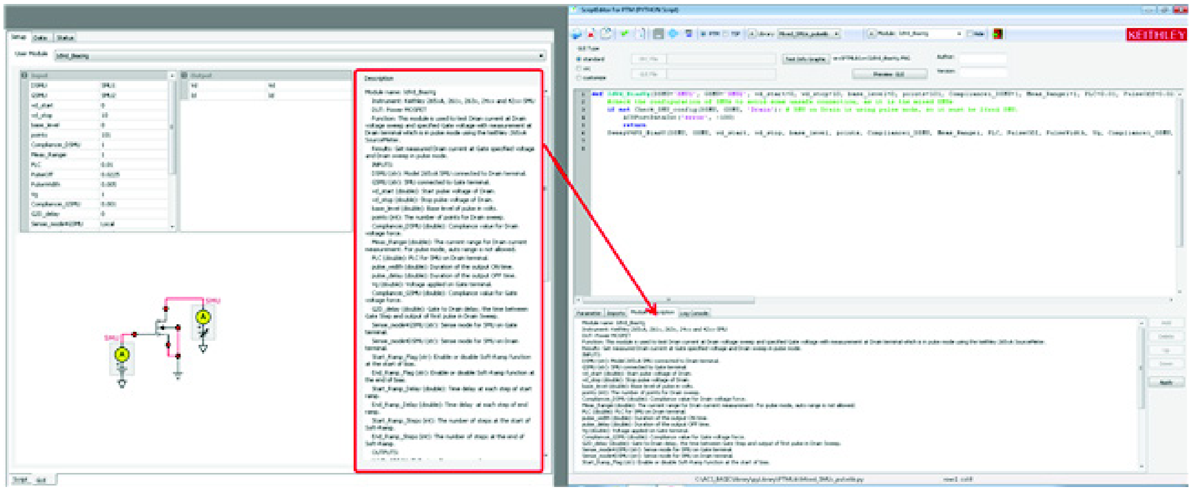
Figure 188: Module Description tab



Module Name: KI37XX_DMM_R_4Wire
Instrument:Keithley 3706 switch/Multimeter.
DUT: 4 wire resistor
Function: Measure 4 wire resistor.

Information entered in this area documents the module for the ACS Basic user and is used to create the user library help (see next figure).

Figure 189: Description tab in ACS Basic



NOTE

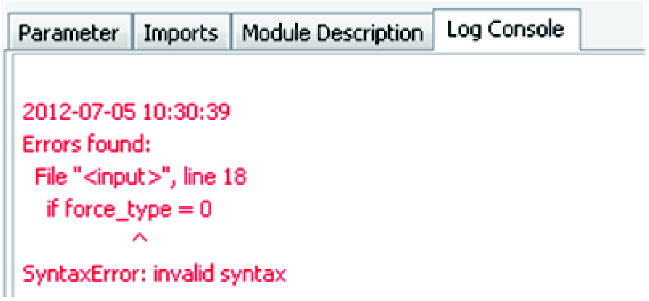
Do not use comment designators (--) in the Description tab area. When the user-module code is compiled, Script Editor also evaluates the text in this area. Code comment designators in the Description tab area can be misinterpreted which will cause errors.

Log Console tab

The Log Console tab area displays any error syntax messages that are generated during a code check operation.

When you click the **Check** icon ✓ on the toolbar, a message opens in the log console tab. When an error is displayed in the Log Console tab area, Script Editor shows the line of code where the error occurred or the next line, depending on how the compiler caught the error (see next figure). This facilitates error corrections. If no errors are found, the Build tab area displays a message stating that fact.

Figure 190: Log Console tab

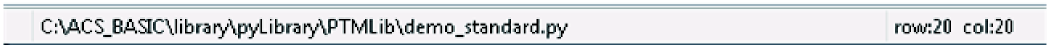


```
2012-07-05 10:30:39
Errors found:
  File "<input>", line 18
    if force_type = 0
                ^
SyntaxError: invalid syntax
```

The status bar

The Status bar, at the bottom of the Script Editor GUI, displays the module directory path and the cursor location in the code-entry area. For example, if the cursor is in row 42, column 4, in the code-entry area, the status bar indicates that information (see next figure).

Figure 191: Status bar

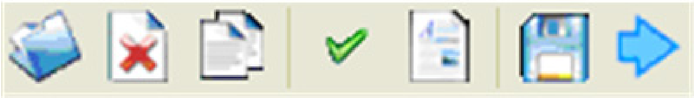


C:\ACS_BASIC\library\pyLibrary\PTMLib\demo_standard.py row:20 col:20

The toolbar

The icons on the toolbar (see next figure) are located on the top of the Script Editor GUI.

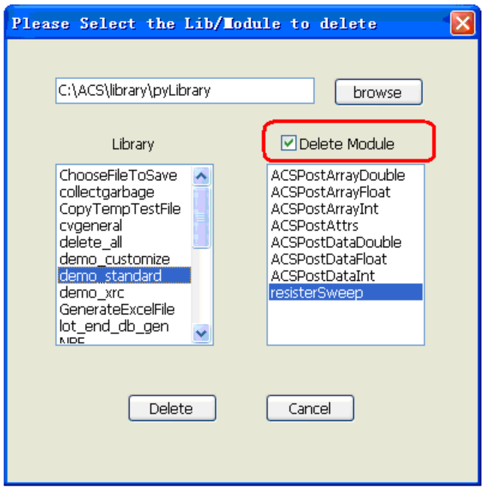
Figure 192: Script Editor Toolbar



The Toolbar icon descriptions:

- **Delete**: removes an existing user library or a user module from an existing library. Clicking the **Delete** icon opens the Please Select the Lib/Module to delete dialog box (see next figure). To delete a selected library and all of its contents, click the Delete button on the dialog box. To delete a highlighted test module from the selected library, select the **Delete Module** option on the dialog box, then click the Delete button.

Figure 193: Choose a module to delete



C:\ACS\library\pyLibrary

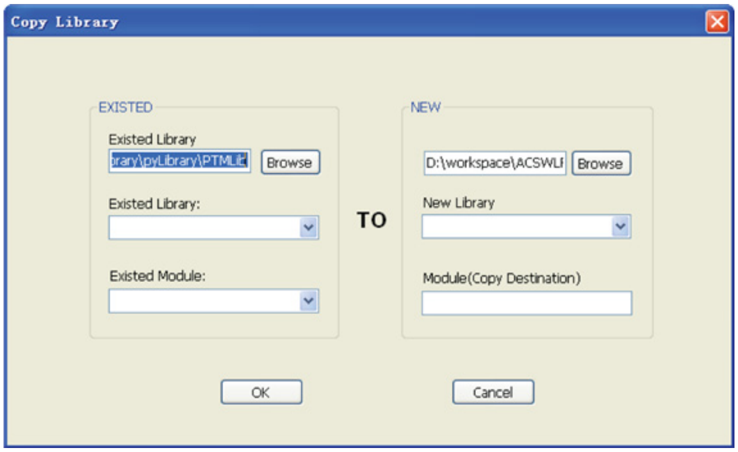
☑ Delete Module

ChooseFileToSave
collectgarbage
CopyTempTestFile
cvgeneral
delete_all
demo_customize
demo_standard
demo_xrc
GenerateExcelFile
lot_end_db_gen

ACSPostArrayDouble
ACSPostArrayFloat
ACSPostArrayInt
ACSPostAttrs
ACSPostDataDouble
ACSPostDataFloat
ACSPostDataInt
resisterSweep

- **Copy Library and Module**: creates a copy of any existing library and module. Clicking the **Copy Module and Library** icon opens the Copy Library dialog box. 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. Click **OK**. The existing library or module will be copied (see next figure)

Figure 194: Copy library and module



Existed Library
brary\pyLibrary\PTMLi

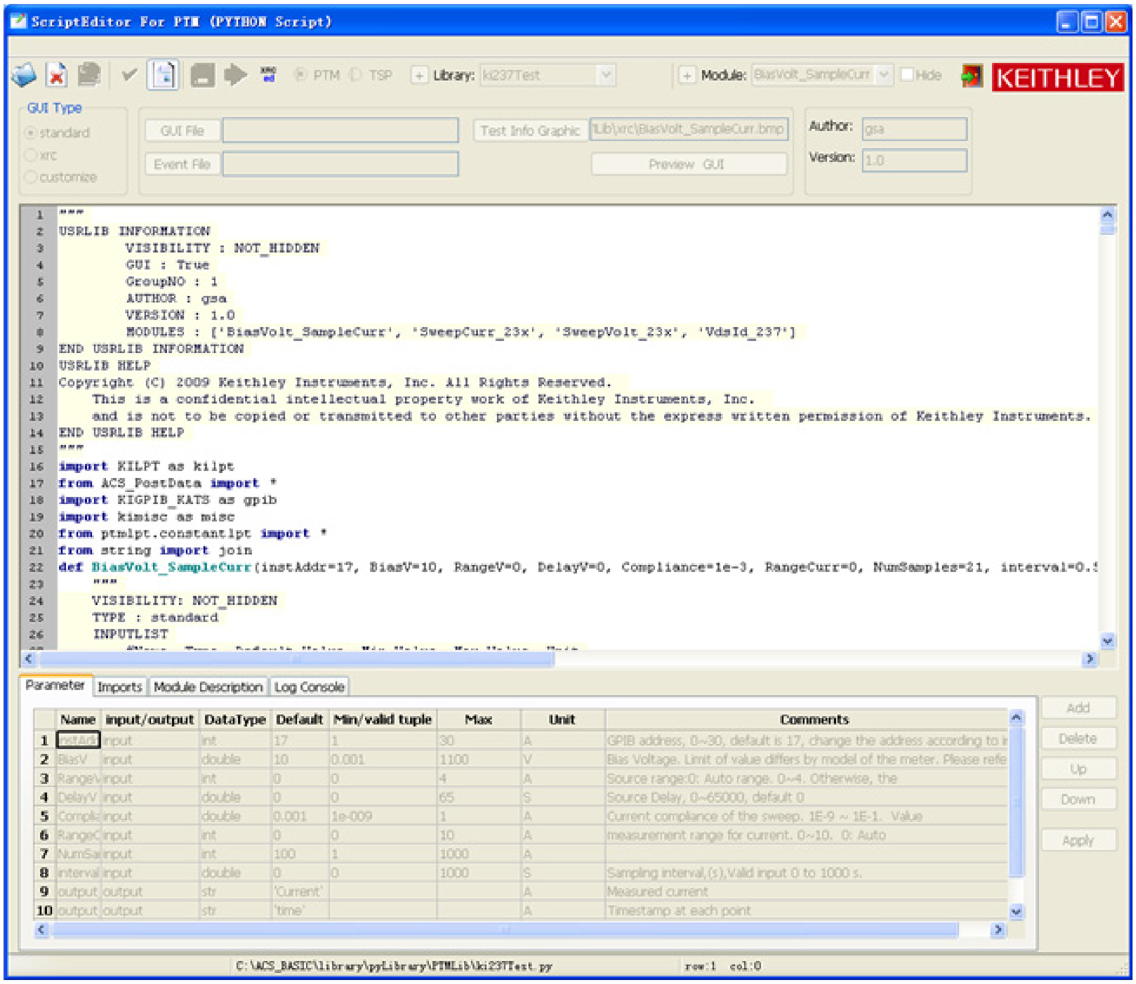
D:\workspace\ACSWLI

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 will be copied to the new library. The module name in the existing area can be empty (all of the modules in the source library will be copied to the destination library).

- **Check**: click this icon ✓ to check the source files of the open user module for syntax errors.
- **Browse or Edit the whole file**: click this icon to see the code-entry area that contains the whole file (note that the tab content is disabled)(see next figure). When you import a TSP or python source code (which is not generated by the Script Editor), it may start this function automatically if the file format does not match the Script Editor file format. You can use this feature to edit tests that were not created with structured inputs, outputs, and descriptions.

Figure 195: Browse the entire file



Library: ki237Test

Module: BiasVolt_SampleCurr

Test Info Graphic: \Lib\xrc\BiasVolt_SampleCurr.bmp

Author: gsa

Version: 1.0

```
"""
USRLIB INFORMATION
        VISIBILITY : NOT_HIDDEN
        GUI : True
        GroupNO : 1
        AUTHOR : gsa
        VERSION : 1.0
        MODULES : ['BiasVolt_SampleCurr', 'SweepCurr_23x', 'SweepVolt_23x', 'VdsId_237']
END USRLIB INFORMATION
USRLIB HELP
Copyright (C) 2009 Keithley Instruments, Inc. All Rights Reserved.
    This is a confidential intellectual property work of Keithley Instruments, Inc.
    and is not to be copied or transmitted to other parties without the express written permission of Keithley Instruments.
END USRLIB HELP
"""
import KILPT as kilpt
from ACS_PostData import *
import KIGPIB_KATS as gpib
import kimisc as misc
from ptmlpt.constantlpt import *
from string import join
def BiasVolt_SampleCurr(instAddr=17, BiasV=10, RangeV=0, DelayV=0, Compliance=1e-3, RangeCurr=0, NumSamples=21, interval=0.5
    """
    VISIBILITY: NOT_HIDDEN
    TYPE : standard
    INPUTLIST
```

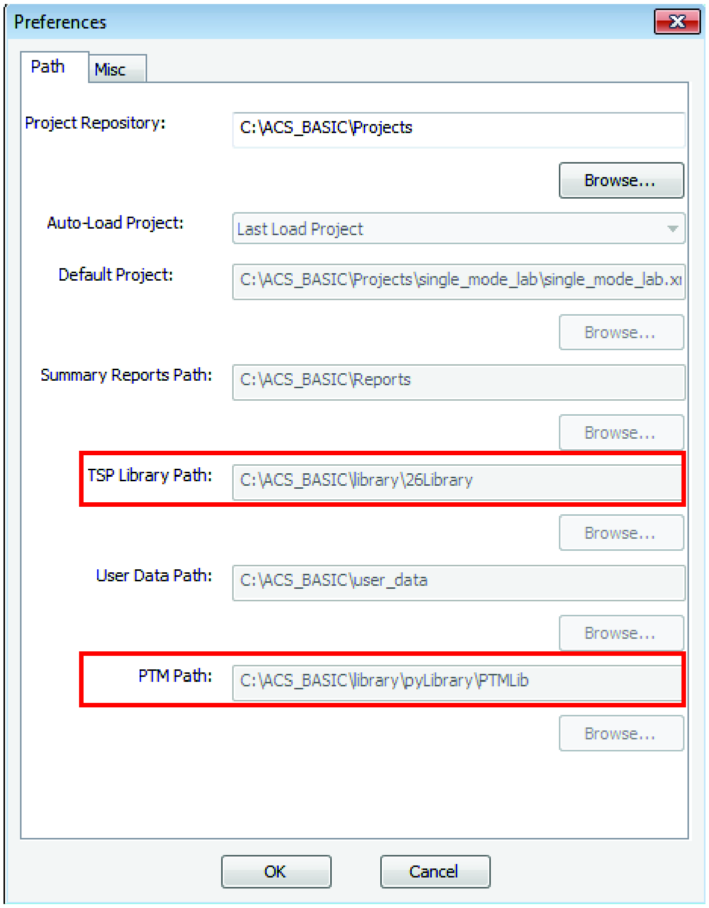
Parameter | Imports | Module Description | Log Console

	Name	input/output	DataType	Default	Min/valid tuple	Max	Unit	Comments
1	nstAd	input	int	17	1	30	A	GPIB address, 0~30, default is 17, change the address according to i
2	BiasV	input	double	10	0.001	1100	V	Bias Voltage. Limit of value differs by model of the meter. Please refe
3	RangeV	input	int	0	0	4	A	Source range:0: Auto range. 0~4. Otherwise, the
4	DelayV	input	double	0	0	65	S	Source Delay, 0~65000, default 0
5	Compli	input	double	0.001	1e-009	1	A	Current compliance of the sweep. 1E-9 ~ 1E-1. Value
6	RangeC	input	int	0	0	10	A	measurement range for current. 0~10. 0: Auto
7	NumSa	input	int	100	1	1000	A	
8	interval	input	double	0	0	1000	S	Sampling interval,(s),Valid input 0 to 1000 s.
9	output	output	str	'Current'			A	Measured current
10	output	output	str	'time'			A	Timestamp at each point

C:\ACS_BASIC\library\pyLibrary\PTMLib\ki237Test.py row:1 col:0

- **Save**: click this icon to save the opened user module source code. The default directory path is \\ACS\library\26Library\TSPLib for a TSP script and \\ACS\library\pyLibrary\PTMLib for a PTM. The path is assigned in the Preference setting dialog box (see next figure).

Figure 196: Choose the TSP and PTM path



Project Repository: C:\ACS_BASIC\Projects

Auto-Load Project: Last Load Project

Default Project: C:\ACS_BASIC\Projects\single_mode_lab\single_mode_lab.xı

Summary Reports Path: C:\ACS_BASIC\Reports

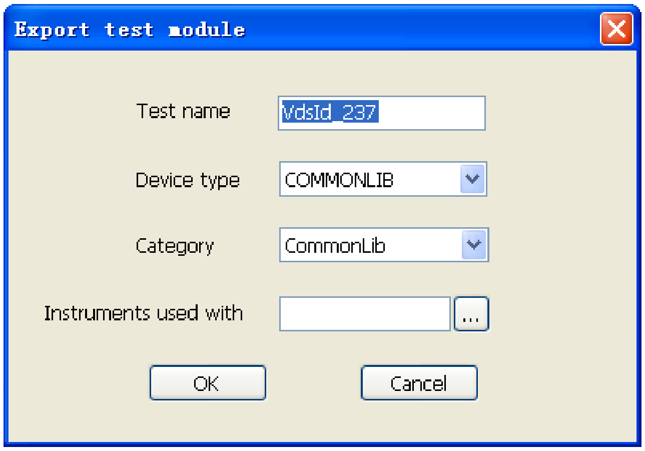
TSP Library Path: C:\ACS_BASIC\library\26Library

User Data Path: C:\ACS_BASIC\user_data

PTM Path: C:\ACS_BASIC\library\pyLibrary\PTMLib

Export: click this icon to open the Export test module dialog box. Enter the name in the Test name edit box; select the Device type and Category for the Test. Click **OK**. The test module will be exported and saved. The saved module is now available to load in ACS Basic.

Figure 197: Export test module dialog box



Test name VdsId_237

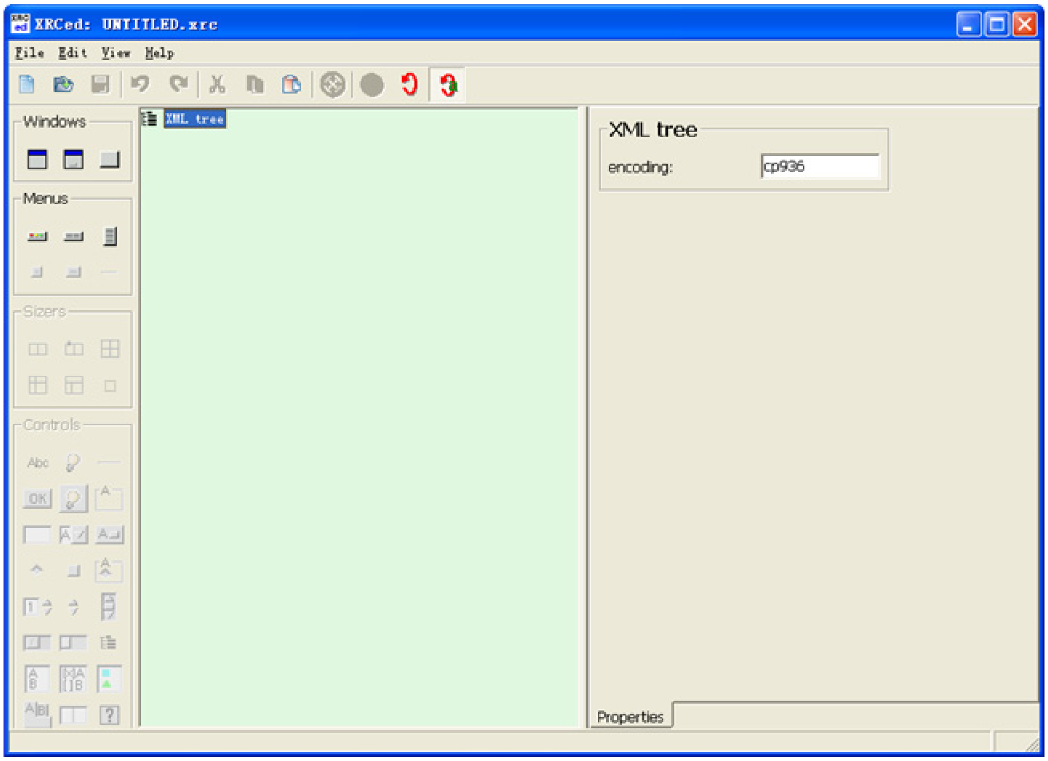
Device type COMMONLIB

Category CommonLib

Instruments used with

- **XRC Builder**: enables you to build GUIs for custom test modules written in either Python or TSP. The TSP scripts work with the Series 2600, Series 2600B, and Model 3706A instruments (see next figure). The Python scripts are used to communicate over GPIB to any of these instruments, as well as any general purpose hardware a user might want to control. Change the GUI Type to XRC to take advantage of these GUI design features (for more information see the Create a XRC GUI file topic).

Figure 198: XRC GUI builder



Script Editor tutorials

This section includes two tutorials. One is for creating and running a new TSP script and the other is for a PTM script. Each tutorial provides systematic instructions for accomplishing common tasks with the Script Editor.

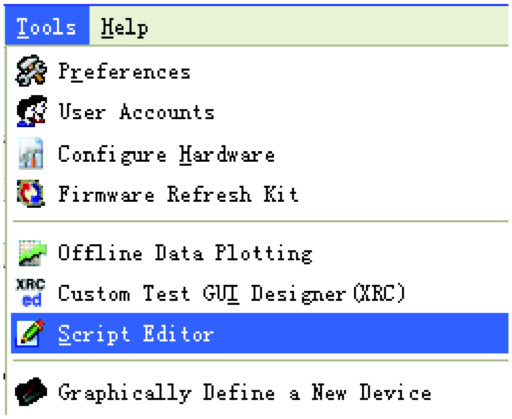
Tutorial 1: Create and run a new TSP script

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 test script processor (TSP) or python test module (PTM). For more information about TSP, refer to Multiple TSP instruments and for PTM refer to Configure a PTM.

This hands-on tutorial is provided to show you how to create a new user library and new user module.

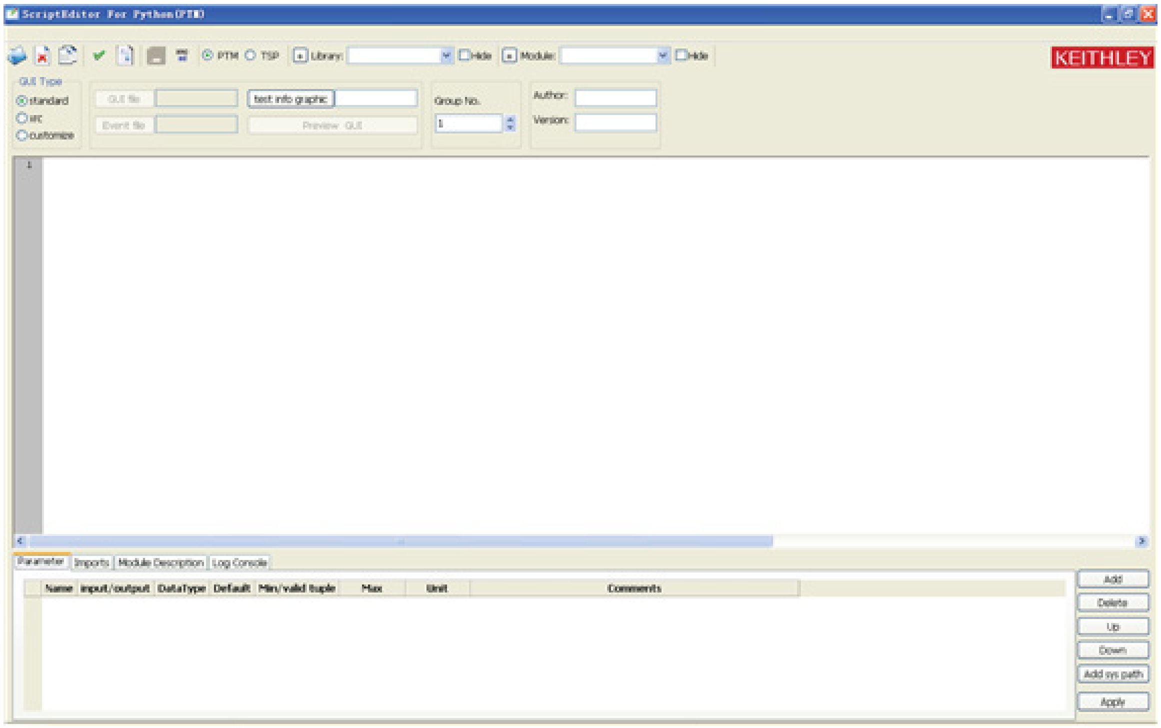
1. Open the Script Editor by selecting **Script Editor** in the Tools Menu (see next figure) or by clicking the Script editor icon on the vertical toolbar in the ACS Basic GUI.

Figure 199: Select Script Editor in the Tools menu



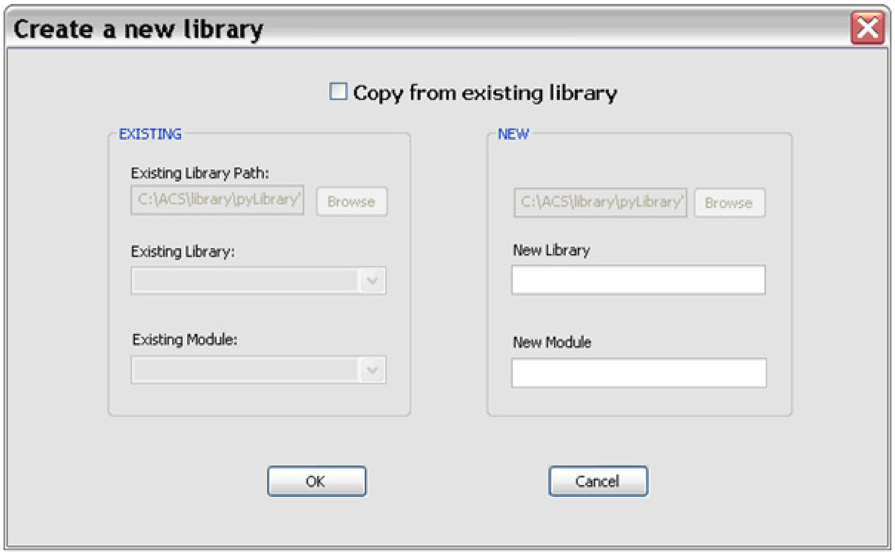
2. Select TSP in the module type area when script editor opens (see next figure).

Figure 200: Blank Script Editor GUI



3. Create a new user library by clicking the Add new library icon ⊞ Library:. The Create a new library dialog box opens (see next figure).

Figure 201: Create a new library dialog box

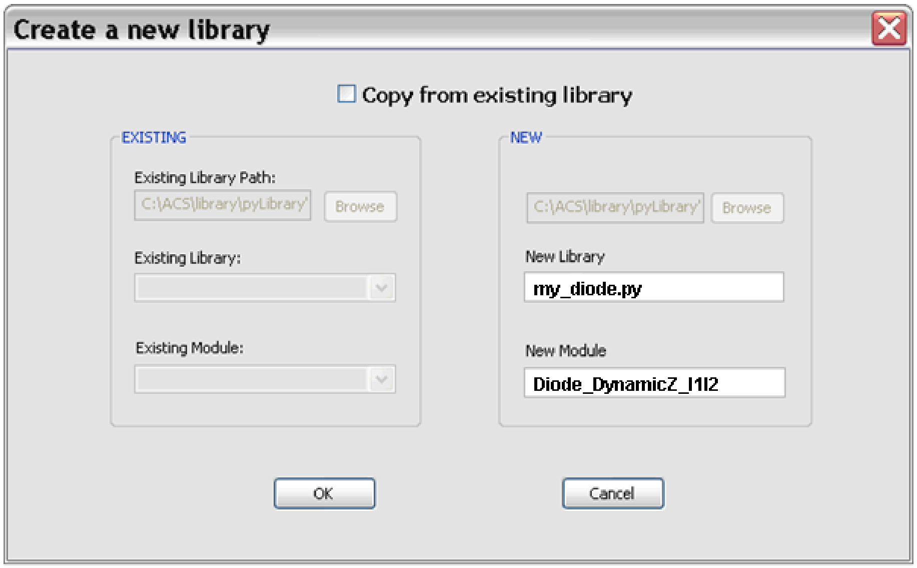


Existing Library Path: C:\ACS\library\pyLibrary\

C:\ACS\library\pyLibrary\

4. Enter the new user library name in the New Library edit box. The library name must have .py or .tsp extension. For this tutorial, enter `mylib1_Resistor.tsp`.
5. Enter `R_test1` as the New Module name (see next figure).

Figure 202: Enter new library and module name



Existing Library Path: C:\ACS\library\pyLibrary\

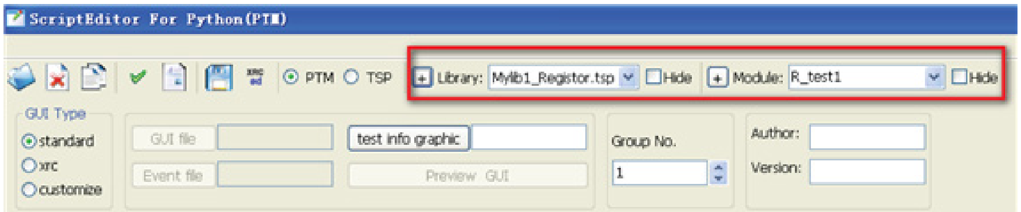
C:\ACS\library\pyLibrary\

New Library: my_diode.py

New Module: Diode_DynamicZ_I1I2

6. Click **OK**. The new library and module names will appear in the script editor GUI (see next figure).

Figure 203: New Library name in Script Editor GUI



ScriptEditor For Python(PTM)

● PTM ○ TSP

Library: Mylb1_Registor.tsp □Hide

Module: R_test1 □Hide

GUI Type: ● standard ○ xrc ○ customize

Group No. 1

Author:

Version:

7. Select a GUI for TSP:
   a. Select a GUI type. If selecting the XRC GUI type, you can select the **GUI File** and the **Event File**. For more information about a STM with a standard GUI, refer to Configure a script test module (STM). For more information about a STM with a XRC GUI, refer to the Create a XRC GUI file topic.
   b. Select a graphic for the TSP script by clicking on the **Test Info Graphic** button.
   c. Enter the **Author** and **Version** information.
8. Enter the required parameters for the code as follows:
   a. Click the **Parameter** tab (if the Parameter tab area is not displayed).
   b. Click the **Add** button.
   c. Enter the first parameter name (or accept the default). For the **mytsp1** user module, replace the default name with **RSMU1**.
   d. Specify the I/O of the first parameter in the input/output cell.

NOTE

For an output parameter, only the string data type is acceptable.

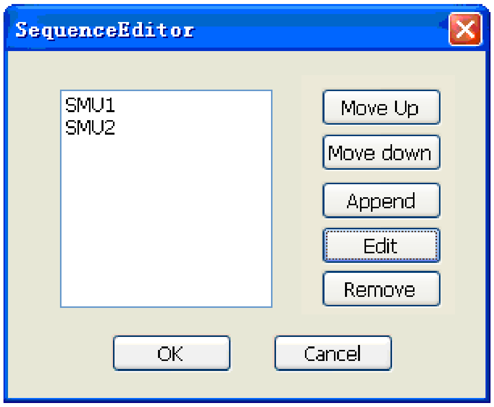
   e. Enter the data type for the first parameter. For this tutorial, enum was selected for the first parameter:

- **Int**: integer number
- **str**: string; can only be enclosed in single quotes ()
- **double**: 64 bit float point number
- **float**: 32 bit float point number
- **enum**: supplies an enumerator that can list the components of a composite moniker
- **table**: tables are created using table constructors, which are defined using curly brackets {}

   a. Enter the default, minimum, maximum, and unit values for the parameter:

- If the data type is enum, you should set the Min/valid tuple value first: Right-click the **Min** area, and the Sequence Editor dialog box opens (see next figure). Click the **Append** button.

Figure 204: Sequence Editor for setting the enumerable parameter

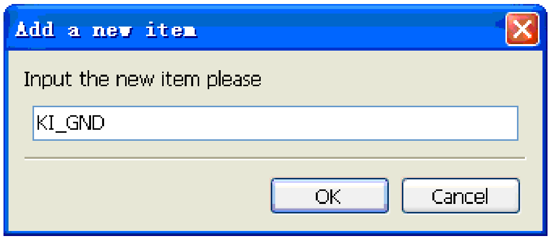


SequenceEditor

SMU1
SMU2

- Input the items in the Add a new item dialog box (see next figure). Click **OK**. The enumerable data is set.

Figure 205: Add a new item for setting the enumerable parameter



Input the new item please

KI_GND

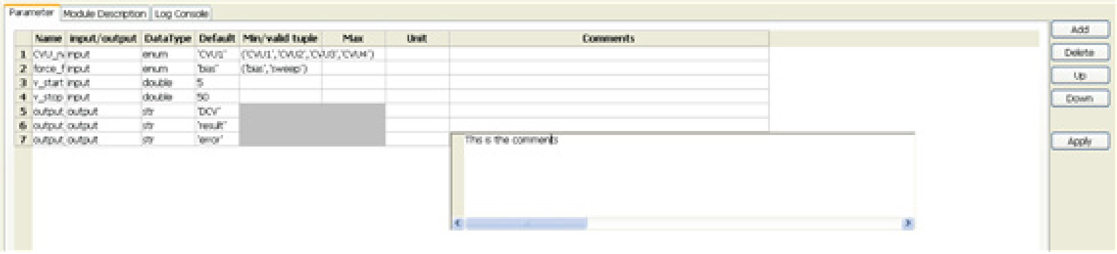
- For table type (often used in TSP), add the items in the table.
  - For the `mylib1` user module, enter `SMU1`, `SMU2`, and `KI_GND` for `RSMU1` enum data.

a. After the Min/valid tuple value is set, the default value can be selected. In the next figure, **SMU1** for **RSMU1** was selected.

b. Enter comments for the parameter, if needed. Click the comment area to display an edit box. Enter the parameter description (see next figure).

> **NOTE**
>
> When the user module code is compiled, Script Editor evaluates the text in the Module Description area. Therefore, do not use comment designators (--) in the Module Description tab area because the designators can be misinterpreted which can cause errors.

Figure 206: Enter comments

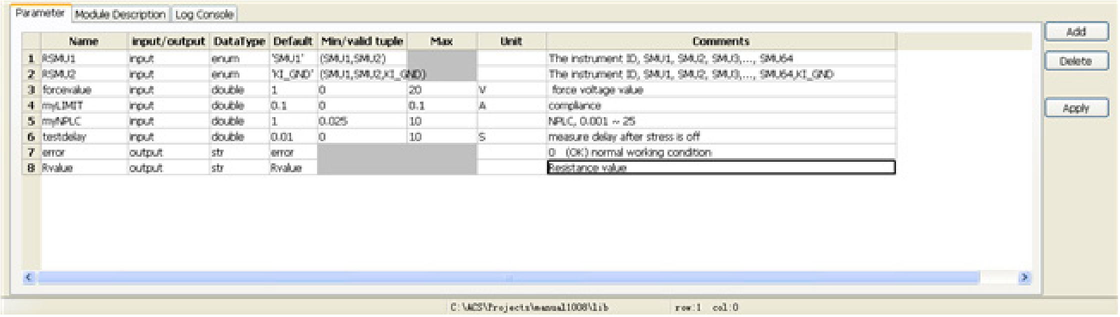


	Name	input/output	DataType	Default	Min/valid tuple	Max	Unit	Comments
1	CVU_n	input	enum	'CVU1'	('CVU1','CVU2','CVU3','CVU4')			
2	force_f	input	enum	'bias'	('bias','sweep')			
3	v_start	input	double	5				
4	v_stop	input	double	50				
5	output_	output	str	'DCV'				
6	output_	output	str	'result'				
7	output_	output	str	'error'				

This is the comments

c. Repeat steps b through g for of the all additional user module input and output parameters. For the `R_test1` module, add eight parameters (see next figure).

Figure 207: Parameter tab for TSP

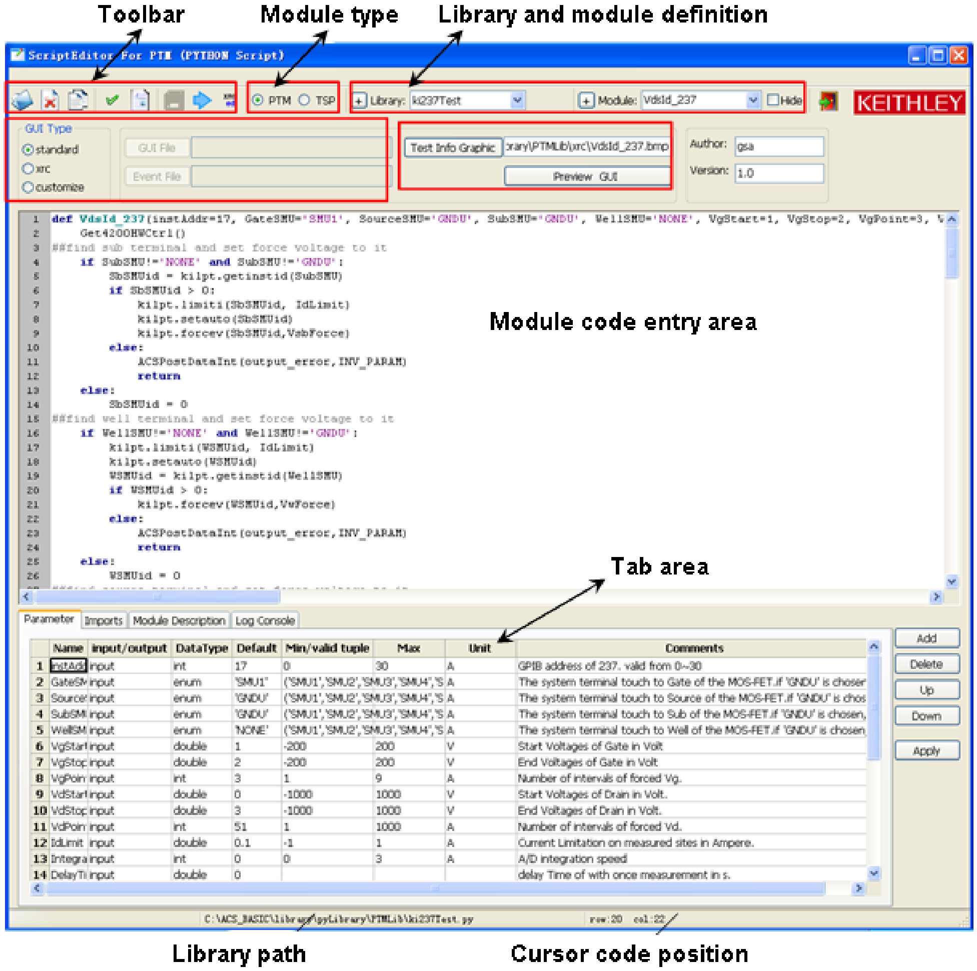


	Name	input/output	DataType	Default	Min/valid tuple	Max	Unit	Comments
1	RSMU1	input	enum	'SMU1'	(SMU1,SMU2)			The instrument ID, SMU1, SMU2, SMU3,..., SMU64
2	RSMU2	input	enum	'KI_GND'	(SMU1,SMU2,KI_GND)			The instrument ID, SMU1, SMU2, SMU3,..., SMU64,KI_GND
3	forcevalue	input	double	1	0	20	V	force voltage value
4	myLIMIT	input	double	0.1	0	0.1	A	compliance
5	myNPLC	input	double	1	0.025	10		NPLC, 0.001 ~ 25
6	testdelay	input	double	0.01	0	10	S	measure delay after stress is off
7	error	output	str	error				0 (OK) normal working condition
8	Rvalue	output	str	Rvalue				Resistance value

C:\ACS\Projects\manual1008\lib row:1 col:0

9. Click the **Apply** button. The parameters are displayed in the Parameter tab (see next figure).

Figure 208: Script Editor with parameters



Test Info Graphic: :rary\PTMLib\xrc\VdsId_237.bmp

Author: gsa

Version: 1.0

```
def VdsId_237(instAddr=17, GateSMU='SMU1', SourceSMU='GNDU', SubSMU='GNDU', WellSMU='NONE', VgStart=1, VgStop=2, VgPoint=3, V
    Get4200HWCtrl()
##find sub terminal and set force voltage to it
    if SubSMU!='NONE' and SubSMU!='GNDU':
        SbSMUid = kilpt.getinstid(SubSMU)
        if SbSMUid > 0:
            kilpt.limiti(SbSMUid, IdLimit)
            kilpt.setauto(SbSMUid)
            kilpt.forcev(SbSMUid,VsbForce)
        else:
            ACSPostDataInt(output_error,INV_PARAM)
            return
    else:
        SbSMUid = 0
##find well terminal and set force voltage to it
    if WellSMU!='NONE' and WellSMU!='GNDU':
        kilpt.limiti(WSMUid, IdLimit)
        kilpt.setauto(WSMUid)
        WSMUid = kilpt.getinstid(WellSMU)
        if WSMUid > 0:
            kilpt.forcev(WSMUid,VwForce)
        else:
            ACSPostDataInt(output_error,INV_PARAM)
            return
    else:
        WSMUid = 0
```

	Name	input/output	DataType	Default	Min/valid tuple	Max	Unit	Comments
1	instAdd	input	int	17	0	30	A	GPIB address of 237. valid from 0~30
2	GateSM	input	enum	'SMU1'	('SMU1','SMU2','SMU3','SMU4','S	A		The system terminal touch to Gate of the MOS-FET.if 'GNDU' is choser
3	Source	input	enum	'GNDU'	('SMU1','SMU2','SMU3','SMU4','S	A		The system terminal touch to Source of the MOS-FET.if 'GNDU' is chos
4	SubSM	input	enum	'GNDU'	('SMU1','SMU2','SMU3','SMU4','S	A		The system terminal touch to Sub of the MOS-FET.if 'GNDU' is chosen,
5	WellSM	input	enum	'NONE'	('SMU1','SMU2','SMU3','SMU4','S	A		The system terminal touch to Well of the MOS-FET.if 'GNDU' is chosen
6	VgStart	input	double	1	-200	200	V	Start Voltages of Gate in Volt
7	VgStop	input	double	2	-200	200	V	End Voltages of Gate in Volt
8	VgPoin	input	int	3	1	9	A	Number of intervals of forced Vg.
9	VdStart	input	double	0	-1000	1000	V	Start Voltages of Drain in Volt.
10	VdStop	input	double	3	-1000	1000	V	End Voltages of Drain in Volt.
11	VdPoin	input	int	51	1	1000	A	Number of intervals of forced Vd.
12	IdLimit	input	double	0.1	-1	1	A	Current Limitation on measured sites in Ampere.
13	Integra	input	int	0	0	3	A	A/D integration speed
14	DelayTi	input	double	0				delay Time of with once measurement in s.

C:\ACS_BASIC\library\pyLibrary\PTMLib\ki237Test.py

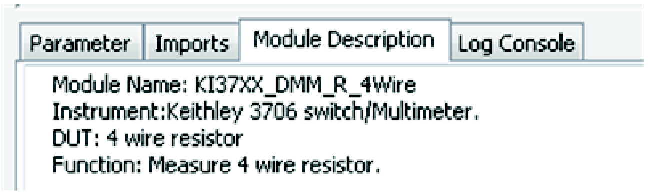
row:20 col:22

10. Enter the module description:
    a. Click the **Module Description** tab (see next figure).
    b. Enter any comments that are needed to document the user module.

NOTE

The comments that you include with the code in Script Editor cannot be viewed by other users of Script Editor.

Figure 209: Module Description tab



Module Name: KI37XX_DMM_R_4Wire
Instrument:Keithley 3706 switch/Multimeter.
DUT: 4 wire resistor
Function: Measure 4 wire resistor.

11. Enter the new TSP code to the module-code entry area. Refer to the **ACS Basic Libraries Reference manual** for a complete list of supported I/O and SMU commands.
12. For the **R_test1** user module, enter the following code:

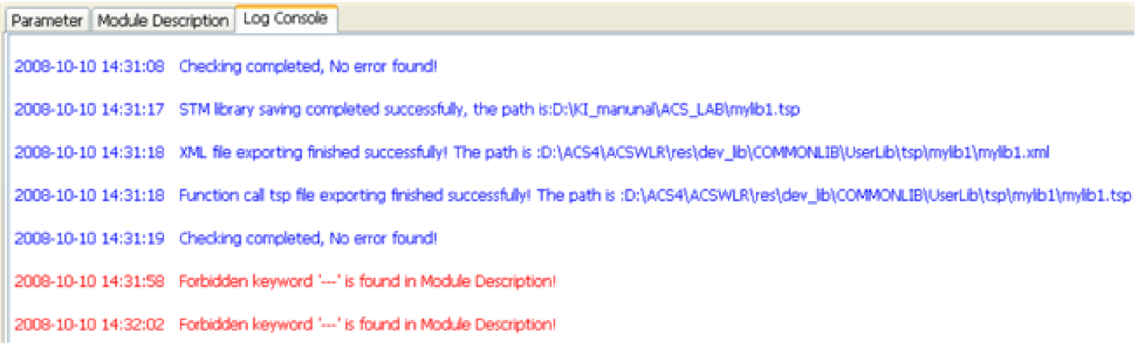
```
local v_value = {}
local i_value = {}
local error code = {}
local R = {}
setmode(RSMU1, KI_INTGPLC, myNPLC)
limiti(RSMU1, myLIMIT)
forcev(RSMU1, forcevalue)

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

delay(testdelay)
intgv(RSMU1, v_value)
intgi(RSMU1, i_value)

R[1] = v_value[1]/i_value[1]
posttable(Rvalue, R)
table.insert(error_code, 0)
posttable(error, error_code)

devint()
end
```

13. Compile the user module. For TSP, the Script Editor checks the user module code as follows:

    a. In the Toolbar, click the **Check** icon ✓ and 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 `mytsp1` user module, the Log console gives you instant feedback (see next figure).

Figure 210: The log console message tab



```
2008-10-10 14:31:08   Checking completed, No error found!
2008-10-10 14:31:17   STM library saving completed successfully, the path is:D:\KI_manunal\ACS_LAB\mylib1.tsp
2008-10-10 14:31:18   XML file exporting finished successfully! The path is :D:\ACS4\ACSWLR\res\dev_lib\COMMONLIB\UserLib\tsp\mylib1\mylib1.xml
2008-10-10 14:31:18   Function call tsp file exporting finished successfully! The path is :D:\ACS4\ACSWLR\res\dev_lib\COMMONLIB\UserLib\tsp\mylib1\mylib1.tsp
2008-10-10 14:31:19   Checking completed, No error found!
2008-10-10 14:31:58   Forbidden keyword '---' is found in Module Description!
2008-10-10 14:32:02   Forbidden keyword '---' is found in Module Description!
```

Make changes and continue checks until no errors exist

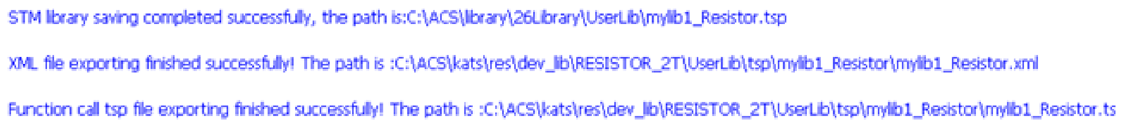
1. Save the user module:

    a. Click the **Save** icon and the source code information is saved to the following directory: C:\ACS\library\26Library\TSPLib.

    b. For example, in the next figure, after clicking the Save icon, the following files are saved:

- The source code file (.tsp). The default directory: C:\ ACS\library\26Library\TSPLib
- The .bmp file and .xrc file. The default directory: C:\ACS\library\26Library\TSPLib\xrc
- The events file. The default directory: C:\ACS\library\26Library\TSPLib\event

Figure 211: File save in the Log Console tab

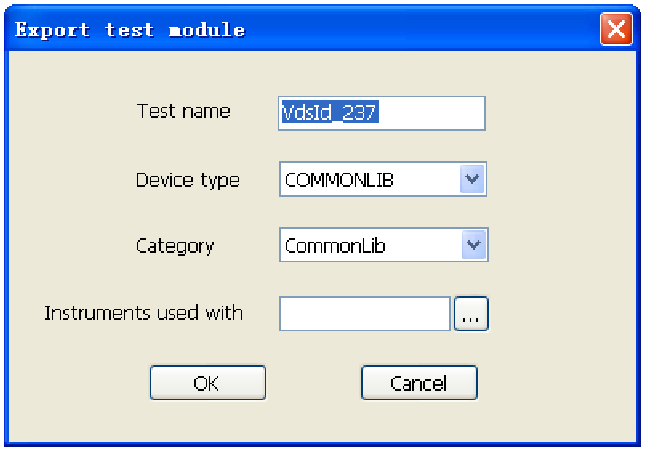


```
STM library saving completed successfully, the path is:C:\ACS\library\26Library\UserLib\mylib1_Resistor.tsp

XML file exporting finished successfully! The path is :C:\ACS\kats\res\dev_lib\RESISTOR_2T\UserLib\tsp\mylib1_Resistor\mylib1_Resistor.xml

Function call tsp file exporting finished successfully! The path is :C:\ACS\kats\res\dev_lib\RESISTOR_2T\UserLib\tsp\mylib1_Resistor\mylib1_Resistor.ts
```

2. Export the user module to the ACS Basic device library:

    a. Click the **Export** icon , and the Export test module dialog box opens (see next figure).

Figure 212: Export test module dialog box



Test name VdsId_237

Device type COMMONLIB

Category CommonLib

Instruments used with

    b. Enter a **Test name** in the edit box.

    c. Select a **Device type** using the drop-down arrow.

    d. Select a **Category** using the drop-down arrow.

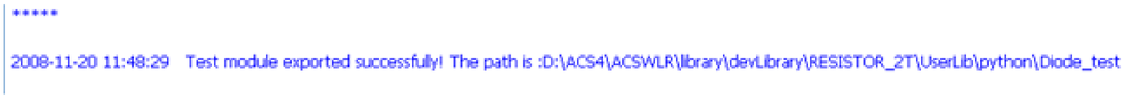
    e. Click the **ellipsis** so that you can choose an "Instrument used with" to export your test module for the library.

    f. Click **OK**.

The exported test module parameter files (.xml, .bmp, .tsp, .csv, and help file) are saved in the following directory: C:\ACS\library\devLibrary\Device_name\Category_name\language\test_module_name.

The exported information displays in the Log Console tab (see next figure).

Figure 213: Export test module Log Console tab



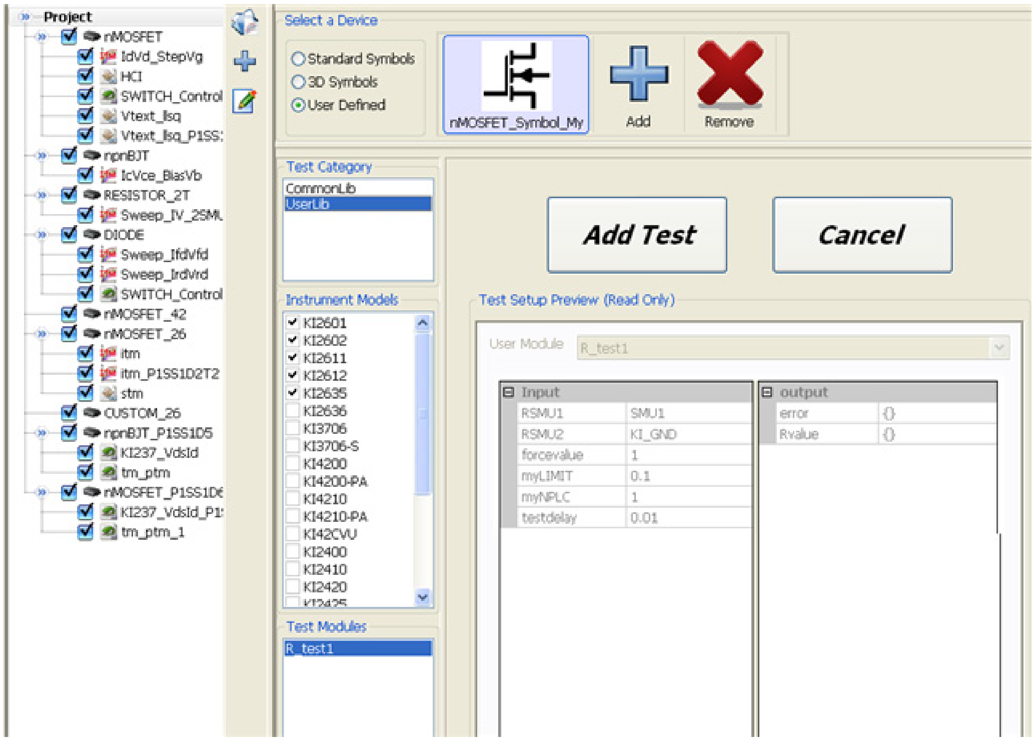
```
*****

2008-11-20 11:48:29   Test module exported successfully! The path is :D:\ACS4\ACSWLR\library\devLibrary\RESISTOR_2T\UserLib\python\Diode_test
```

The default directory path is: C:\ACS\library\devLibrary\RESISTOR_2T\UserLib\tsp\R_test. The new standard TSP script is created and saved in the default directory. You can open and run this module in ACS Basic.

Run the STM

1. Click the **Return to Library View** icon on the ACS Basic toolbar. If ACS Basic is in Multi-test mode, you must build a configuration navigator in order to add the test. For more information on how to build the configuration navigator in Multi-test mode, refer to the Build a project plan topic.
2. Select the device where the user module `R_test1` is saved. In the next figure, the device Resistor_2T is selected.
3. Select **UserLib** in the Test Category. The user module library opens in the Test modules area. Select the user module library that you want to test. In the next figure, `mylib1_Resistor` is selected.

Figure 214: Open the UserLib module



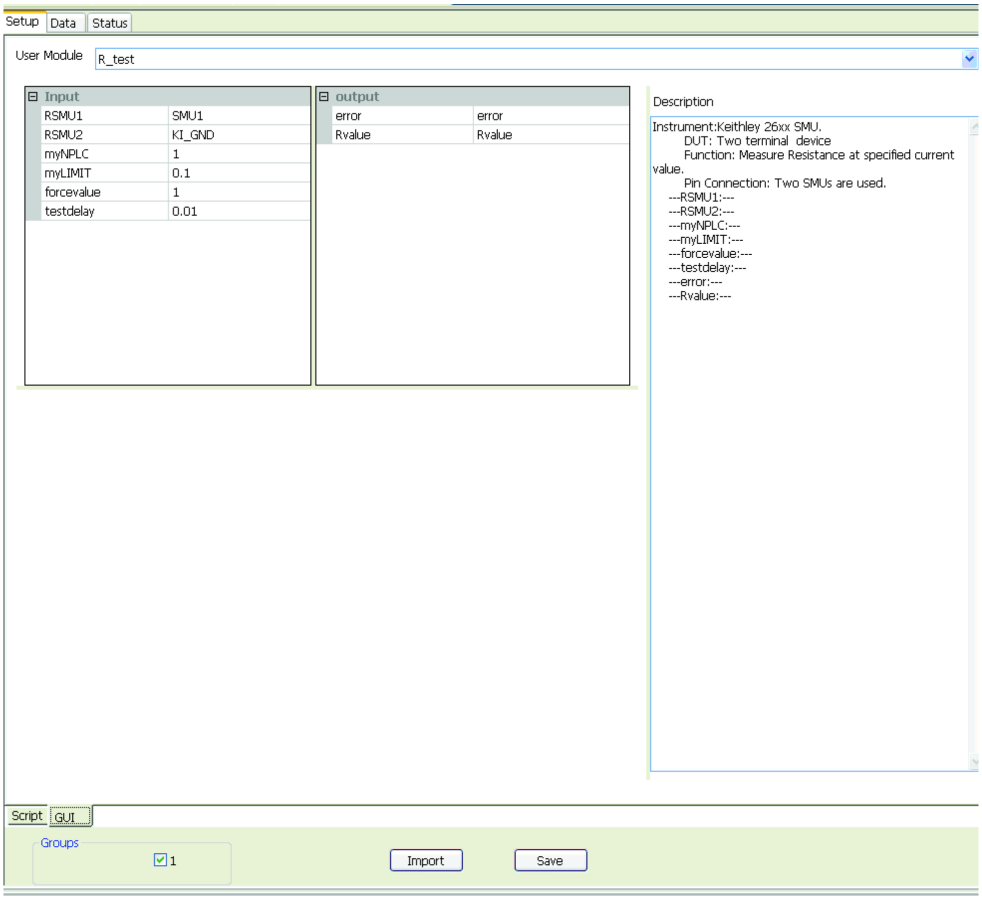
User Module: R_test1

Input	
RSMU1	SMU1
RSMU2	KI_GND
forcevalue	1
myLIMIT	0.1
myNPLC	1
testdelay	0.01

output	
error	{}
Rvalue	{}

4. Click the **Open Test** button (see next figure). Accept the default parameters for this example.

Figure 215: UserLib module opening



User Module R_test

⊟ Input	
RSMU1	SMU1
RSMU2	KI_GND
myNPLC	1
myLIMIT	0.1
forcevalue	1
testdelay	0.01

⊟ output	
error	error
Rvalue	Rvalue

Description

```
Instrument:Keithley 26xx SMU.
      DUT: Two terminal  device
      Function: Measure Resistance at specified current
value.
      Pin Connection: Two SMUs are used.
   ---RSMU1:---
   ---RSMU2:---
   ---myNPLC:---
   ---myLIMIT:---
   ---forcevalue:---
   ---testdelay:---
   ---error:---
   ---Rvalue:---
```

☑ 1

5. Save the STM and the project by clicking the **Save All** button.
6. Run the STM by clicking the Run icon ▶.
7. If the user module that you created generates data, check the execution results in the STM Data worksheet. To view the Data worksheet, click the **Data** tab.

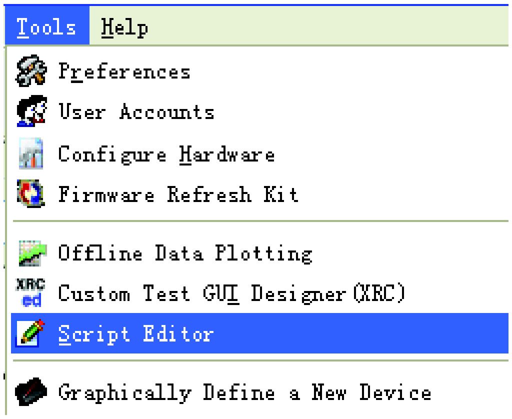
Tutorial 2: Create and run a new PTM

This hands-on tutorial is provided to show you how to create a new PTM. For more information on PTMs, refer to Configure a PTM.

Create a new PTM

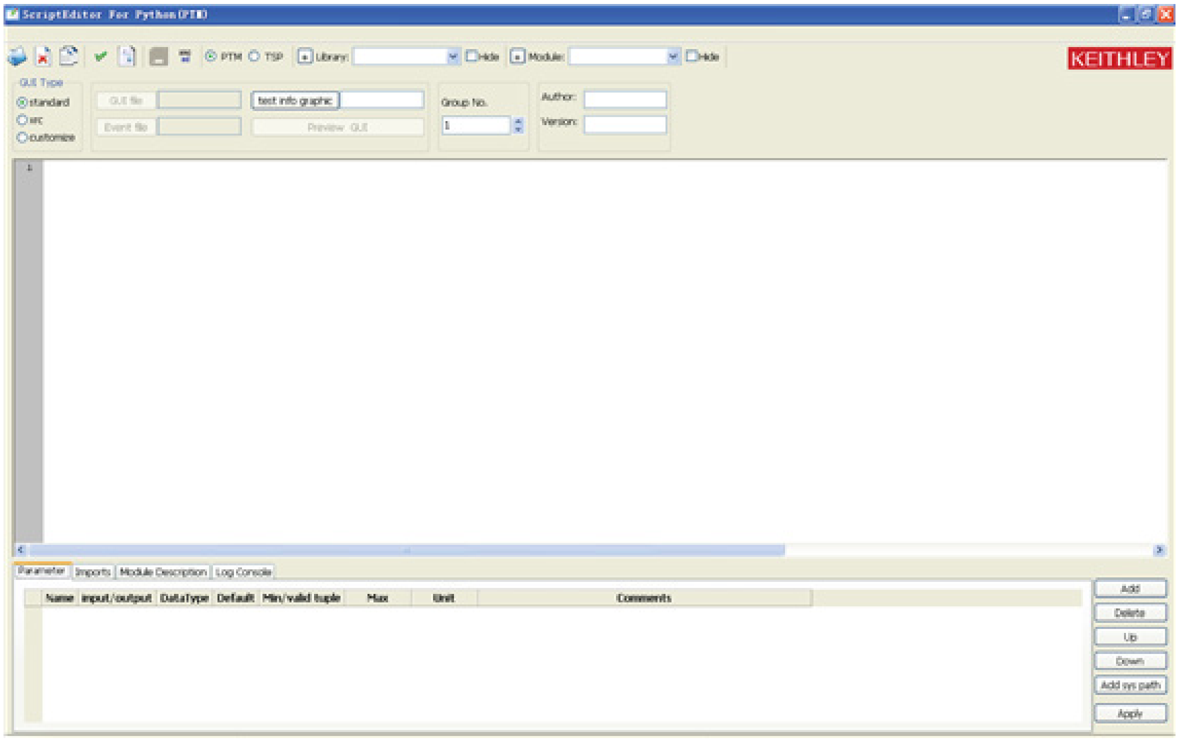
1. Open the Script Editor by selecting **Script Editor** in the Tools Menu (see next figure) or by clicking the Script editor icon on the vertical toolbar in the ACS Basic GUI.

Figure 216: Select Script Editor in the Tools menu



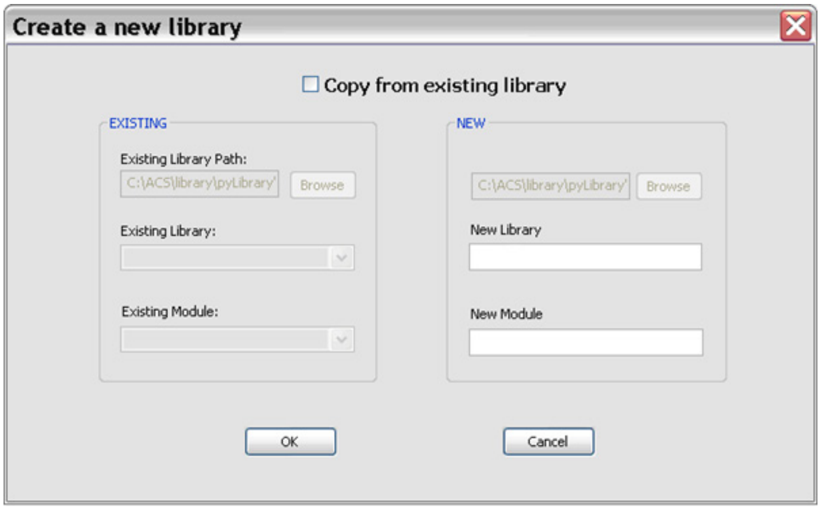
2. Select PTM in the module type area when script editor opens (see next figure).

Figure 217: Blank Script Editor GUI



3. Create a new user library by clicking the Add a new library icon ⊞ Library:. The Create a new library dialog box opens (see next figure).

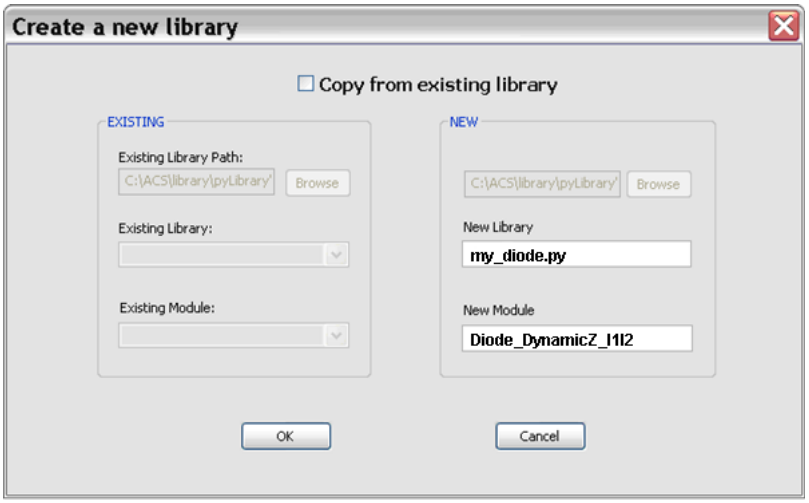
Figure 218: Create a new library dialog box



4. Enter the new user library name in the New Library edit box. The library name must have .py or .tsp extension. For this tutorial, enter `my_diode.py`.

5. Enter `Diode_DynamicZ_I1I2` as the New Module name (see next figure).

Figure 219: Enter new library and module name

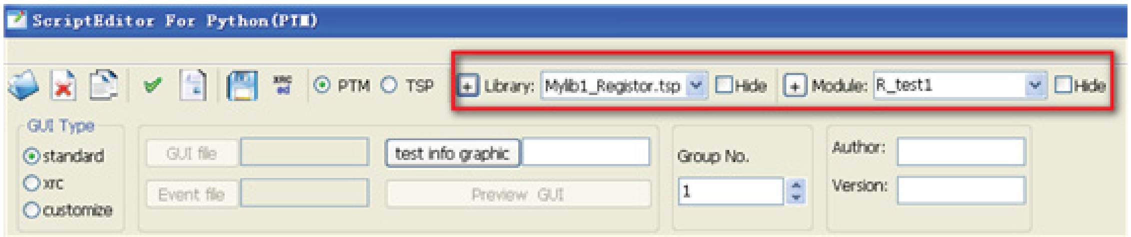


New Library: my_diode.py

New Module: Diode_DynamicZ_I1I2

6. Click **OK**. The new library and module names will appear in the script editor GUI (see next figure).

Figure 220: New Library name in Script Editor GUI



⊞ Library: Mylib1_Registor.tsp

⊞ Module: R_test1

Group No. 1

Author:

Version:

7. Select a GUI for PTM:
    a. Select the **standard** GUI type. If selecting the XRC GUI type, you can click on the **GUI File** and the **Event File** buttons. For more information on the STM with XRC GUI, refer to the Configure an XRC STM topic.
    b. Select a graphic for the PTM script by clicking on the **Test Info Graphic** button.
    c. Enter the **Author** and **Version** information in the edit boxes.
8. Enter the required parameters for the code as follows:
    a. Click the **Parameter** tab (if the Parameter tab area is not displayed).
    b. Click the **Add** button.
    c. Enter the first parameter name (or accept the default). For the **mytsp1** user module, replace the default name with **RSMU1**.
    d. Specify the I/O of the first parameter in the input/output cell.

NOTE

The string data type is only valid for output parameters.

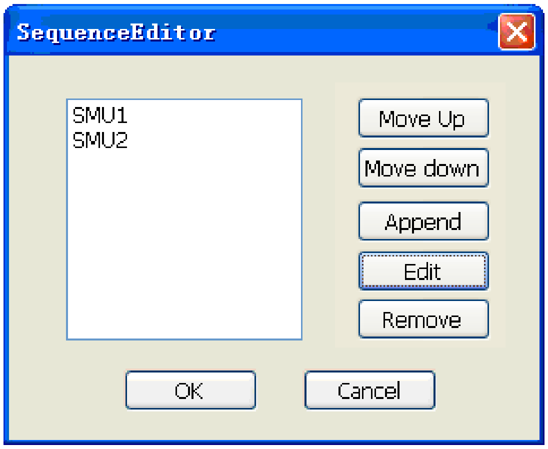
e. Enter the data type for the first parameter. For this tutorial, enum was selected for the first parameter:

- **Int**: integer number
- **str**: string; can only be enclosed in single quotes ()
- **double**: 64 bit float point number
- **float**: 32 bit float point number
- **enum**: supplies an enumerator that can list the components of a composite moniker
- **list**: can be written as a list of comma-separated values (.csv); it's not necessary to have all the same type and are defined using square brackets [ ].
- **dict**: defines one-to-one relationships between keys and values; first, you create a new dictionary with two elements and assign it to the variable; each element is a key-value pair, and the whole set of elements is enclosed in curly brackets { }.

f. Enter the default, minimum, maximum, and unit values for the parameter:

- If the data type is enum, you should set the Min/valid tuple value first: Click the **Min** area, and the Sequence Editor dialog box opens (see next figure). Click the **Append** button.

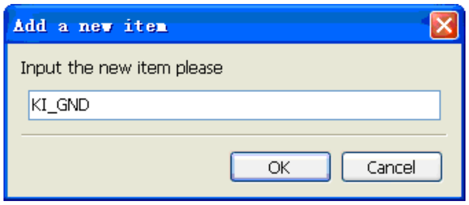
Figure 221: Sequence Editor for setting the enumerable parameter



SMU1
SMU2

- Input the items in the Add a new item dialog box (see next figure). Click **OK**. The enumerable data is set.
- If the data type is a list, set the default value as follows: Select the default value (the Sequence Editor dialog box opens). Add the item and then Click **OK**. The value will appear in the default box with this format [a,b,c].

Figure 222: Add a new item for setting the enumerable parameter



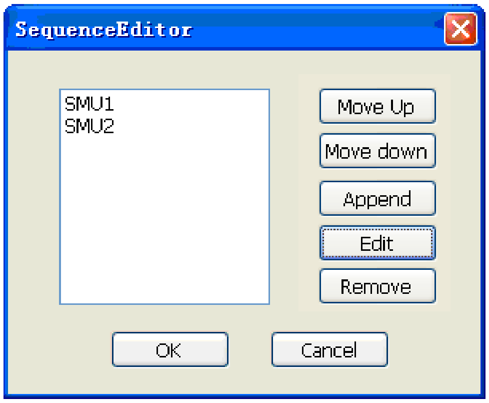
Add a new item

Input the new item please

KI_GND

- For dict type (dictionary type, only used in PTM): Select the default value and the Sequence Editor opens (see next figure).

Figure 223: Sequence Editor for setting the enumerable parameter

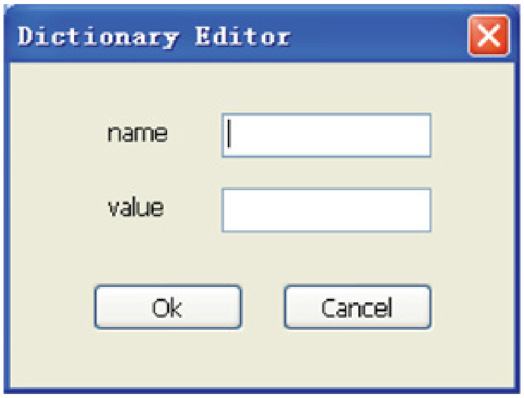


SequenceEditor

SMU1
SMU2

- Click the **Append** function and the Dictionary Editor dialog box opens (see next figure). Input the name and value in the Sequence Editor. The value will be formatted as {a:1,b:2,c:3}.

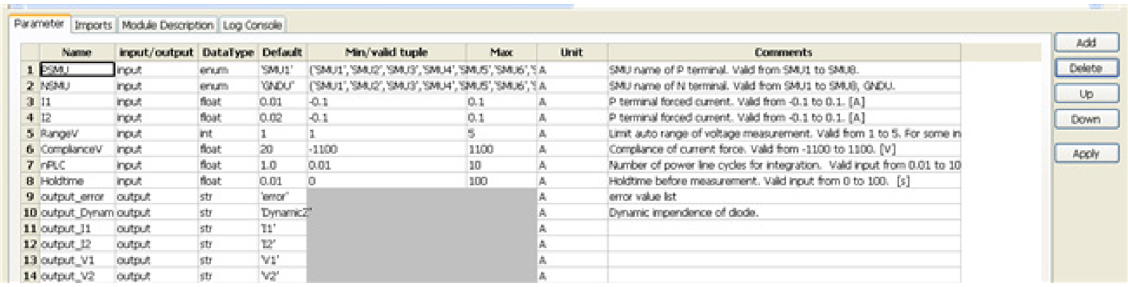
Figure 224: Dictionary Editor dialog box



Dictionary Editor

9. The next figure shows the PTM parameters.

Figure 225: Enter parameters for PTM

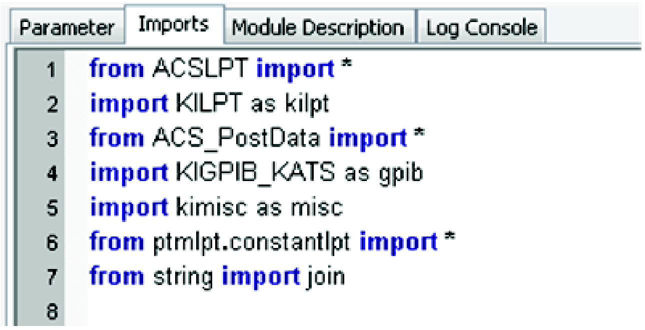


	Name	input/output	DataType	Default	Min/valid tuple	Max	Unit	Comments
1	PSMU	input	enum	'SMU1'	('SMU1','SMU2','SMU3','SMU4','SMU5','SMU6','		A	SMU name of P terminal. Valid from SMU1 to SMU8.
2	NSMU	input	enum	'GNDU'	('SMU1','SMU2','SMU3','SMU4','SMU5','SMU6','		A	SMU name of N terminal. Valid from SMU1 to SMU8, GNDU.
3	I1	input	float	0.01	-0.1	0.1	A	P terminal forced current. Valid from -0.1 to 0.1. [A]
4	I2	input	float	0.02	-0.1	0.1	A	P terminal forced current. Valid from -0.1 to 0.1. [A]
5	RangeV	input	int	1	1	5	A	Limit auto range of voltage measurement. Valid from 1 to 5. For some in
6	ComplianceV	input	float	20	-1100	1100	A	Compliance of current force. Valid from -1100 to 1100. [V]
7	nPLC	input	float	1.0	0.01	10	A	Number of power line cycles for integration. Valid input from 0.01 to 10
8	Holdtime	input	float	0.01	0	100	A	Holdtime before measurement. Valid input from 0 to 100. [s]
9	output_error	output	str	'error'			A	error value list
10	output_Dynam	output	str	'DynamicZ'			A	Dynamic impendence of diode.
11	output_I1	output	str	'I1'			A	
12	output_I2	output	str	'I2'			A	
13	output_V1	output	str	'V1'			A	
14	output_V2	output	str	'V2'			A	

10. Import other libraries to use in the current library (if desired).

    a. Click on the **Imports** tab. The Imports tab area opens (see next two figures).

Figure 226: Imports tab



```
from ACSLPT import *
import KILPT as kilpt
from ACS_PostData import *
import KIGPIB_KATS as gpib
import kimisc as misc
from ptmlpt.constantlpt import *
from string import join
```

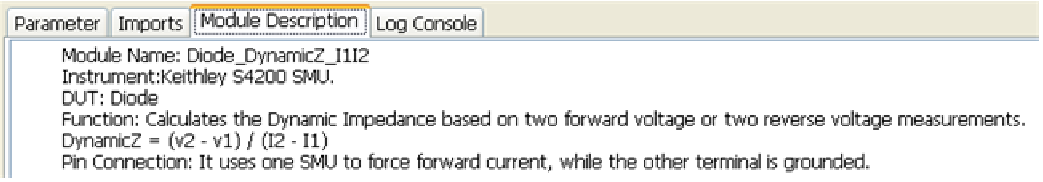
    b. Enter any additional import files that are needed for the user module.

11. Enter a Module Description:

    a. Click the **Module Description** tab.

    b. Enter needed text to document the user module. A Script Editor user will not see the comments that you include with the code (see next figure).

Figure 227: Module Description tab area DynamicZ



```
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.
```

NOTE

When the user module code is compiled, Script Editor evaluates the text in the Module Description area. Therefore, do not use comment designators (---) in the Module Description tab area because the designators can be misinterpreted which can cause errors.

12. Click the **Apply** button.

13. Enter the new PTM code to the module-code entry area. Refer to the ACS Basic Libraries Reference manual for a complete list of supported I/O and SMU commands.

14. For the `DynamicZ` user module, enter the following code:

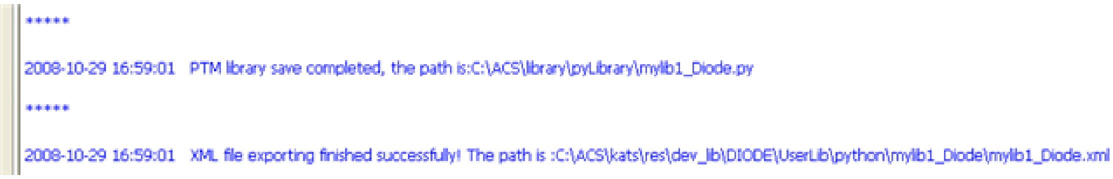
```
def Diode_DynamicZ_I1I2(PSMU='SMU1', NSMU='GNDU', I1=0.01, I2=0.02, RangeV=1,
ComplianceV=20, nPLC=1.0, Holdtime=0.01, output_error='error',
output_DynamicZ='DynamicZ', output_I1='I1', output_I2='I2', output_V1='V1',
output_V2='V2'):
   Get4200HWCtrl()
   error = [ ]
   DynamicZ = 0.0
   #tstsel()
   #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":
   #if NSMUId != 4096:
          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":
    #if NSMUId != 4096:
           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
```

15. Compile the user module.
16. Save the user module.
    a. Click the **Save** icon 💾; the source code information will be saved to the following directory: C:\ACS\library\pyLibrary\PTMLib.
    b. For example, in the next figure, after clicking **Save**, the following files are saved:

- The source code file (.py file). The default directory: C:\ACS\library\pyLibrary\PTMLib
- The .bmp file. The default directory: C:\ACS\library\pyLibrary\PTMLib\xrc
- The event file. The default directory: C:\ACS\library\pyLibrary\PTMLib\event

Figure 228: File save in the Log Console PTM

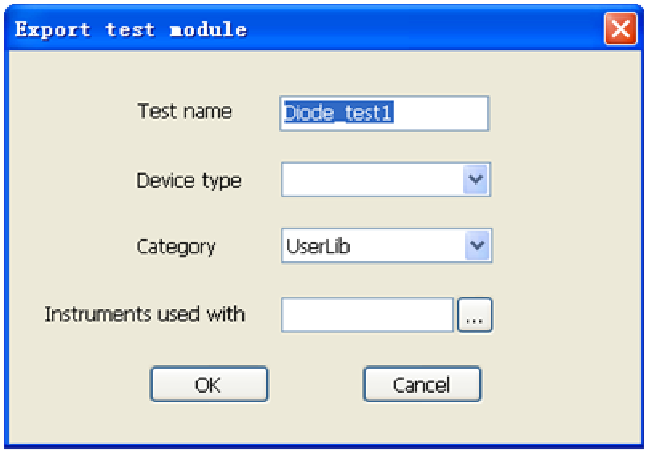


```
*****
2008-10-29 16:59:01   PTM library save completed, the path is:C:\ACS\library\pyLibrary\mylib1_Diode.py
*****
2008-10-29 16:59:01   XML file exporting finished successfully! The path is :C:\ACS\kats\res\dev_lib\DIODE\UserLib\python\mylib1_Diode\mylib1_Diode.xml
```

17. Export the user module:

    a. Click the **Export** icon, and the Export test module dialog box opens (see next figure).

Figure 229: Export test module PTM

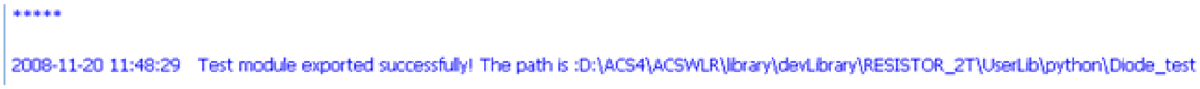


    b. Enter a **Test name** in the edit box.
    c. Select a **Device type** using the drop-down arrow.
    d. Select a **Category** using the drop-down arrow.
    e. Click the **ellipsis** so that you can choose an "Instrument used with" to export your test module for the library.
    f. Click **OK**.

The exported test module parameter files (.xml, .bmp, .tsp, .csv, and help file) are saved in the following directory:
C:\ACS\library\devLibrary\Device_name\Category_name\language\test_module_name.

The exported information displays in the Log Console tab (see next figure).

Figure 230: Export PTM information



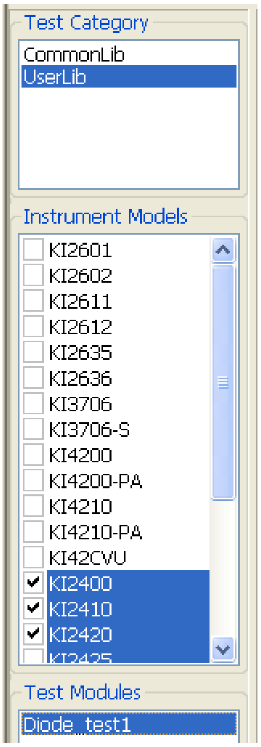
```
*****
2008-11-20 11:48:29   Test module exported successfully! The path is :D:\ACS4\ACSWLR\library\devLibrary\RESISTOR_2T\UserLib\python\Diode_test
```

The default directory path is: C:\ACS\library\devLibrary\RESISTOR_2T\UserLib\ptm\Diode_test1. The new standard PTM script is created and saved in the default directory. You can open and run this module in ACS Basic.

Run the PTM

1. Click the **Return to Library View** icon on the ACS Basic toolbar. If ACS Basic is in Multi-test mode, you must build a configuration navigator in order to add the test. For more information on how to build the configuration navigator in Multi-test mode, refer to the Build a project plan topic.
2. Select the device where the user module `Diode_test1` is saved.
3. Select **UserLib** in the Test Category. The user module library opens in the Test modules area. Select the user module library that you want to test. In the next figure, `mylib1_Diode` is selected.

Figure 231: Open UserLib in Test Category



Test Category

CommonLib
UserLib

Instrument Models

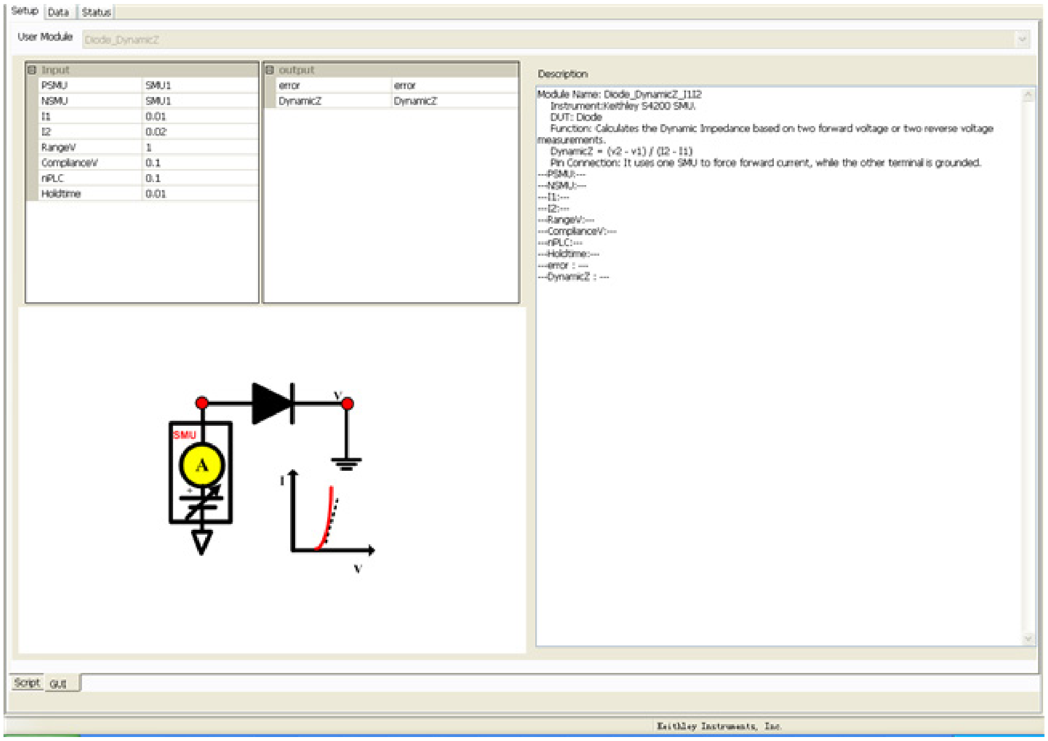
☐ KI2601
☐ KI2602
☐ KI2611
☐ KI2612
☐ KI2635
☐ KI2636
☐ KI3706
☐ KI3706-S
☐ KI4200
☐ KI4200-PA
☐ KI4210
☐ KI4210-PA
☐ KI42CVU
☑ KI2400
☑ KI2410
☑ KI2420
☐ KI2425

Test Modules

Diode_test1

4. Click the **Open Test** button (see next figure). Accept the default parameters for this example.

Figure 232: PTM UserLib module



User Module: Diode_DynamicZ

Input	
PSMU	SMU1
NSMU	SMU1
I1	0.01
I2	0.02
RangeV	1
ComplianceV	0.1
nPLC	0.1
Holdtime	0.01

output	
error	error
DynamicZ	DynamicZ

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:---
---NSMU:---
---I1:---
---I2:---
---RangeV:---
---ComplianceV:---
---nPLC:---
---Holdtime:---
---error : ---
---DynamicZ : ---
```

5. Save the PTM and the project by clicking the **Save All** button.
6. Run the PTM by clicking the Run icon ▶.
7. If the user module that you created generates data, check the execution results in the PTM Data worksheet. To view the Data worksheet, click the **Data** tab.

Section 8

Formulator function reference

In this section:

ABS

Purpose: Calculates the absolute value of each value in the designated column (vector) or the absolute value of any operand.

Format: ABS(X)

X = The name of any column (vector) listed under Columns or any operand.

Example: F2 = ABS(I_GATE)

Remarks: This function can be used to perform calculations in realtime, while a test is executing.

AT

Purpose: Extracts and returns a single value from a column (vector).1

Format: AT(V, POS)

V = The name of any column (vector) listed under Columns.

POS = The row number of column V where the single value is located.

Example: IDSAT = AT(I_DRAIN, 36)

AVG

Purpose: Returns the average of all values in the column (vector).

Format: AVG(V)

V = The name of any column (vector) listed under Columns.

Example: LEAKAGE = AVG(I_GATE)

CURRENT NOISE

Purpose: This formula is used to calculate noise current from Ig.

Format: CURRENT_NOISE(IMEAS,POINT_NUM)

IMEAS = measured current

POINT_NUM = The noise points number for calculate.

Example: Ig_noise=CURRENT_NOISE(Ig,50)

DELTA

Purpose: Returns the differences between the adjacent values in a column (vector). That is, for column V, DELTA returns (V2 - V1), (V3 -V2), etc.

Format: DELTA(V)

V = The name of any column (vector) listed under Columns.

Example: GM = DELTA(I_DRAIN)/DELTA(V_GATE)

Remarks: This function can be used to perform calculations in realtime, while a test is executing.

DIFF

Purpose: For all of the values in two selected columns (vectors), DIFF returns a third column (vector) containing the coefficient differences. That is, for columns V1 and V2, DIFF returns the following: (V12 - V11)/(V22 - V21), (V13 - V12)/(V23 - V22), etc.

Format: DIFF(V1, V2)

V1 = The name of any column (vector) listed under Columns.

V2 = The name of any column (vector) listed under Columns.

Example: GM = DIFF(I_DRAIN, V_GATE)

Remarks: This function can be used to perform calculations in realtime, while a test is executing.

DIMENSION

Purpose: Returns the number of occurrences of the last elements of the array.

Format: DIMENSION(V)

V = The name of any column (vector) listed under Columns.

Example: V2 = DIMENSION(V1)

Remarks: Returns the occurrence number.

EXP

Purpose: Returns the exponential, e_value, for each value in a column (vector) or for any operand.

Format: EXP(X)

X = The name of any column (vector) listed under Columns or any operand.

Example: NEWCURRENT = CURRENT*EXP(ANODEV)

Remarks: This function can be used to perform calculations in realtime, while a test is executing.

EXPFIT

Purpose: 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 the above exponential relationship, returns a new column (vector) containing Y values calculated from all X values in column VX.

Format: 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 (index) of the starting values.

ENDPOS = For the range of X and Y values to be exponentially fitted, the row number (index) of the ending values.

Example: DIODEI = EXPFIT(ANODEV, ANODEI, 2, LASTPOS(ANODEV))

Remarks: If a VX or VY value at either STARTPOS or ENDPOS is an invalid number (i.e., the value is #REF), the function will not return a valid result.

EXPFITA

Purpose: 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) (EXPFITA and EXPFITB are fit constants. Returns the value of the constant EXPFITA.)

Format: 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 (index) of the starting values.

ENDPOS = For the range of X and Y values to be exponentially fitted, the row number (index) of the ending values.

Example: DIODEOFFSET = EXPFITA(ANODEV, ANODEI, 2, LASTPOS(ANODEV))

Remarks: If a VX or VY value at either STARTPOS or ENDPOS is an invalid number (i.e., the value is #REF), the function will not return a valid result.

EXPFITB

Purpose: 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.

Returns the value of the constant EXPFITB in the relationship above.

Format: EXPFITB(VX, VY, STARTPOS, ENDPOS)

X = 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 (index) of the starting values.

ENDPOS = For the range of X and Y values to be exponentially fitted, the row number (index) of the ending values.

Example: DIODEIDEALITY = 1/(EXPFITB(ANODEV, ANODEI, 2, LASTPOS(ANODEV))*0.0257)

Remarks: If a VX or VY value at either STARTPOS or ENDPOS is an invalid number (i.e., the value is #REF), the function will not return a valid result.

FINDD

Purpose: Given a column (vector) V, beginning at START, FINDD searches down the column until it finds a value that matches the specified value X. Then it returns the row number (index) of that value. If FINDD does not find an exact match for X, it returns the row number (index) of the V value that is closest to X.

Format: FINDD(V, X, START)

V = The name of any column (vector) listed under Columns.

X = Any value (which may be the result of another calculation[s]).

START = The row number (index) of the starting value for the search.

Example: IF = AT(ANODEI, FINDD(ANODEV, 0.7, FIRSTPOS(ANODEV)))

Remarks: Refer to the similar functions FINDLIN (Find using linear interpolation) and FINDU (Find up).

FINDLIN

Purpose: Find using linear interpolation - given a column (vector) V, beginning at START, FINDLIN searches down the column until it finds a value that is closest (but does not exceed) the specified value X. Linear interpolation is then used to determine its decimal location between the found value and the next value in the column (vector). The returned index number (in decimal format) indicates the position of the specified value.

For example, assume you want to use FINDLIN to locate value 6 in the following array:

(Index 1)V

(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) which is between Index 3 (value 4) and Index 4 (value 8). Value 6 is halfway between Index 3 and Index 4. Therefore, FINDLIN will return Index 3.5.

Format: FINDD(V, X, START)

V = The name of any column (vector) listed under Columns.

X = Any value (which may be the result of another calculation[s]).

START = The row number (index) of the starting value for the search.

Example: IF = AT(ANODEI, FINDLIN(ANODEV, 0.7, FIRSTPOS(ANODEV)))

Remarks: Refer to the similar functions FINDD (Find down) and FINDU (Find up).

FINDU

Purpose: Given a column (vector) V, beginning at START, FINDU searches up the column until it finds a value that matches the specified value X. It then returns the row number (index) of that value. If FINDU does not find an exact match for X, it returns the row number (index) of the V value that is closest to X.

Format: 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[s]).

STARTPOS = The row number (index) of the starting value for the search.

Example: IF = AT(ANODEI, FINDU(ANODEV, 0.7, LASTPOS(ANODEV)))

Remarks: Refer to the similar functions FINDLIN (Find using linear interpolation) and FINDD (Find down).

FIRSTPOS

Purpose: Returns the row number (index) of the first value in a column (vector), typically the number 2.

Format: FIRSTPOS(V)

V = The name of any column (vector) listed under Columns.

Example: STARTOFARRAY = FIRSTPOS(I_DRAIN)

Remarks: Refer to the function LASTPOS.

GET FREQ

Purpose: Get the FFT frequency for a input time list.

Format: GET_FREQ(time_source)

Time_source = The name of time column (vector) listed under Columns.

Example: Fre1=GET_FREQ(time1)

Remarks: this formulator function just used in 1/f noise test project.

GET TBD

Purpose: This formula is used to get breakdown(BD) time of TDDB test.

Format: GET_TBD(IS, TIME, CUR_J, SLOPE_J, NOISE_J, FIX_CUR_J, SLOPE_P_NUM, NOISE_P_NUM, AND_OR,METHODS)

It includes the following parameters.

- two variables: IS and TIME.
- four methods for BD judgment: CUR_J, SLOPE_J,NOISE_J, IX_CUR_J.
- two parameters for calculation: SLOPE_P_NUM, NOISE_P_NUM.
- one logic relationship: AND_OR.

You must choose the methods:

- IS = gate current.
- TIME = test time.
- CUR_J= judgment current times for breakdown, as |Ig[n]/Ig[0]| >= CUR_J.
- SLOPE_J= judgment slope times for breakdown, as |slope[n]/slope[n-1]| >= SLOPE_J.
- NOISE_J= judgment noise times for breakdown, as |Inoi[n]/Inoi[n-1]| >= NOISE_J.
- FIX_CUR_J=fixed cursor for judgment, as |Ig[n]| >= FIX_CUR_J.
- SLOPE_P_NUM= the slope points number for each step calculate. If equals to 5, then slope of Ig is calculated every 5 points.
- NOISE_P_NUM= the noise points number for each step calculate. If equals to 5, then noise current of Ig is calculated every 5 points.
- AND_OR= ‘0’ or ’1’. ‘1’ stands ‘and’, means all methods selected are achieved then BD happens; ‘0’ stands ‘or’, means any of the methods selected are achieved then BD happens.
- METHODS= the used methods serial number. ‘1234’ means all 4 methods are chosen. ‘14’ means only first and fourth are chosen.

Example: TBD=GET_TBD(Ig, time, 100, 50, 10000, 1E-4, 5, 5, 0,14)

Remarks: The above example means calculating TBD using Ig and time array. The judgment current times is 100, judgment slope times is 50, the judgment noise times is 10000, the fixed cursor data for judgment is 1E-4. Slope of Ig is calculated every 5 points and noise current of Ig is calculated every 5 points. Using the first and fourth methods, and these two methods are achieved then BD happens.

GET TBD2

Purpose: This formula is used to get breakdown(BD) time of TDDB test.

Format: 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)

It includes the following parameters:

- four variables: IS and IS_TIME, ISILC, SILC_TIME.
- four methods for BD judgment: CUR_J, SLOPE_J,NOISE_J, FIX_CUR_J.
- two parameters for calculation: SLOPE_P_NUM, NOISE_P_NUM.

You must choose the methods:

- IS = gate stressed current.
- IS_TIME = stressed time.
- ISILC = gate leakage current.
- ISILC_TIME = measure ISILC time.
- CUR_J= judgment current times for breakdown, as |Ig[n]/Ig[0]| >= CUR_J.
- SLOPE_J= judgment slope times for breakdown, as |slope[n]/slope[n-1]| >= SLOPE_J.
- NOISE_J= judgment noise times for breakdown, as |Inoi[n]/Inoi[n-1]| >= NOISE_J.
- FIX_CUR_J=fixed cursor for judgment, as |Ig[n]| >= FIX_CUR_J.
- SLOPE_P_NUM= the slope points number for each step calculate. If equals to 5, then slope of Ig is calculated every 5 points.
- NOISE_P_NUM= the noise points number for each step calculate. If equals to 5, then noise current of Ig is calculated every 5 points.
- METHODS= the used methods serial number. '1234' means all 4 methods are chosen. '14' means only first and fourth are chosen.

Example: TBD=GET_TBD2(Ig, time_str, I_silc, time_silc, 100, 50, 10000, 1E-4, 5, 5, 0,14)

Remarks: The above example means calculating TBD using two group of arrays: Ig and time_str, I_silc and time_silc. The judgment current times is 100, judgment slope times is 50, the judgment noise times is 10000, the fixed cursor data for judgment is 1E-4. Slope of Ig is calculated every five points and noise current of Ig is calculated every five points. Using the first and fourth methods, and these two methods are achieved then BD happens.

GET VBD

Purpose: Get breakdown (BD) voltage in VRamp test.

Format: GET_VBD(IS, TIME, CUR_J, SLOPE_J, SLOPE_P_NUM, AND_OR, METHODS)

You must choose the methods:

- IS = gate current
- TIME = test time
- CUR_J = judgment current times for breakdown, as |Ig[n]/Ig[0]| >= CUR_J
- SLOPE_J = judgment slope times for breakdown, as |slope[n]/slope[n-1]| >= SLOPE_J
- SLOPE_P_NUM= the slope points number for each step calculate. If equals to 5, then slope of Ig is calculated every 5 points.
- AND_OR= ‘0’ or ’1’. ‘1’ stands ‘and’, means all methods selected are achieved then BD happens; ‘0’ stands ‘or’, means any of the methods selected are achieved then BD happens.
- METHODS= the used methods serial number. ‘12’ means two methods are chosen.

Example: VBD=GET_VBD(Ig, time, 100, 50,5, 5, 0,1)

Remarks: The above example means calculating VBD using Ig and time array. The judgment current times is 100, judgment slope times is 50. Slope of Ig is calculated every 5 points. Using the 1st methods, and it achieved then BD happens.

GMMAX

Purpose: Returns the maximum value from the two columns(vectors).

Format: GMMAX(Id, Vg)

Id= the I_drain vector.

Vg= the V_gate vector.

GM= DIFF(Id, Vg).

Example: Neg= GMMAX(DIFF(I_drain, V_gate))

JOIN

Purpose: to join multiply arrays together, and return a new array.

Format: JOIN(array1,array2)

Example: newarray=JOIN(array1,array2)

LASTPOS

Purpose: Returns the row number (index) of the last value in a column (vector).

Format: LASTPOS(V)

V = The name of any column (vector) listed under Columns.

Example: NUMSWEEPPTS = LASTPOS(COLLECTORI)

Remarks: Refer to the function FIRSTPOS.

LINFIT

Purpose: 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.

Format: 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 (index) of the first set of X and Y values.

ENDPOS = The row number (index) of the second set of X and Y values.

Example: RESISTORFIT = LINFIT (RESV, RESI, FIRSTPOS(RESV), LASTPOS(RESV))

Remarks: If a VX or VY value at either STARTPOS or ENDPOS is an invalid number (i.e., 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.

LINFITSLP

Purpose: 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 slope of the linear equation (value of "b" in Y = a + bX).

Format: 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 (index) of the first set of X and Y values.

ENDPOS = The row number (index) of the second set of X and Y values.

Example: RESISTANCE = 1/LINFITSLP(RESV, RESI, FIRSTPOS(RESV), LASTPOS(RESV))

Remarks: If a VX or VY value at either STARTPOS or ENDPOS is an invalid number (i.e., 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.

LINFITXINT

Purpose: 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).

Format: 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 (index) of the first set of X and Y values.

ENDPOS = The row number (index) of the second set of X and Y values.

Example: EARLYV = LINFITXINT(COLLECTORV, COLLECTORI, 56, 75)

Remarks: If a VX or VY value at either STARTPOS or ENDPOS is an invalid number (i.e., 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.

LINFITYINT

Purpose: 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 Y intercept of the linear equation:

(value of "a" in Y = a +bX).

Format: 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 (index) of the first set of X and Y values.

ENDPOS = The row number (index) of the second set of X and Y values.

Example: OFFSET = LINFITYINT(GATEV, GATEI, FIRSTPOS(GATEV), LASTPOS(GATEV))

Remarks: If a VX or VY value at either STARTPOS or ENDPOS is an invalid number (i.e., 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 (vectors), use the REGFITYINT function.

LN

Purpose: Returns the base-e (natural, Napierian) log of each value in a designated column (vector) or the Napierian log of any operand.

Format: LN(X)

X = The name of any column (vector) listed under Columns or any operand.

Example: DIODEV = LN(ANODEI)*0.026

Remarks: This function can be used to perform calculations in realtime, while a test is executing.

LOG

Purpose: Returns the log value of each value in a designated column (vector) or the log of any operand. Note that it cannot accept two arrays in its parameters.

Format: LOG(V,VBASE)

V = The name of any column (vector) listed under Columns or a number.

VBASE= The base value of logarithms.

Example: F1 = LOG(I_DRAIN, 2)

Remarks: This function can be used to perform calculations in realtime, while a test is executing.

LOG10

Purpose: Returns the base-10 log of each value in a designated column (vector) or the base-10 log of any operand.

Format: LOG10(V)

V = The name of any column (vector) listed under Columns or any operand.

Example: F2 = LOG10(I_DRAIN)

Remarks: This function can be used to perform calculations in realtime, while a test is executing.

LOGFIT

Purpose: Finds a log equation of the form Y = a + b*log10(X) from two sets of X and Y values selected from two columns (vectors), VX and VY. The two points are specified by the arguments STARTPOS and ENDPOS.

Using the above log relationship, returns a new column (vector) containing Y values calculated from all X values in column VX.

Format: 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.

Example: F3=LOGFIT(I_FM_pre1,I_Neg_pre1,23,56)

Remarks: If a VX or VY value at either STARTPOS or ENDPOS is an invalid number (i.e., the value is #REF), the function will not return a valid result.

MAX

Purpose: Searches all values in a column (vector) and returns the maximum value.

Format: MAX(V)

V = The name of any column (vector) listed under Columns.

Example: MAXGM = MAX(DIFF(DRAINI, GATEV))

MAXPOS

Purpose: Searches all values in a column (vector), finds the maximum value, and returns the row number (index) of the maximum value.

Format: MAXPOS(V)

V = The name of any column (vector) listed under Columns.

Example: PEAKSTRESS = AT(GATEV, MAXPOS(SUBSTRATEI))

MIN

Purpose: Searches all values in a column (vector) and returns the minimum value.

Format: MIN(V)

V = The name of any column (vector) listed under Columns.

Example: SMALLESTI = MIN(DRAINI)

MINPOS

Purpose: Searches all values in a column (vector), finds the minimum value, and returns the row number (index) of the minimum value.

Format: MINPOS(V)

V = The name of any column (vector) listed under Columns.

Example: LOCATION = MINPOS(DRAINI)

MOBILITY

Purpose: Mobility at Vg_tar by Id-Vg curve.

Format: 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

Example: U1= MOBILITY(10,0.03,1,0.05,0.078,Id,Vg)

NOISE UTM

Purpose: This formula is used to generate correct time array for noise current

Format: NOISE_UTIM(MEASTIMES,POINT_NUM)

POW

Purpose: Returns the power value of each value in a designated column (vector) or any operand. Note that it cannot accept two arrays in its parameters.

Format: 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.

Example: I2 = POW(I_DRAIN, 2) or I2pow = POW(2, I_DRAIN)

POWERFFT

Purpose: Perform the forward FFT for input data source, get the power spectrum.

Format: POWERFFT(data_source, dialog box)

data_source: The value list of the sample data . a list

dialog box: The name of the dialog box function. integer

Value mappings of the parameter dialog box:

POWERFFT value table

Value	Dialog box	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)

Remarks: Return the power spectrum calculated for input source, this function only used in 1/f noise test project

REGFIT

Purpose: Performs a linear regression fit.

Fits the relationship of the form Y = aX + b to a specified range of values in two columns (vectors): column VX containing X values and column VY containing Y values:

Y = REGFITYINT + REGFITSLP * X (REGFITSLP and REGFITYINT are slope and Y-intercept constants. Using the above linear relationship, returns a new column (vector) containing Y values calculated from all X values in column VX).

Format: 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 (index) of the starting values.

ENDPOS = For the range of X and Y values to be fitted, the row number (index) of the ending values.

Example: COLLECTORFIT = REGFIT(COLLECTORV, COLLECTORI, 25, LASTPOS(COLLECTORV))

Remarks: If a VX or VY value at either STARTPOS or ENDPOS is an invalid number (i.e., the value is #REF), the function will not return a valid result.

REGFITSLP

Purpose: Fits the relationship of the form Y = a + bX to a specified range of values in two columns (vectors): 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.

Format: 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 (index) of the starting values.

ENDPOS = For the range of X and Y values to be fitted, the row number (index) of the ending values.

Example: COLLECTORRES = 1/REGFITSLP(COLLECTORV, COLLECTORI, 25, LASTPOS(COLLECTORV))

Remarks: If a VX or VY value at either STARTPOS or ENDPOS is an invalid number (i.e., the value is #REF), the function will not return a valid result.

REGFITXINT

Purpose: Fits the relationship of the form Y = a + bX to a specified range of values in two columns (vectors): column VX containing X values and column VY containing Y values: Y = REGFITYINT + REGFITYINT * X where REGFITSLP and REGFITYINT are slope and Y-intercept constants.

Returns the value of the X intercept for relationship above (–REGFITYINT/REGFITSLP).

Format: 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 (index) of the starting values.

ENDPOS = For the range of X and Y values to be fitted, the row number (index) of the ending values.

Example: EARLYV = REGFITXINT(COLLECTORV, COLLECTORI, 25, LASTPOS(COLLECTORV))

Remarks: If a VX or VY value at either STARTPOS or ENDPOS is an invalid number (i.e., the value is #REF), the function will not return a valid result.

REGFITYINT

Purpose: Fits the relationship of the form Y = a + bX to a specified range of values in two columns (vectors)—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 (REGFITYINT) for relationship above.

Format: 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 (index) of the starting values.

ENDPOS = For the range of X and Y values to be fitted, the row number (index) of the ending values.

Example: OFFSET = REGFITYINT(COLLECTORV, COLLECTORI, 25, LASTPOS(COLLECTORV))

Remarks: If a VX or VY value at either STARTPOS or ENDPOS is an invalid number (i.e., the value is #REF), the function will not return a valid result.

REGFIT LGX LGY

Purpose: Performs a linear regression fit.

Fits the relationship of the form LGY = aLGX + b to a specified range of values in two columns (vectors):

column VX extracting LGX values and column VY extracting LGY values: LGY = REGFITYINT + REGFITSLP * LG X where REGFITSLP and REGFITYINT are slope and LGY-intercept constants.

Using the above linear relationship, returns a new column (vector) containing LGY values calculated from all LGX values. That is, input value is normal VX, VY, but the fit relationship is built on LGY and LGX.

Format: 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 (index) of the starting values.

ENDPOS = For the range of X and Y values to be fitted, the row number (index) of the ending values.

Example: COLLECTORFIT1 = REGFIT_LGX_LGY(COLLECTORV, COLLECTORI, 25, LASTPOS(COLLECTORV))

Remarks: If a VX or VY value at either STARTPOS or ENDPOS is an invalid number (i.e., the value is #REF), the function will not return a valid result.

REGFIT LGX Y

Purpose: Performs a linear regression fit.

Fits the relationship of the form Y = aLGX + b to a specified range of values in two columns (vectors): column VX extracting LGX values and column VY containing Y values: Y = REGFITYINT + REGFITSLP * LGX where REGFITSLP and REGFITYINT are slope and Y-intercept constants.

Using the above linear relationship, returns a new column (vector) containing Y values calculated from all LGX values in column VX. That is, input value is normal value VX,VY, but the fit relationship is built on Y and LGX.

Format: 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 (index) of the starting values.

ENDPOS = For the range of X and Y values to be fitted, the row number (index) of the ending values.

Example: COLLECTORFIT 2 = REGFIT_LGX_Y(COLLECTORV, COLLECTORI, 25, LASTPOS(COLLECTORV))

Remarks: If a VX or VY value at either STARTPOS or ENDPOS is an invalid number (i.e., the value is #REF), the function will not return a valid result.

REGFIT X LGY

Purpose: Performs a linear regression fit.

Fits the relationship of the form LGY = a X+ b to a specified range of values in two columns (vectors)—column VX containing X values and column VY extracting LGY values: LGY = REGFITYINT + REGFITSLP * X where REGFITSLP and REGFITYINT are slope and LGY-intercept constants.

Using the above linear relationship, returns a new column (vector) containing LGY values calculated from all X values in column VX. That is, input value is normal value VX, VY, but the fit relationship is built on LGY and X.

Format: 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 (index) of the starting values.

ENDPOS = For the range of X and Y values to be fitted, the row number (index) of the ending values.

Example: COLLECTORFIT3= REGFIT_X_LGY (COLLECTORV, COLLECTORI, 25, LASTPOS(COLLECTORV))

Remarks: If a VX or VY value at either STARTPOS or ENDPOS is an invalid number (i.e., the value is #REF), the function will not return a valid result.

RES

Purpose: Calculate resistance vector value by giving V and I.

Format: RES(V,I))

V=voltage vector listed in Variables box

I=current vector listed in Variables box

Example: RG_1=RES(V_Gate_1,I_Gate_1)

Remarks: If a I or V value is an invalid number (i.e., the value is #REF), the function will not return a valid result.

RES 4WIRE

Purpose: calculate 4-wire resistance vector value by giving V and I.

Format: 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

Example: RGD=RES_4WIRE(V_Gate_1,V_Drain_1,I_Gate_1)

Remarks: If VPOS, VNEG, or I value is an invalid number (i.e., the value is #REF), the function will not return a valid result.

RES 4WIRE AVG (VPOS, VNEG, I)

Purpose: calculate 4-wire resistance average value by giving V and I.

Format: 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

Example: RGD_AVG=RES_4WIRE_AVG(V_Gate_1,V_Drain_1,I_Gate_1)

Remarks: If VPOS, VNEG , or I value is an invalid number (i.e., the value is #REF), the function will not return a valid result.

RES AVG

Purpose: calculate resistance average value by giving V and I

Format: RES_AVG(V,I)

V=voltage vector listed in Variables box

I=current vector listed in Variables box

Example: RG_1=RES_ AVG (V_Gate_1, I_Gate_1)

Remarks: If a I or V value is an invalid number (i.e., the value is #REF), the function will not return a valid result.

SMOOTH

Purpose: Smooth the vector using FFT.

Format: SMOOTH (V, RANK)

V = The name of any column (vector) listed under Columns.

RANK = The rank number of smooth level.

Example: I2 = SMOOTH (I1, 2)

Remarks: If a V value is an invalid number (i.e., the value is #REF), the function will not return a valid result.

SQRT

Purpose: Returns the square root of each value in a designated column (vector) or the square root of any operand.

Format: SQRT(X)

X = The name of any column (vector) listed under Columns or any operand.

Example: TWO = SQRT(4)

Remarks: A negative value of X returns #REF in the Data worksheet. This function can be used to perform calculations in realtime, while a test is executing.

SS

Purpose: Extract the sub-threshold swing between a certain duration of current. The sub-threshold swing is defined as the deferential of Vg and log Id, for example, SS=dVg/dlogId.

Format: SS (I_DRAIN, V_GATE, START_I, STOP_I)

I_DRAIN = the drain current measured in the test.

V_GATE = the gate voltage forced in the test.

START_I = the start current to define the calculation region.

STOP_I = the stop current to define the calculation region.

Example: S0=SS (I_Drain_1, V_Gate_1, 1.3e-5, 1.5e-5)

Remarks: If an I_DRAIN or V_GATE value is an invalid number (for example, the value is #REF), the function will not return a valid result.

SSVTCI

Purpose: Extract the sub-threshold swing from the certain duration between left offset and right offset of the constant current threshold voltage.

Format: SSVTCI(I_DRAIN, V_GATE, VTCI,LEFT_OFFSET,RIGHT_OFFSET)

I_DRAIN = the drain current measured in test.

V_GATE= the gate voltage.

VTCI= constant current threshold voltage.

LEFT_OFFSET,RIGHT_OFFSET= left offset and right offset of the constant current threshold voltage

Example: SLOPE=SSVTCI(I_Drain_1,V_Gate_1,VTH,-0.1,-0.1)

Remarks: If a I_DRAIN or V_GATE value is an invalid number (i.e., the value is #REF), the function will not return a valid result.

SUBARRAY1

Purpose: Extract a subarray based on full array.

Format: SUBARRAY1(V, start, period)

V: vector array

start: start index

period: extraction period

Example: VGate2=SUBARRAY1(VGate, 1, 5)

SUBARRAY2

Purpose: Extract a subarray based on full array.

Format: SUBARRAY2(V, start, stop)

V: vector array

start: start index

stop: stop index

Example: VGate2=SUBARRAY1(VGate, 1, 50)

TANFIT

Purpose: Finds a linear equation of the form Y = aX + b from two columns (vectors), 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 (vector) containing Y value calculated from all X values in column VX.

Format: 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 (index) of the value where the tangent is to be found.

Example: VTFIT = TANFIT(GATEV, DRAINI, MAXPOS(GM))

Remarks: If a VX or VY value at POS is an invalid number (i.e., the value is #REF), the function will not return a valid result.

TANFITSLP

Purpose: Finds a linear equation of the form Y = aX+ b from two columns (vectors), 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 = aX + b).

Format: 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 (index) of the value where the tangent is to be found.

Example: VTSLOPE = TANFITSLP(GATEV, DRAINI, MAXPOS(GM))

Remarks: If a VX or VY value at POS is an invalid number (i.e., the value is #REF), the function will not return a valid result.

TANFITXINT

Purpose: Finds a linear equation of the form Y = aX + b from two columns (vectors), 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 = aX + b).

Format: 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 (index) of the value where the tangent is to be found.

Example: VT = TANFITXINT(GATEV, DRAINI, MAXPOS(GM))

Remarks: If a VX or VY value at POS is an invalid number (i.e., the value is #REF), the function will not return a valid result.

TANFITYINT

Purpose: Finds a linear equation of the form Y = aX + b from two columns (vectors), 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 = aX + b).

Format: 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 (index) of the value where the tangent is to be found.

Example: OFFSET = TANFITYINT(GATEV, DRAINI, GMMAX)

Remarks: If a VX or VY value at POS is an invalid number (i.e., the value is #REF), the function will not return a valid result.

TAR Y at X

Purpose: Find y_tar in y corresponding to x_tar in x’s index

Format: TAR_YatX(x, y, x_tar)

x	x array
y	y array
x_tar	target x

Example: y0=TAR_YatX(V, I, 0.1)

TTF DID LGT

Purpose: TTF means time to failure. 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, the fitting relationship is built on time log scale and degradation of drain current.

Format: TTF_DID_LGT(DID, T, START_POS, END_POS, GOAL)

DID = degradation (in percentage) of drain current listed under Columns

T= time array listed under Columns

STARTPOS = For the range of T and DID values to be fitted, the row number (index) of the starting values.

ENDPOS = For the range of T and DID values to be fitted, the row number (index) of the ending values.

GOAL = the aim degradation (in percentage) value.

Example: F1= TTF_DID_LGT (I_DRAIN_DEG, TIME, 5, 11, 10)

Remarks: If a T or DID value at pos is an invalid number (i.e., the value is #REF), the function will not return a valid result.

TTF LGDID T

Purpose: TTF means time to failure. 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, the fitting relationship is built on time array and log value of degradation of drain current.

Format: TTF_LGDID_T(DID, T, START_POS, END_POS, GOAL)

DID = degradation (in percentage) of drain current listed under Columns

T= time array listed under Columns.

STARTPOS = For the range of T and DID values to be fitted, the row number (index) of the starting values.

ENDPOS = For the range of T and DID values to be fitted, the row number (index) of the ending values.

GOAL = the aim degradation(in percentage) value

Example: F2 = TTF_LGDID_T (I_DRAIN_DEG, TIME, 5, 11, 10)

Remarks: If a T or DID value at pos is an invalid number (i.e., the value is #REF), the function will not return a valid result.

TTF DID T

Purpose: TTF means time to failure. 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, the fitting relationship is built on time array and degradation of drain current.

Format: TTF_DID_T(DID, T, START_POS, END_POS, GOAL)

DID = degradation (in percentage) of drain current listed under Columns.

T = time array listed under Columns.

STARTPOS = For the range of T and DID values to be fitted, the row number (index) of the starting values.

ENDPOS = For the range of T and DID values to be fitted, the row number (index) of the ending values.

GOAL = the aim degradation (in percentage).

Example: F3= TTF_DID_T (I_DRAIN_DEG, TIME, 5, 11, 10)

Remarks: If a T or DID value at pos is an invalid number (i.e., the value is #REF), the function will not return a valid result.

TTF LGDID LGT

Purpose: TTF means time to failure. 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, the fitting relationship is built on time log scale and degradation of drain current.

Format: TTF_LGDID_LGT(DID, T, START_POS, END_POS, GOAL)

DID = degradation (in percentage) of drain current listed under Columns.

T = time array listed under Columns.

STARTPOS = For the range of T and DID values to be fitted, the row number (index) of the starting values.

ENDPOS = For the range of T and DID values to be fitted, the row number (index) of the ending values.

GOAL = the aim degradation(in percentage) value.

Example: F4= TTF_LGDID_LGT (I_DRAIN_DEG, TIME, 5, 11, 10)

Remarks: If a T or DID value at pos is an invalid number (i.e., the value is #REF), the function will not return a valid result.

VTCI

Purpose: algorithm by using u, w, l to calculate Id_target.

Format: VTCI(u, w, l, I_DRAIN,V_GATE)

u: the current density

w: the device width

l: the device length

I_DRAIN : the I_drain vector

V_GATE: the V_gate vector

Example: VTH1 = VTCI(0.001,1,0.03,I_DRAIN,V_GATE)

Remarks: If an I_DRAIN or V_GATE value at POS is an invalid number (i.e., the value is #REF), the function will not return a valid result.

VTLINGM

Purpose: Extrapolates threshold voltage from measurement of maximum slope of the ID-VGS curve, as below:

Vth = VGS(@Gmmax)-ID(@Gmmax)/Gmmax (VGS(@Gmmax is the gate voltage at the point of the maximum slope of the ID-VGS curve; ID(@Gmmax is the drain current at the point of the maximum slope of the ID-VGS curve; Gmmax is the maximum slope of the ID-VGS curve).

Format: VTLINGM(I_DRAIN, V_GATE)

I_DRAIN is the drain current measured in test. V_GATE is the gate voltage.

Example: VTH0 = VTLINGM(I_DRAIN, V_GATE)

Remarks: DC bias voltages for Vth measurements are VDS = VDS_lin, VBS = VBB typically, for PMOS, VDS_lin=-0.1 V(@VDD=5V); for NMOS, VDS_lin=0.1V(@VDD=5V). If a I_DRAIN or V_GATE value is an invalid number (i.e., the value is #REF), the function will not return a valid result.

VTSATGM

Purpose: Extrapolates threshold voltage from measurement of the IDS-VGS curve. In saturation region of MOSFET, Vth=Vgs(@Ids=0).

DC bias voltages for Vth saturation measurements are

VDS = VGS = VDD, VBS = VBB

Format: VTSATGM(I_DRAIN, V_GATE)

Example: VTH1 = VTSATGM(I_DRAIN, V_GATE)

Remarks: If an I_DRAIN or V_GATE value at POS is an invalid number (i.e., the value is #REF), the function will not return a valid result.

VT SAT TRI

Purpose: Vt algorithm by triple divide Id square root curve.

Format: VT_SAT_TRI(Vg_tar, Id, Vg)

Vg_tar Target VG

Id Measured Drain current

Vg Measured Gate voltage

Example: VT0 = VT_SAT_TRI(0.08, Id, Vg

KEITHLEY

A GREATER MEASURE OF CONFIDENCE

Keithley Instruments, Inc.

Corporate Headquarters • 28775 Aurora Road • Cleveland, Ohio 44139 • 440-248-0400 • Fax: 440-248-6168 • 1-888-KEITHLEY • www.keithley.com

12/06