TI-Nspire™ CX Technology Release History:
v4.0 and above

TEXAS INSTRUMENTS

What's New

What's New in Version 4.5

Unit Conversions

- Unit conversions are now available on all platforms (Handheld, Windows®, Mac®, and iPad®) for the TI-Nspire™ CX.

- A Unit Conversion Assistant has been added that allows the user to automatically generate unit conversion statements without having to type each unit.

Constants and Values

- Scientific constants are now available on all platforms (Handheld, Windows®, Mac®, and iPad®) for the TI-Nspire™ CX.

- New scientific constants have been added.

- Values for scientific constants from previous versions of TI-Nspire™ CX have been updated.

Program Editor

- The following enhancements have been made to the Program Editor to reduce the number of steps to run a program or function and avoid potential syntax errors caused by the location of the cursor.

  - Added a **Run** item to the **Check Syntax & Store** menu. This automatically inserts a new Calculator application (if one doesn't exist) and pastes the name of the current program or function into the Calculator.

  - When creating a new program or function, the cursor now defaults to the first blank line instead of inside the parentheses.

  - When editing inside a quotes template such as **Disp** or **Send** commands, pressing the Enter key will now create a new line instead of remaining inside the template.

- The Hub menu has been updated with:

  - commands for the TI-Innovator™ Rover

  - entries for the vibration motor (part of the TI-Innovator™ I/O Module pack)

Programming Updates

- Two new commands have been added:

  - **DispAt** - Allows the user to create a dashboard-like display of data from a loop rather than a long and growing list. This also reduces the amount of history stored which will improve the overall performance of programs that run for long periods of time.

  - **getKey** - Allows programs to read and respond to keypad input.

- Added support for **Get**, **GetStr**, and **Send** commands in math boxes within the Notes application.

To report issues and help us improve our apps, please email us at ti-cares@ti.com.
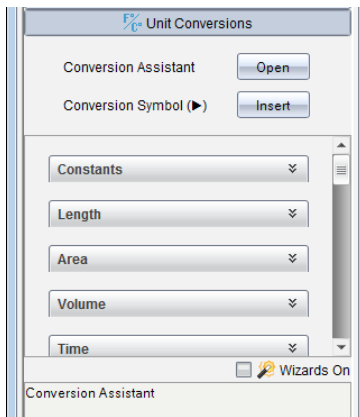
Using the Unit Conversion Assistant

In any application where math input is allowed, you can generate unit conversions using the Unit Conversion Assistant. This can help reduce syntax errors by automatically entering the units for you.

Example: Convert 528 minutes to hours. The desired expression is 528•_min▶_hr.

1. Type **528** on the entry line.

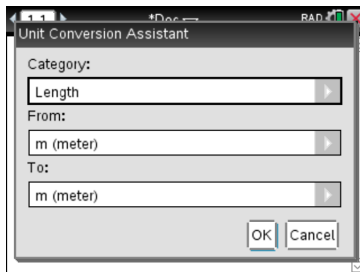2. On the **Utilities** tab, click the **Unit Conversions** bar.

   Handheld: Press 📖 3.

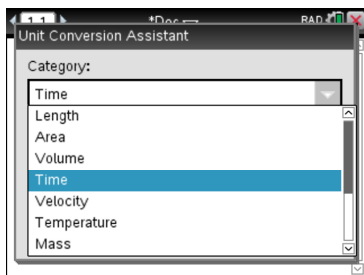3. Click the **Open** button next to **Conversion Assistant**.

   Handheld: Press enter.

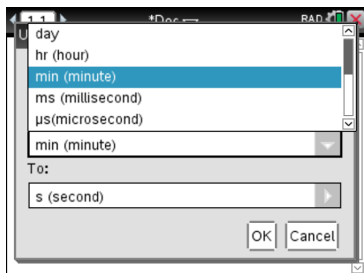   The **Unit Conversion Assistant** dialog box displays:

4. Click the **Category** list and select **Time**.

   Handheld: Scroll to the **Time** category and press enter.

5. Click the **From** list and select **min (minute)**.

    Handheld: Scroll to **min (minute)** and press `enter`.

Note: You can select **Use existing unit** at the bottom of the list if you have already entered a unit. In this example, you might have already entered 528•_min.

6. Click the **To** list and select **hr (hour)**.

    Handheld: Scroll to **hr (hour)** and press `enter`.
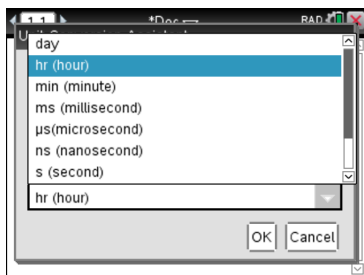
7. Click **OK** to paste **_min▶_hr** to the entry line.
8. Press **Enter** to evaluate the expression.

    Handheld: Press `enter`.

Note:

- The last Category, From, and To selections will be retained until:
    - the software is closed and re-opened (Desktop)
    - the device is reset (Handheld)
    - the language is changed, or the app is uninstalled or upgraded (iPad)
- Inserting a conversion into a Notes text field will automatically create a Math Box.
- Inserting a conversion into an empty line in the Calculator will automatically insert **Ans** before the conversion.

Conversion Categories and Units

The following table lists categories and their respective units that are available in the Unit Conversion Assistant. When the units are pasted to the entry line, underscores (_) are added before the name of each unit to differentiate them from other variables. For example:

- _cm►_m
- _km/_hr►_m/_s
- _°C►_°F

Category	Units
Length	Å (Angstrom)
	au (astronomical unit)
	cm (centimeter)
	dm (decimeter)
	fath (fathom)
	fm (fermi)
	ft (foot)
	in (inch)
	km (kilometer)
	ltyr (light-year)

Category	Units
	m (meter)
	µm (micron)
	mi (mile)
	mil (1/1000 inch)
	mm (millimeter)
	nm (nanometer)
	Nmi (nautical mile)
	pc (parsec)
	rod (rod)
	yd (yard)
Area	acre (acre)
	cm^2
	dm^2
	ft^2
	ha (hectare)
	in^2
	km^2
	m^2
	mi^2
	mm^2
	yd^2
Volume	cm^3
	cup (cup)
	dm^3
	ft^3
	floz (US fluid ounce)
	flozUK (British fluid ounce)
	gal (US gallon)
	galUK (British gallon)
	in^3
	l (liter)
	m^3
	ml (milliliter)

Category	Units
	pt (pint)
	qt (quart)
	tbsp (tablespoon)
	tsp (teaspoon)
	yd^3
Time	day (day)
	hr (hour)
	min (minute)
	ms (millisecond)
	µs (microsecond)
	ns (nanosecond)
	s (second)
	week (week)
	yr (year)
Velocity	ft/min
	ft/s
	knot (knot)
	km/h
	km/min
	km/s
	m/s
	mi/h
	mi/min
	mi/s
Temperature	°C (Celsius)
	°F (Fahrenheit)
	K (kelvin)
	°R (Rankine)
Mass	amu (atomic mass unit)
	gm (gram)
	kg (kilogram)
	lb (pound)
	mg (milligram)
	mton (metric ton)
	oz (ounce)
	slug (slug)
	ton (ton)

Category	Units
	tonUK (long ton)
Force	dyne (dyne)
	kgf (kilogram force)
	lbf (pound force)
	N (newton)
	tonf (ton force)
Energy	BTU (British thermal unit)
	cal (calorie)
	erg (erg)
	eV (electron volt)
	ftlb (foot-pound)
	J (joule)
	kcal (kilocalorie)
	kJ (kilojoule)
	kgf*m
	kWh (kilowatt-hour)
	latm (liter-atmosphere)
Power	hp (horsepower)
	kW (kilowatt)
	PS (metric horsepower)
	W (watt)
Pressure	atm (atmosphere)
	bar (bar)
	inH_2O (inches of water)
	inHg (inches of mercury)
	kPa (kilopascal)
	kgf/cm^2
	lbf/in^2
	mbar (millibar)
	mmH_2O (millimeters of water)
	mmHg (millimeters of mercury)
	N/m^2
	Pa (pascal)
	psi (pounds per square inch)
	torr (millimeters of mercury)

Getting Started with the Program Editor

You can create user-defined functions or programs by typing definition statements on the Calculator entry line or by using the Program Editor. The Program Editor offers some advantages, and it is covered in this section. For more information, see *Calculator*.

- The editor has programming templates and dialog boxes to help you define functions and programs using correct syntax.
- The editor lets you enter multiple-line programming statements without requiring a special key sequence to add each line.
- You can easily create private and public library objects (variables, functions, and programs). For more information, see *Libraries*.

Launching the Program Editor

▶ To add a new Program Editor page in the current problem:

From the toolbar, click **Insert > Program Editor > New**.

Handheld: Press doc▾ and select **Insert > Program Editor > New**.

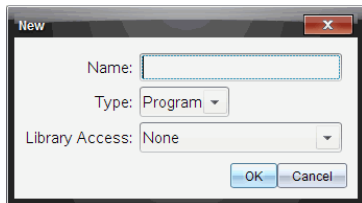**Note:** The editor is also accessible from the **Functions & Programs** menu of a Calculator page.
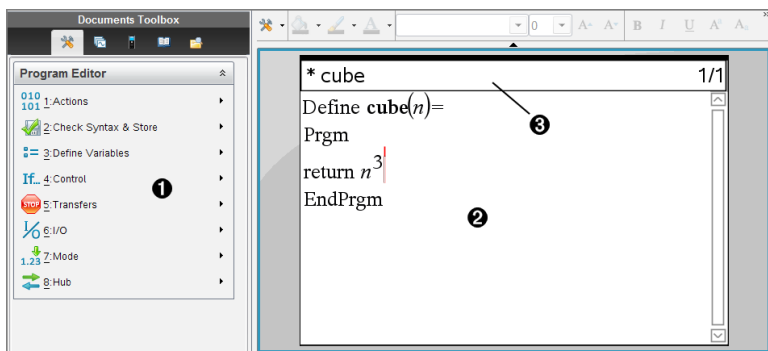
❶ Program Editor menu – This menu is available anytime you are in the Program Editor work area using the Normal view mode.

❷ Program Editor work area

❸ Status line shows line-number information and the name of the function or program being edited. An asterisk (*****) indicates that this function is "dirty," which means that it has changed since the last time its syntax has been checked and it has been stored.

## *Defining a Program or Function*

Starting a new Program Editor

1. Make sure you are in the document and problem in which you want to create the program or function.

2. Click **Insert** button 🔁 on the application toolbar, and select **Program Editor > New**. (On the handheld, press 📄 and select **Insert > Program Editor > New**.)

3. Type a name for the function or program you are defining.

4. Select the **Type** (**Program** or **Function**).

5. Set the **Library Access**:

   - To use the function or program only from the current document and problem, select **None**.

   - To make the function or program accessible from any document but not visible in the Catalog, select **LibPriv**.

   - To make the function or program accessible from any document and also visible in the Catalog, select **LibPub (Show in Catalog)**. For more information, see *Libraries*.

6. Click **OK**.

   A new instance of the Program Editor opens, with a template matching the selections you made.
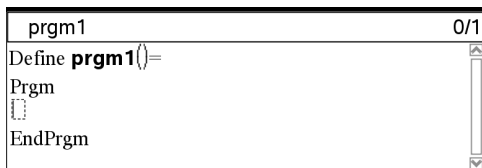
Entering Lines into a Function or Program

The Program Editor does not execute the commands or evaluate expressions as you type them. They are executed only when you evaluate the function or run the program.

1. If your function or program will require the user to supply arguments, type parameter names in the parentheses that follow the name. Separate parameters with a comma.

Define **prgm1**(a,b)=
Prgm

EndPrgm

2. Between the Func and EndFunc (or Prgm and EndPrgm) lines, type the lines of
    statements that make up your function or program.

Define **prgm1**(a,b)=
Prgm
    Disp "a=",a
    Disp "b=",b
    Disp "a^b=",a^b
EndPrgm

- You can either type the names of functions and commands or insert them from
  the Catalog.

- A line can be longer than the width of the screen; if so, you might have to scroll
  to view the entire statement.

- After typing each line, press **Enter**. This inserts a new blank line and lets you
  continue entering another line.

- Use the ◀, ▶, ▲, and ▼ arrow keys to scroll through the function or program
  for entering or editing commands.

Inserting Comments

A comment symbol (©) lets you enter a remark. Comments can be useful to someone
viewing or editing the program. Comments are not displayed when the program runs,
and they have no effect on program flow.

Define LibPub volcyl(ht,r) =
Prgm
©volcyl(ht,r) => volume of cylinder ❶
  Disp "Volume =", approx($\pi \cdot r^2 \cdot ht$)
©This is another comment.
EndPrgm

❶ Comment showing required syntax. Because this library object is public and this
   comment is the first line in a Func or Prgm block, the comment is displayed in the
   Catalog as help. For more information, see *Libraries*.

To insert a comment:

1. Position the cursor at the end of the line in which you want to insert a comment.

2. From the **Actions** menu, click **Insert Comment**.

3. Type the text of the comment after the © symbol.

Checking Syntax

The Program Editor lets you check the function or program for correct syntax.

▶ From the **Check Syntax & Store** menu, click **Check Syntax**.

If the syntax checker finds any syntax errors, it displays an error message and tries to position the cursor near the first error so you can correct it.
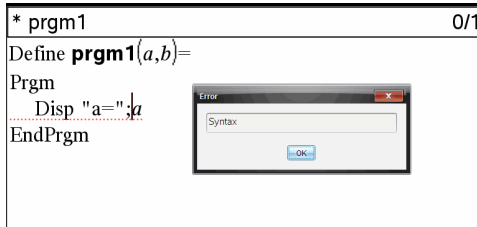
Storing the Function or Program

You must store your function or program to make it accessible. The Program Editor automatically checks the syntax before storing.

An asterisk (*) is displayed in the upper left corner of the Program Editor to indicate that the function or program has not been stored.

▶ From the **Check Syntax & Store** menu, click **Check Syntax & Store**.

If the syntax checker finds any syntax errors, it displays an error message and tries to position the cursor near the first error.

If no syntax errors are found, the message "Stored successfully"is displayed in the status line at the top of the Program Editor.

Note: If the function or program is defined as a library object, you must also save the document in the designated library folder and refresh libraries to make the object accessible to other documents. For more information, see *Libraries*.
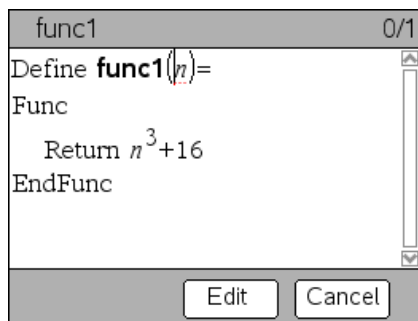
Viewing a Program or Function

1. From the **Actions** menu, click **View**.
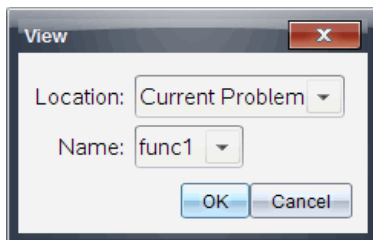
2. If the function or program is a library object, select its library from the **Location** list.

3. Select the function or program name from the **Name** list.

   The function or program is displayed in a viewer.

4. Use the arrow keys to view the function or program.

5. If you want to edit the program, click **Edit**.

   Handheld: Press [tab] to highlight **Edit**, and then press [enter].

   Note: The **Edit** selection is available only for functions and programs defined in the current problem. To edit a library object, you must first open its library document.

Opening a Function or Program for Editing

You can open a function or program from the current problem only.

Note: You cannot modify a locked program or function. To unlock the object, go to a Calculator page and use the **unLock** command.

1. Display the list of available functions and programs.
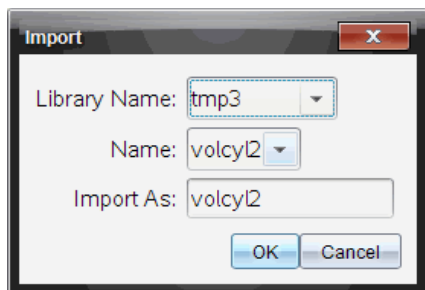
   - From the **Actions** menu, click **Open**.
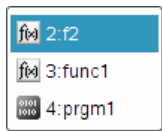
2. Click the item to open.

Importing a Program from a Library

You can import a function or program defined as a library object into a Program Editor within the current problem. The imported copy is not locked, even if the original is locked.

1. From the **Actions** menu, click **Import**.

2. Select the **Library Name**.

3. Select the **Name** of the object.

4. If you want the imported object to have a different name, type the name under **Import As**.

Creating a Copy of a Function or Program

When creating a new function or program, you might find it easier to start with a copy of the current one. The copy that you create is not locked, even if the original is locked.

1. From the **Actions** menu, click **Create Copy**.

2. Type a new name, or click **OK** to accept the proposed name.

3. If you want to change the access level, select **Library Access**, and select a new level.

Renaming a Program or Function

You can rename and (optionally) change the access level of the current function or program.
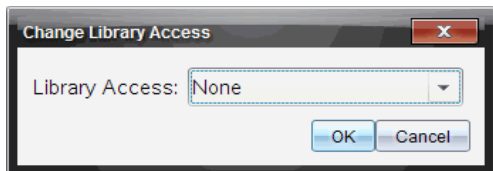
1. From the **Actions** menu, click **Rename**.
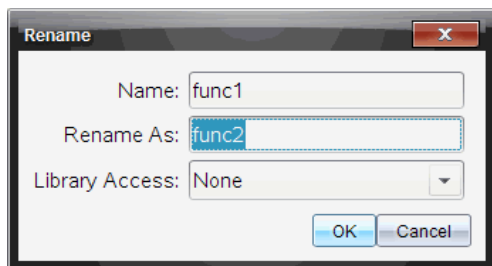
2. Type a new name, or click **OK** to accept the proposed name.

3. If you want to change the access level, select **Library Access**, and select a new level.

Changing the Library Access Level

1. From the **Actions** menu, click **Change Library Access**.

2. Select the **Library Access**:

    -   To use the function or program only from the current Calculator problem, select **None**.

    -   To make function or program accessible from any document but not visible in the Catalog, select **LibPriv**.

    -   To make the function or program accessible from any document and also visible in the Catalog, select **LibPub**.
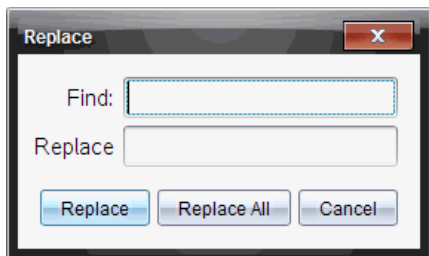
Finding Text

1. From the **Actions** menu, click **Find**.
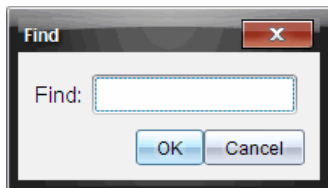
2. Type the text that you want to find, and click **OK**.

    -   If the text is found, it is highlighted in the program.

    -   If the text is not found, a notification message is displayed.

Finding and Replacing Text

1. From the **Actions** menu, click **Find and Replace**.

2. Type the text that you want to find.

3. Type the replacement text.

4. Click **Replace** to replace the first occurrence after the cursor position.
—or—
Click **Replace All** to replace every occurrence.

   Note: If the text is found in a math template, a message is displayed to warn you that your replacement text will replace the whole template—not just the found text.

Closing the Current Function or Program

▶ From the **Actions** menu, click **Close**.

   If the function or program has unstored changes, you are prompted to check syntax and store before closing.

## *Running Programs and Evaluating Functions*

After defining and storing a program or function, you can use it from an application. All the applications can evaluate functions, but only the Calculator and Notes applications can run programs.

The program statements are executed in sequential order (although some commands alter the program flow). The output, if any, is displayed in the application's work area.

• Program execution continues until it reaches the last statement or a **Stop** command.

• Function execution continues until it reaches a **Return** command.

Running a Program or Function from the Program Editor

1. Make sure you have defined a program or function and the Program Editor is the active pane (computer) or page (handheld).

2. On the toolbar, click the **Document Tools** button ⚒ and select **Check Syntax & Store > Run**.
—or—
Press **Ctrl+R**.

   Handheld: Press menu [2] [3], or press ctrl [R].
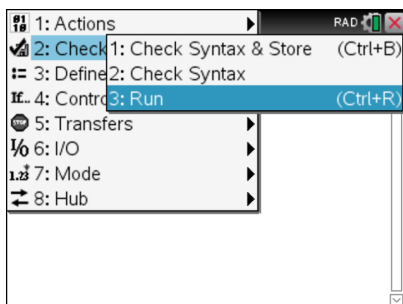
This will automatically:

- check the syntax and store the program or function,
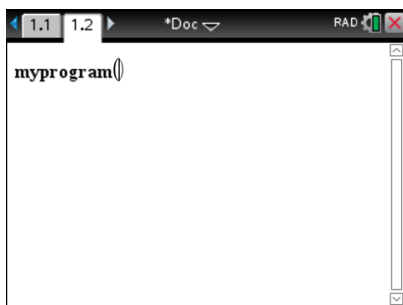- paste the program or function name on the first available line of the Calculator application immediately following the Program Editor. If no Calculator exists in that position, a new one is inserted.

3. If the program or function requires you to supply one or more arguments, type the values or variable names inside the parentheses.

4. Press [enter].

Note: You can also run a program or function in Calculator or Notes applications by typing the name of the program with parentheses and any required arguments and pressing [enter].

Using Short and Long Names

Anytime you are in the same problem where an object is defined, you can access it by entering its short name (the name given in the object's **Define** command). This is the case for all defined objects, including private, public, and non-library objects.

You can access a library object from any document by typing the object's long name. A long name consists of the name of the object's library document followed by a backslash "\" followed by the name of the object. For example, the long name of the

object defined as **func1** in the library document **lib1** is **lib1\func1**. To type the "\" character on the handheld, press ⇧shift ÷.

Note: If you cannot remember the exact name or the order of arguments required for a private library object, you can open the library document or use the Program Editor to view the object. You also can use **getVarInfo** to view a list of objects in a library.

Using a Public Library Program or Function

1. Make sure you have defined the object in the document's first problem, stored the object, saved the library document in the MyLib folder, and refreshed the libraries.

2. Open the TI-Nspire™ application in which you want to use the program or function.

   Note: All applications can evaluate functions, but only the Calculator and Notes applications can run programs.

3. Open the Catalog and use the library tab to find and insert the object.
   —or—
   Type the name of the object. In the case of a program or function, always follow the name with parentheses.

   libs2\func1()

4. If the program or function requires you to supply one or more arguments, type the values or variable names inside the parentheses.

   libs2\func1(34,power)

5. Press enter.

Using a Private Library Program or Function

To use a Private library object, you must know its long name. For example, the long name of the object defined as **func1** in the library document **lib1** is **lib1\func1**.

Note: If you cannot remember the exact name or the order of arguments required for a private library object, you can open the library document or use the Program Editor to view the object.

1. Make sure you have defined the object in the document's first problem, stored the object, saved the library document in the MyLib folder, and refreshed the libraries.

2. Open the TI-Nspire™ application in which you want to use the program or function.

   Note: All applications can evaluate functions, but only the Calculator and Notes applications can run programs.

3. Type the name of the object. In the case of a program or function, always follow the name with parentheses.

   libs2\func1()

4. If the object requires you to supply one or more arguments, type the values or variable names inside the parentheses.

```
libs2\func1(34,power)
```

5. Press enter .

Interrupting a Running Program or Function

While a program or function is running, the busy pointer $\odot$ is displayed.

▶ To stop the program or function,

-   Windows®: Hold down the **F12** key and press **Enter** repeatedly.
-   Mac®: Hold down the **F5** key and press **Enter** repeatedly.
-   Handheld: Hold down the ⌂ on key and press enter repeatedly.

    A message is displayed. To edit the program or function in the Program Editor, select **Go To**. The cursor appears at the command where the break occurred.

Getting Values into a Program

You can choose from several methods to supply the values that a function or program uses in calculations.

Embedding the Values Within the Program or Function

This method is useful primarily for values that must be the same each time the program or function is used.

1. Define the program.

```
Define calculatearea()=
Prgm
w:=3
h:=23.64
area:=w*h
EndPrgm
```

2. Run the program.

calculatearea() :area	70.92

Letting the User Assign the Values to Variables

A program or function can refer to variables created beforehand. This method requires users to remember the variable names and to assign values to them before using the object.

1. Define the program.

```
Define calculatearea()=
Prgm
area:=w*h
EndPrgm
```

2. Supply the variables, and then run the program.

---
w:=3 : h:=23.64
calculatearea() :area   70.92

Letting the User Supply the Values as Arguments

This method lets users pass one or more values as arguments within the expression that calls the program or function.

The following program, **volcyl**, calculates the volume of a cylinder. It requires the user to supply two values: height and radius of the cylinder.

1. Define the **volcyl** program.

Definevolcyl(height,radius) =
Prgm
Disp "Volume =", approx($\pi \bullet radius^2 \bullet height$)
EndPrgm
---

2. Run the program to display the volume of a cylinder with a height of 34 mm and a radius of 5 mm.

---
volcyl(34,5)   Volume = 534.071

   Note: You do not have to use the parameter names when you run the **volcyl** program, but you must supply two arguments (as values, variables, or expressions). The first must represent the height, and the second must represent the radius.

Requesting the Values from the User (Programs Only)

You can use the **Request** and **RequestStr** commands in a program to make the program pause and display a dialog box prompting the user for information. This method does not require users to remember variable names or the order in which they are needed.

You cannot use the **Request** or **RequestStr** command in a function.

1. Define the program.

Define calculatearea()=
Prgm
  Request "Width: ",w
  Request "Height: ",h
area:=w*h
EndPrgm
---

2. Run the program and respond to the requests.

---
calculatearea() : area
Width: 3   (3 entered as a response)
Height: 23.64 (23.64 entered as a response)
                                70.92

Use **RequestStr** instead of **Request** when you want the program to interpret the user's response as a character string rather than a math expression. This avoids requiring the user to enclose the response in quotation marks ("").

Displaying Information from a Function or Program

A running function or program does not display intermediate calculated results unless you include a command to display them. This is an important difference between performing a calculation on the entry line and performing it in a function or program.

The following calculations, for example, do not display a result in a function or program (although they do from the entry line).

⋮
x:=12•6
cos(π/4)
⋮

Displaying Information in the History

You can use the **Disp** command in a program or function to display information, including intermediate results, in the history.

⋮
Disp 12•6
Disp "Result:",cos(π/4)
⋮

Displaying Information in a Dialog Box

You can use the **Text** command to pause a running program and display information in a dialog box. The user clicks **OK** to continue or clicks **Cancel** to stop the program.

You cannot use the **Text** command in a function.

⋮
Text "Area=" & area
⋮

Note: Displaying a result with **Disp** or **Text** does not store that result. If you expect to refer later to a result, store it to a global variable.

⋮
cos(π/4)→maximum
Disp maximum
⋮

Using Local Variables

A local variable is a temporary variable that exists only while a user-defined function is being evaluated or a user-defined program is running.

Example of a Local Variable

The following program segment shows a **For...EndFor** loop (which is discussed later in this module). The variable i is the loop counter. In most cases, the variable i is used only while the program is running.

```
Local i ❶
For i,0,5,1
  Disp i
EndFor
Disp i
```

❶ Declares variable i as local.

Note: When possible, declare as local any variable that is used only within the program and does not need to be available after the program stops.

What Causes an Undefined Variable Error Message?

An **Undefined** variable error message is displayed when you evaluate a user-defined function or run a user-defined program that references a local variable that is not initialized (assigned a value).

For example:

```
Define fact(n)=Func
  Local m ❶
  While n>1
   n•m➜m: n–1➜n
  EndWhile
  Return m
  EndFunc
```

❶ Local variable m is not assigned an initial value.

Initialize Local Variables

All local variables must be assigned an initial value before they are referenced.

```
Define fact(n)=Func
  Local m: 1➜m ❶
  While n>1
   n•m➜m: n–1➜n
  EndWhile
  Return m
  EndFunc
```

❶ 1 is stored as the initial value for m.

Note (CAS): Functions and programs cannot use a local variable to perform symbolic calculations.

CAS: Performing Symbolic Calculations

If you want a function or program to perform symbolic calculations, you must use a global variable instead of a local. However, you must be certain that the global variable does not already exist outside of the program. The following methods can help.

- Refer to a global variable name, typically with two or more characters, that is not likely to exist outside of the function or program.

- Include **DelVar** within a program to delete the global variable, if it exists, before referring to it. (**DelVar** does not delete locked or linked variables.)

Differences Between Functions and Programs

A function defined in the Program Editor is similar to the functions built into the TI-Nspire™ Software.

- Functions must return a result, which can be graphed or entered in a table. Programs do not return a result.

- You can use a function (but not a program) within an expression. For example: **3 • func1(3)** is valid, but not **3 • prog1(3)**.

- You can run programs from Calculator and Notes applications only. However, you can evaluate functions in Calculator, Notes, Lists & Spreadsheet, Graphs & Geometry, and Data & Statistics.

- A function can refer to any variable; however, it can store a value to a local variable only. Programs can store to local and global variables.

   Note: Arguments used to pass values to a function are treated as local variables automatically. If you want to store to any other variables, you must declare them as **Local** from within the function.
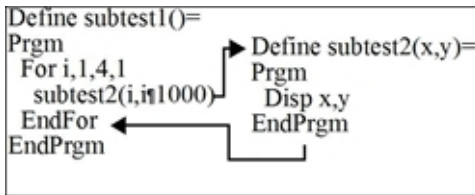
- A function cannot call a program as a subroutine, but it can call another user-defined function.

- You cannot define a program within a function.

- A function cannot define a global function, but it can define a local function.

Calling One Program from Another

One program can call another program as a subroutine. The subroutine can be external (a separate program) or internal (included in the main program). Subroutines are useful when a program needs to repeat the same group of commands at several different places.

Calling a Separate Program

To call a separate program, use the same syntax that you use to run the program from the entry line.

```
Define subtest1()=
Prgm                   ┌► Define subtest2(x,y)=
 For i,1,4,1           │  Prgm
  subtest2(i,i*1000)─┘    Disp x,y
 EndFor ◄──────────────┐ EndPrgm
EndPrgm                │
                       └────────┐
```

Defining and Calling an Internal Subroutine

To define an internal subroutine, use the **Define** command with **Prgm...EndPrgm**. Because a subroutine must be defined before it can be called, it is a good practice to define subroutines at the beginning of the main program.

An internal subroutine is called and executed in the same way as a separate program.

```
Define subtest1()=
 Prgm
 local subtest2        ❶
 Define subtest2(x,y)=  ❷
 Prgm
  Disp x,y
 EndPrgm
 ©Beginning of main program
 For i,1,4,1
  subtest2(i,I*1000)    ❸
 EndFor
 EndPrgm
```

❶ Declares the subroutine as a local variable.

❷ Defines the subroutine.

❸ Calls the subroutine.

Note: Use the Program Editor's **Var** menu to enter the **Define** and **Prgm...EndPrgm** commands.

Notes about Using Subroutines

At the end of a subroutine, execution returns to the calling program. To exit a subroutine at any other time, use **Return** with no argument.

A subroutine cannot access local variables declared in the calling program. Likewise, the calling program cannot access local variables declared in a subroutine.

Lbl commands are local to the programs in which they are located. Therefore, a **Goto** command in the calling program cannot branch to a label in a subroutine or vice versa.

Avoiding Circular-Definition Errors

When evaluating a user-defined function or running a program, you can specify an argument that includes the same variable that was used to define the function or create the program. However, to avoid circular-definition errors, you must assign a value for variables that are used in evaluating the function or running the program. For example:

---

x+1→x ❶

– or –

For i,i,10,1
  Disp i ❶
EndFor

---

❶ Causes a **Circular definition** error message if x or i does not have a value. The error does not occur if x or i has already been assigned a value.

Controlling the Flow of a Function or Program

When you run a program or evaluate a function, the program lines are executed in sequential order. However, some commands alter the program flow. For example:

• Control structures such as **If...EndIf** commands use a conditional test to decide which part of a program to execute.

• Loop commands such as **For...EndFor** repeat a group of commands.

Using If, Lbl, and Goto to Control Program Flow

The **If** command and several **If...EndIf** structures let you execute a statement or block of statements conditionally, that is, based on the result of a test (such as x>5). **Lbl** (label) and **Goto** commands let you branch, or jump, from one place to another in a function or program.

The **If** command and several **If...EndIf** structures reside on the Program Editor's **Control** menu.

When you insert a structure such as **If...Then...EndIf**, a template is inserted at the cursor location. The cursor is positioned so that you can enter a conditional test.

If Command

To execute a single command when a conditional test is true, use the general form:

If x>5
  Disp "x is greater than 5" ❶
Disp x ❷

---

❶ Executed only if x>5; otherwise, skipped.

❷ Always displays the value of x.

In this example, you must store a value to x before executing the **If** command.

If...Then...EndIf Structures

To execute one group of commands if a conditional test is true, use the structure:

```
If x>5 Then
  Disp "x is greater than 5"  ❶
  2•x→x  ❶
EndIf
Disp x  ❷
```

❶ Executed only if x>5.

❷ Displays the value of:
  2x if x>5
  x if x≤5

Note: EndIf marks the end of the **Then** block that is executed if the condition is true.

If...Then...Else...EndIf Structures

To execute one group of commands if a conditional test is true and a different group if the condition is false, use this structure:

```
If x>5 Then
  Disp "x is greater than 5"  ❶
  2•x→x  ❶
Else
  Disp "x is less than or equal to 5"  ❷
  5•x→x  ❷
EndIf
Disp x  ❸
```

❶ Executed only if x>5.

❷ Executed only if x≤5.

❸ Displays value of:
  2x if x>5
  5x if x≤5

If...Then...ElseIf... EndIf Structures

A more complex form of the **If** command lets you test for multiple conditions. Suppose you want a program to test a user-supplied argument that signifies one of four options.

To test for each option (If Choice=1, If Choice=2, and so on), use the **If...Then...ElseIf...EndIf** structure.

Lbl and Goto Commands

You can also control the flow by using **Lbl** (label) and **Goto** commands. These commands reside on the Program Editor's **Transfers** menu.

Use the **Lbl** command to label (assign a name to) a particular location in the function or program.

Lbl *labelName*	name to assign to this location (use the same naming convention as a variable name)

You can then use the **Goto** command at any point in the function or program to branch to the location that corresponds to the specified label.

Goto *labelName*	specifies which **Lbl** command to branch to

Because a **Goto** command is unconditional (it always branches to the specified label), it is often used with an **If** command so that you can specify a conditional test. For example:

```
If x>5
  Goto GT5   ❶
Disp x
--------
--------   ❷
Lbl GT5
Disp "The number was > 5"
```

❶ If x>5, branches directly to label GT5.

❷ For this example, the program must include commands (such as **Stop**) that prevent **Lbl** GT5 from being executed if x≤5.

Using Loops to Repeat a Group of Commands

To repeat the same group of commands successively, use one of the loop structures. Several types of loops are available. Each type gives you a different way to exit the loop, based on a conditional test.

Loop and loop-related commands reside on the Program Editor's **Control** and **Transfers** menus.

When you insert one of the loop structures, its template is inserted at the cursor location. You can then begin entering the commands that will be executed within the loop.

For...EndFor Loops

A **For...EndFor** loop uses a counter to control the number of times the loop is repeated. The syntax of the **For** command is:

For *variable, begin, end, increment*
      ❶   ❷   ❸   ❹

❶ Name of a variable to be used as a counter

❷ Value assigned to *variable* when the **For** loop begins.

❸ Value compared to the current value of *variable* at each iteration of the loop. The loop exits when *variable* exceeds *end*.

❹ Value added to *variable* at each iteration of the loop (This argument is optional. The default *increment* is 1.)

At each iteration of the **For** loop, the *variable* value is compared to the *end* value. If *variable* does not exceed *end*, the commands within the **For...EndFor** loop are executed and the loop repeats; otherwise, control jumps to the command following **EndFor**.
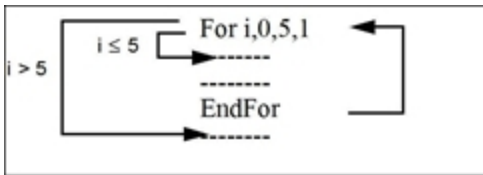
Note: The **For** command automatically increments the counter variable so that the function or program can exit the loop after a certain number of repetitions.

At the end of the loop (**EndFor**), control loops back to the **For** command, where the counter variable is incremented and compared to *end*.

For example:

```
For i,0,5,1
  Disp i ❶
EndFor
Disp i ❷
```

❶ Displays 0, 1, 2, 3, 4, and 5.

❷ Displays 6. When *variable* increments to 6, the loop is not executed.

Notes:

- You can declare *variable* as local if it does not need to be saved after the function or program stops.

- You can set *end* to a value less than *begin*, provided you also set *increment* to a negative value.

While...EndWhile Loops

A **While...EndWhile** loop repeats a block of commands as long as a specified condition is true. The syntax of the **While** command is:

While *condition*

When **While** is executed, *condition* is evaluated. If *condition* is true, the loop is executed; otherwise, control jumps to the command following **EndWhile**.
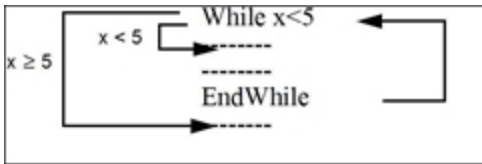
Note: The **While** command does not automatically change the condition. You must include commands that allow the function or program to exit the loop.

At the end of the loop (**EndWhile**), control jumps back to the **While** command, where condition is re-evaluated.

To execute the loop the first time, the condition must initially be true.

- Any variables referenced in the condition must be set before the **While** command. (You can build the values into the function or program, or you can prompt the user to enter the values.)

- The loop must contain commands that change the values in the condition, eventually causing it to be false. Otherwise, the condition is always true and the function or program cannot exit the loop (called an infinite loop).
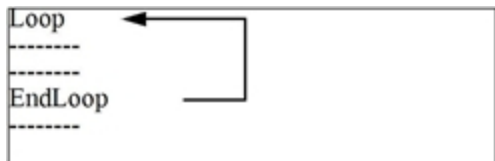
For example:

```
0→x  ❶
While x<5
 Disp x  ❷
 x+1→x  ❸
EndWhile
Disp x  ❹
```

❶ Initially sets x.

❷ Displays 0, 1, 2, 3, and 4.

❸ Increments x.

❹ Displays 5. When x increments to 5, the loop is not executed.

Loop...EndLoop Loops

A **Loop...EndLoop** creates an infinite loop, which is repeated endlessly. The **Loop** command does not have any arguments.

Typically, you insert commands in the loop that let the program exit from the loop. Commonly used commands are: **If**, **Exit**, **Goto**, and **Lbl** (label). For example:

```
0→x
Loop
  Disp x
  x+1→x
  If x>5      ❶
   Exit
EndLoop
Disp x        ❷
```

❶ An **If** command checks the condition.

❷ Exits the loop and jumps to here when x increments to 6.

Note: The **Exit** command exits from the current loop.

In this example, the **If** command can be anywhere in the loop.

| When the If command is:	The loop is:
At the beginning of the loop	Executed only if the condition is true.
At the end of the loop	Executed at least once and repeated only if the condition is true.

The **If** command could also use a **Goto** command to transfer program control to a specified **Lbl** (label) command.

Repeating a Loop Immediately

The **Cycle** command immediately transfers program control to the next iteration of a loop (before the current iteration is complete). This command works with **For...EndFor**, **While...EndWhile**, and **Loop...EndLoop**.

Lbl and Goto Loops

Although the **Lbl** (label) and **Goto** commands are not strictly loop commands, they can be used to create an infinite loop. For example:


```
Lbl START  ◄─────────┐
--------              │
--------              │
Goto START  ──────────┘
--------
```
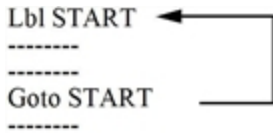
As with **Loop...EndLoop**, the loop should contain commands that let the function or program exit from the loop.

Changing Mode Settings

Functions and programs can use the **setMode()** function to temporarily set specific calculation or result modes. The Program Editor's **Mode** menu makes it easy to enter the correct syntax without requiring you to memorize numeric codes.

Note: Mode changes made within a function or program definition do not persist outside the function or program.

Setting a Mode

1. Position the cursor where you want to insert the **setMode** function.

2. From the **Mode** menu, click the mode to change, and click the new setting.

    The correct syntax is inserted at the cursor location. For example:

   setMode(1,3)

Debugging Programs and Handling Errors

After you write a function or program, you can use several techniques to find and correct errors. You can also build an error-handling command into the function or program itself.

If your function or program allows the user to select from several options, be sure to run it and test each option.

Techniques for Debugging

Run-time error messages can locate syntax errors but not errors in program logic. The following techniques may be useful.

•   Temporarily insert **Disp** commands to display the values of critical variables.

•   To confirm that a loop is executed the correct number of times, use **Disp** to display the counter variable or the values in the conditional test.

- To confirm that a subroutine is executed, use **Disp** to display messages such as "Entering subroutine" and "Exiting subroutine" at the beginning and end of the subroutine.

- To stop a program or function manually,

  - Windows®: Hold down the **F12** key and press **Enter** repeatedly.

  - Mac®: Hold down the **F5** key and press **Enter** repeatedly.

  - Handheld: Hold down the 🏠on key and press enter repeatedly.

Error-handling Commands

Command	Description
Try...EndTry	Defines a block that lets a function or program execute a command and, if necessary, recover from an error generated by that command.
ClrErr	Clears the error status and sets system variable errCode to zero. For an example of using errCode, see the Try command in the *Reference Guide*.
PassErr	Passes an error to the next level of the **Try...EndTry** block.

What's New

What's New in Version4.4

Restricted Domains

- Allows the user to plot higher order polynomial relations and restrict the relations.
- Examples:
    - $x^3 + y^3 - 6*x*y = 0$
    - y=sin(x) and *-2π < x ≤ 2p
    - y≤ x^2 | y ≥ -2 and 0 ≤ x ≤ 3
    - $x^2 + y^2$ ≤ 3, y ≥ 0 and x ≤ 0

Sequence Graphing

- Allows the user to plot sequences in the form u(n+1)=, u(n+2), u(n+k), where k is an explicit positive integer.

Inverse Binomial (invBinom())

- Given the number of trials ($NumTrials$) and the probability of success of each trial ($Prob$), this function returns the minimum number of successes, k, such that the cumulative probability of k successes is greater than or equal to the given cumulative probability ($CumulativeProb$).

Inverse Binomial with respect to N (invBinomN())

- Given the probability of success of each trial ($Prob$), and the number of successes (NumSuccess), this function returns the minimum number of trials, N, such that the cumulative probability of x successes is less than or equal to the given cumulative probability ($CumulativeProb$).

Widgets

- A Widget is a .tns document that is stored in your MyWidgets folder.
- You can use Widgets to:
    - Easily access text files
    - Insert and run scripts (such as Stopwatch)
    - Quickly insert a saved problem into a document
- When you add a Widget, TI-Inspire™ CX extracts only the first page of the selected .tns file, and inserts it into your open document.

Using Sensor Data in Programmes

- You can access sensor data from all connected sensor probes in your TI-Basic programme by using this command:

    RefreshProbeVars statusVar

- The *RefreshProbeVars* command will be valid only when Vernier DataQuest™ is launched and is in 'meter' mode.

To report issues and help us improve our apps, please email us at ti-cares@ti.com.

Graphing Relations

Relation graphing is available on Graphs pages and in the Analytic Window of Geometry pages.

You can define relations using $\leq$, $<$, $=$, $>$, or $\geq$. The inequality operator ($\neq$) is not supported in relation graphing.

Relation type	Examples
Equations and inequalities equivalent to y = f(x)	• y = sqrt(x) • y-sqrt(x) = 1/2 • -2*y-sqrt(x) = 1/2 • y-sqrt(x) $\geq$ 1/2 • -2*y-sqrt(x) $\geq$ 1/2
Equations and inequalities equivalent to x = g(y)	• x = sin(y) • x-sin(y) = 1/2 • x-sin(y) $\geq$ 1/2
Polynomial equations and inequalities	• x^2+y^2 = 5 • x^2-y^2 $\geq$ 1/2+y • x^3+y^3-6*x*y=0
The above relations on domains restricted by rectangles	• y=sin(x) and -2π<x$\leq$2π • y$\leq$x2|y$\geq$-2 and 0$\leq$x$\leq$3 • {$x^2+y^2$$\leq$3, y$\geq$0 and x$\leq$0

Note: Restrictions imposed by an active Press-to-Test session may limit the types of relations you can graph.

To Graph a Relation:

1. From the **Graph Entry/Edit** menu, select **Relation**.

2. Type an expression for the relation.

3. Press **Enter** to graph the relation.

Tips for Graphing Relations

▶ You can quickly define a relation from the Function entry line. Position the cursor to the immediate right of the = sign, and then press the **Backspace** key. A small menu appears with the relation operators and a **Relation** option. Choosing from the menu places the cursor in the Relation entry line.

▶ You can type a relation as text on a Graphs page and then drag the text object over either axis. The relation is graphed and added to the relation history.

Warning and Error Message

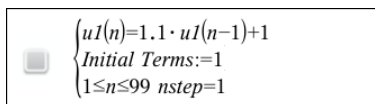| Error Condition	Additional Information
Relation input not supported	**Relation input not supported** **Note:** The following relation inputs are supported: • Relations using ≤, <, =, >, or ≥. • Polynomial relations in x and y • Relations equivalent to y=f(x) or x=g(y) or corresponding inequalities • The above relations on domains restricted by rectangles
Domain Restrictions not supported for certain classes of relations equivalent to y=f(x) or x=g(y) or corresponding inequalities.	• Relations equivalent to y=f(x) and corresponding inequalities can only have constraints on x • For example: y=√(x) and 0≤x≤1 will work but y=√(x) and 0≤y≤1 will not • Relations equivalent to x=g(y) and corresponding inequalities can only have constraints on y • For example: x=sin(y)|−1≤y≤1 will work but x=sin(y)|−1≤x≤1 will not

Plotting Sequences

The Graphs application lets you plot two types of sequences. Each type has a separate template for defining the sequence.
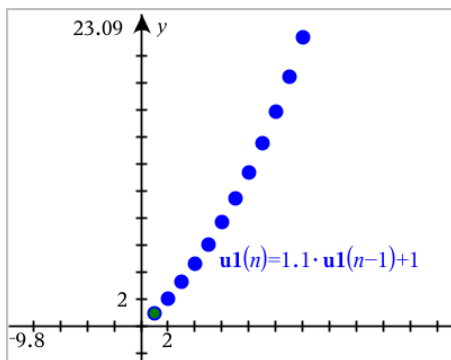
Defining a Sequence

1. From the **Graph Entry/Edit** menu, select **Sequence > Sequence**.

$$\begin{cases} u1(n)= \\ Initial\ Terms:= \\ 1 \le n \le 99\ \ nstep=1 \end{cases}$$

2. Type the expression to define the sequence. Update the independent variable field to m+1, m+2, etc., if necessary.

3. Type an initial term. If the sequence expression references more than one prior term, such as $u1(n-1)$ and $u1(n-2)$, (or $u1(n)$ and $u1(n+1)$), separate the terms with commas.

$$\begin{cases} u1(n)=1.1 \cdot u1(n-1)+1 \\ Initial\ Terms:=1 \\ 1 \le n \le 99\ \ nstep=1 \end{cases}$$

4. Press **Enter**.

Defining a Custom Sequence

A custom sequence plot shows the relationship between two sequences by plotting one on the x axis and the other on the y axis.

This example simulates the Predator-Prey model from biology.

1. Use the relations shown here to define two sequences: one for a rabbit population, and another for a fox population. Replace the default sequence names with **rabbit** and **fox**.

$\begin{cases} \textbf{rabbit}(n)=\textbf{rabbit}(n-1)\cdot\left(1+0.05-0.001\cdot\textbf{fox}(n-1)\right) \\ \textit{Initial Terms}:=200 \\ 1\leq n\leq400\ \textit{nstep}=1 \end{cases}$

$\begin{cases} \textbf{fox}(n)=\textbf{fox}(n-1)\cdot\left(1+2.\textbf{E}\text{-}4\cdot\textbf{rabbit}(n-1)-0.03\right) \\ \textit{Initial Terms}:=50 \\ 1\leq n\leq400\ \textit{nstep}=1 \end{cases}$

.05 = the growth rate of rabbits if there are no foxes
.001 = the rate at which foxes can kill rabbits
.0002 = the growth rate of foxes if there are rabbits
.03 = the death rate of foxes if there are no rabbits

Note: If you want to see the plots of the two sequences, zoom the window to the **Zoom - Fit** setting.

2. From the **Graph Entry/Edit** menu, select **Sequence > Custom**.

3. Specify the **rabbit** and **fox** sequences to plot on the x and y axes, respectively.

p1 $\begin{cases} x\leftarrow\textbf{rabbit}(n) \\ y\leftarrow\textbf{fox}(n) \\ 1\leq n\leq400\ \textit{nstep}=1 \end{cases}$

4. Press **Enter** to create the custom plot.

5. Zoom the window to the **Zoom - Fit** setting.

6. Explore the custom plot by dragging the point that represents the initial term.

Distributions

Calculating a Distribution

Example: Calculate a distribution to fit the Normal Pdf distribution model.

1. Click the column formula cell (second cell from the top) in column A.

2. Click **Statistics > Distributions > Normal Pdf** to choose the Distribution model.
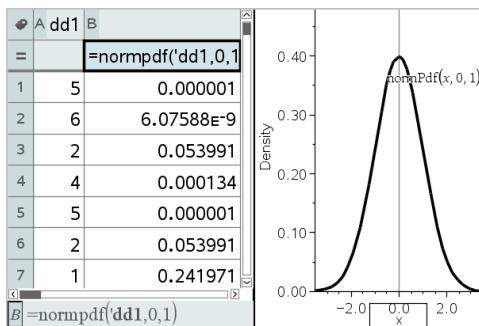
    The Normal Pdf dialog box opens and displays fields for typing or selecting the arguments for the calculation.

3. Press **Tab** as necessary to move from field to field and provide each argument. You can type values, or select them from the drop down list:

    -   **X Value:** Click the drop-down arrow to choose any list in the problem to provide the x values for the calculation.

    -   **Mean:** Type a value for the mean or click the drop-down arrow to choose a variable that contains the mean.

    -   **Standard Deviation:** Type a value for the standard deviation or choose a variable that contains the standard deviation.

4. Click the **Draw** check box to see the distribution plotted in Data & Statistics.

    Note: The Draw option is not available for all distributions.

5. Click **OK**.

    Lists & Spreadsheet inserts two columns: one containing the names of the results, and one containing the corresponding values. The results are plotted in Data & Statistics.

Note: The results are linked to the source data. For example, you can change a value in Column A, and the equation updates automatically.

Supported Distribution Functions

The following distributions are available from the Lists & Spreadsheet application. For more information regarding these functions, see the *TI-Nspire™ Reference Guide*.

• To return a single distribution result based on a single value, type the function in a single cell.

• To return a list of distribution results based on a list of values, type the function in a column formula cell. In this case, you specify a list (column) that contains the values. For each value in the list, the distribution returns a corresponding result.

Note: For distribution functions that support the draw option (**normPDF**, **t PDF**, χ^2 **Pdf**, and $\overline{F}$ **Pdf**), the option is available only if you type the distribution function in a formula cell.

Normal Pdf (normPdf)

Computes the probability density function (**pdf**) for the normal distribution at a specified x value. The defaults are mean μ=0 and standard deviation σ=1. The probability density function (pdf) is:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \sigma > 0$$

This distribution is used to determine the probability of the occurrence of a certain value in a normal distribution. The draw option is available when Normal PDF is invoked from a formula cell.

When you access distributions from the formula cell, you must select a valid list from the drop-down list to avoid unexpected results. If accessed from a cell, you must specify a number for the x-value. The distribution returns the probability that the value you specify will occur.

Normal Cdf (normCdf)

Computes the normal distribution probability between *Lower Bound* and *Upper Bound* for the specified mean, μ (default=0) and the standard deviation, σ (default=1). You can click the **Draw (Shade area)** check box to shade the area between the lower and upper bounds. Changes to the initial *Lower Bound* and *Upper Bound* automatically update the distribution.

This distribution is useful in determining the probability of an occurrence of any value between the lower and upper bounds in the normal distribution. It is equivalent to finding the area under the specified normal curve between the bounds.

Inverse Normal (invNorm)

Computes the inverse cumulative normal distribution function for a given *area* under the normal distribution curve specified by mean, μ, and standard deviation, σ.

This distribution is useful in determining the x-value of data in the area from 0 to x<1 when the percentile is known.

t Pdf (tPdf)

Computes the probability density function (**pdf**) for the t-distribution at a specified x value. df (degrees of freedom) must be > 0. The probability density function (**pdf**) is:

$$f(x) = \frac{\Gamma[(df+1)/2]}{\Gamma(df/2)} \quad \frac{(1+x^2/df)^{-(df+1)/2}}{\sqrt{\pi df}}$$

This distribution is useful in determining the probability of the occurrence of a value when the population standard deviation is not known and the sample size is small. The draw option is available when **t Pdf** is invoked from a formula cell.

t Cdf (tCdf)

Computes the Student-t distribution probability between *Lower Bound* and *Upper Bound* for the specified df (degrees of freedom). You can click the **Draw (Shade area)** check box to shade the area between the bounds. Changes to the initial *Lower Bound* and *Upper Bound* automatically update the distribution.

This distribution is useful in determining the probability of the occurrence of a value within an interval defined by the lower and upper bound for a normally distributed population when the population standard deviation is not known.

Inverse t (invt)

Computes the inverse cumulative t-distribution probability function specified by Degrees of Freedom, df, for a given area under the curve.

This distribution is useful in determining the probability of an occurrence of data in the area from 0 to x<1. This function is used when the population mean and/or population standard deviation is not known.

χ^2 *Pdf (χ^2 *Pdf())*

Computes the probability density function (**pdf**) for the χ^2 (chi-square) distribution at a specified x value. df (degrees of freedom) must be an integer > 0. The probability density function (**pdf**) is:

$$f(x) = \frac{1}{\Gamma(df/2)}(1/2)^{df/2}x^{df/2-1}e^{-x/2}, x \geq 0$$

This distribution is useful in determining the probability of the occurrence of a given value from a population with a χ^2 distribution. The draw option is available when χ^2 **Pdf** is invoked from a formula cell.

χ^2 *Cdf (χ^2 *Cdf())*

Computes the χ^2 (chi-square) distribution probability between *lowBound* and *upBound* for the specified df (degrees of freedom). You can click the **Draw Shade area** check box to shade the area between the lower and upper bounds. Changes to the initial *lowBound* and *upBound* automatically update the distribution.

This distribution is useful in determining the probability of the occurrence of value within given boundaries of a population with a χ^2 distribution.

$\mathbb{F}$ *Pdf ($\mathbb{F}$ *Pdf())*

Computes the probability density function (**pdf**) for the $\mathbb{F}$ distribution at a specified x value. *numerator df* (degrees of freedom) and *denominator df* must be integers > 0. The probability density function (**pdf**) is:

$$f(x) = \frac{\Gamma[(n+d)/2]}{\Gamma(n/2)\Gamma(d/2)}\left(\frac{n}{d}\right)^{n/2}x^{n/2-1}(1+nx/d)^{-(n+d)/2}, x \geq 0$$

> where n = numerator degrees of freedom
> d = denominator degrees of freedom

This distribution is useful in determining the probability that two samples have the same variance. The draw option is available when $\mathbb{F}$ Pdf is invoked from a formula cell.

$\mathbb{F}$ *Cdf ($\mathbb{F}$ *Cdf())*

Computes the $\mathbb{F}$ distribution probability between *lowBound* and *upBound* for the specified *dfnumer* (degrees of freedom) and *dfDenom*. You can click the **Draw (Shade area)** check box to shade the area between the lower and upper bounds. Changes to the initial *lowBound* and *upBound* automatically update the distribution.

This distribution is useful in determining the probability that a single observation falls within the range between the lower bound and upper bound.

Binomial Pdf (binomPdf())

Computes a probability at x for the discrete binomial distribution with the specified *numtrials* and probability of success (p) on each trial. The x parameter can be an integer or a list of integers. $0 \leq p \leq 1$ must be true. *numtrials* must be an integer > 0. If you do not specify x, a list of probabilities from 0 to *numtrials* is returned. The probability density function (**pdf**) is:

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x}, x = 0, 1, \ldots, n$$

where $n = numtrials$

This distribution is useful in determining the probability of success in a success/failure trial, at trial n. For example, you could use this distribution to predict the probability of getting heads in a coin toss on the fifth toss.

Binomial Cdf (binomCdf())

Computes a cumulative probability for the discrete binomial distribution with n number of trials and probability p of success on each trial.

This distribution is useful in determining the probability of a success on one trial before all trials are completed. For example, if heads is a successful coin toss and you plan to toss the coin 10 times, this distribution would predict the chance of obtaining heads at least once in the 10 tosses.

Inverse Binomial (invBinom())

Given the number of trials (*NumTrials*) and the probability of success of each trial (*Prob*), this function returns the minimum number of successes, k, such that the cumulative probability of k successes is greater than or equal to the given cumulative probability (*CumulativeProb*).

This distribution is useful in determining the upper bound input of the binomial cdf. For example, if you flip a coin 10 times, and you want the probability of getting x heads or less to be greater than 75%, this distribution will help determine what x should be.

Inverse Binomial with respect to N (invBinomN())

Given the probability of success of each trial (*Prob*), and the number of successes (NumSuccess), this function returns the minimum number of trials, N, such that the cumulative probability of x successes is less than or equal to the given cumulative probability (*CumulativeProb*).

Poisson Pdf (poissPdf())

Computes a probability at x for the discrete Poisson distribution with the specified mean, μ, which must be a real number > 0. x can be an integer or a list of integers. The probability density function (**pdf**) is:

$$f(x) = e^{-\mu}\mu^{x}/x!, x = 0,1,2,...$$

This distribution is useful in determining the probability of obtaining a certain number of successes before a trial begins. For example, you could use this calculation to predict the number of heads that would occur in eight tosses of a coin.

Poisson Cdf (poissCdf())

Computes a cumulative probability for the discrete Poisson distribution with specified mean, $\bar{x}$.

This distribution is useful in determining the probability that a certain number of successes occur between the upper and lower bounds of a trial. For example, you could use this calculation to predict the number of heads displayed between coin toss #3 and toss #8.

Geometric Pdf (geomPdf())

Computes a probability at x, the number of the trial on which the first success occurs, for the discrete geometric distribution with the specified probability of success p. $0 \leq p \leq 1$ must be true. x can be an integer or a list of integers. The probability density function (pdf) is:

$$f(x) = p(1-p)^{x-1}, x = 1,2,...$$

This distribution is useful in determining the likeliest number of trials before a success is obtained. For example, you could use this calculation to predict the number of coin tosses that would be made before a heads resulted.

Geometric Cdf (geomCdf())

Computes a cumulative geometric probability from lowBound to upBound with the specified probability of success, p.

This distribution is useful in determining the probability associated with the first success occurring during trials 1 through n. For example, you could use this calculation to determine the probability that heads display on toss #1, #2, #3, ..., #n.

Widgets

All work that you create and save using TI-Nspire™ applications is stored as a document, which you can share with others using TI-Nspire™ software, a TI-Nspire™ handheld, or the TI-Nspire™ App for iPad®. You save these TI-Nspire™ documents as .tns files.

A Widget is a .tns document that is stored in your MyWidgets folder.

You can use Widgets to:

• Easily access text files

• Insert and run scripts (such as the pre-loaded widget example: Stopwatch.tns)

• Quickly insert a saved problem into a document

When you add a Widget, TI-Inspire™ CX extracts only the first page of the selected .tns file, and inserts it into your open document.

Creating a Widget

A document is regarded as a Widget when it is saved or copied to the designated MyWidgets folder. The default location is:

• Windows®: My Documents\TI-Nspire\MyWidgets.

• Mac®: Documents/TI-Nspire/MyWidgets.

• Handheld: MyWidgets

• TI-Nspire™ App for iPad® and TI-Nspire™ CAS App for iPad®: MyWidgets

If the MyWidget folder has been inadvertently deleted, you must create it before attempting to use a Widget.

Note: When you add a Widget, TI-Inspire™ CX extracts only the first page of the selected .tns file, and inserts it into your open document.

Adding a Widget

Adding a Widget to a New Document

1. Open a new document.

2. Click **Add Widget**.
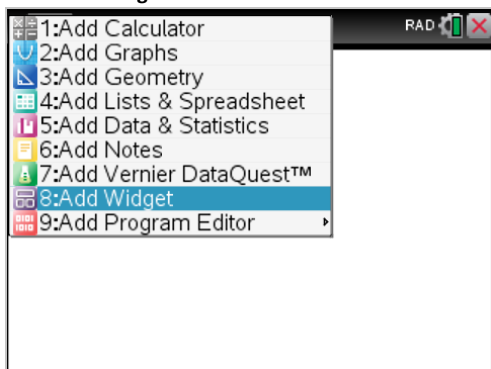
3. Scroll to select a .tns file from the box.
4. Click **Add**.

Note: Stopwatch is a preloaded .tns file. Any saved .tns file will show up in this list.
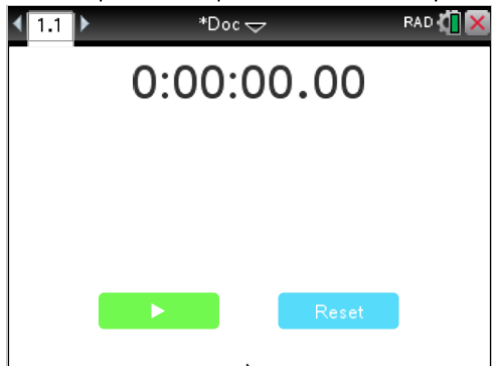
Adding a Widget to an Existing Document

1. Click the ▼ on **Doc▼** > **Insert** > **Widget**.
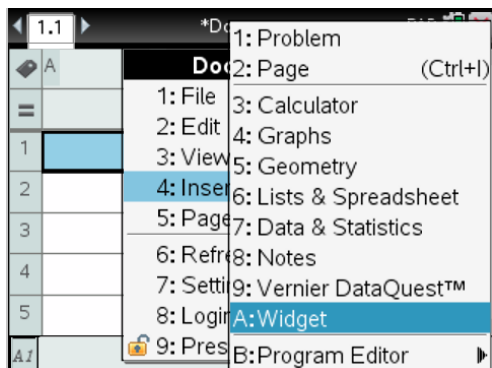
2. Click **Add**.
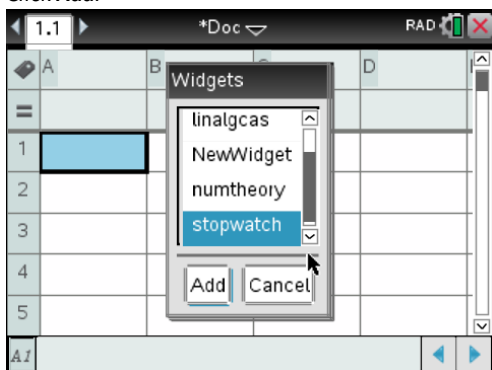
Note: You can also add a Widget to a new or existing document using the **Insert** menu.

Saving a Widget

1. Click .

2. Navigate to MyDocuments > MyWidgets.

3. Type in a name for your Widget.
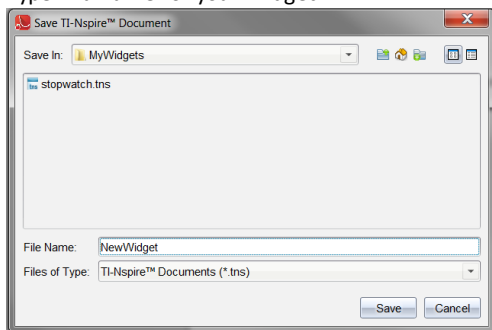
4. Click **Save**.

Using Sensor Data in Programs

You can access sensor data from all connected sensor probes through your TI-Basic program by using this command:

RefreshProbeVars statusVar

•   You must first launch the Vernier DataQuest™ application, or you will receive an error.

   Note: The Vernier DataQuest™ application will auto-launch when you connect a sensor or a lab cradle to the TI-Nspire™ software or handheld.

•   The *RefreshProbeVars* command will be valid only when Vernier DataQuest™ is in 'meter' mode.

•   *statusVar* is an optional parameter that indicates the status of the command. These are the *statusVar* values:

StatusVar Value	Status
statusVar =0	Normal (continue with the program)
statusVa	The Vernier DataQuest™ application is in data collection mode.

StatusVar Value	Status
r=1	**Note:** The Vernier DataQuest™ application must be in meter mode for this command to work. ⊕
statusVar =2	The Vernier DataQuest™ application is not launched.
statusVar =3	The Vernier DataQuest™ application is launched, but you have not connected any probes.

- Your TI-Basic program will read directly from Vernier DataQuest™ variables in the symbol table.

- The meter.time variable shows the last value of the variable; it does not update automatically. If no data collection has occurred, meter.time will be 0 (zero).

- Use of variable names without corresponding probes being physically attached will result in a "Variable not defined" error.

Collecting Sensor Data using RefreshProbeVars

1. Launch the Vernier DataQuest™ application.

2. Connect the sensor(s) you need to collect the data.

3. Run the program you wish to use to collect data in the calculator application.

4. Manipulate the sensors and collect the data.

Note: You may create a program to interact with the TI-Innovator Hub using menu **> Hub > Send**. (See Example 2, below.) This is optional.

Example 1

```
Define temp()=
Prgm
© Check if system is ready
RefreshProbeVars status
If status=0 Then
Disp "ready"
For n,1,50
RefreshProbeVars status
temperature:=meter.temperature
Disp "Temperature: ",temperature
If temperature>30 Then
Disp "Too hot"
EndIf
© Wait for 1 second between samples
Wait 1
EndFor
Else
Disp "Not ready. Try again later"
EndIf
EndPrgm
```

Example 2- with TI-Innovator™ Hub

```
Define tempwithhub()=
Prgm
© Check if system is ready
RefreshProbeVars status
If status=0 Then
Disp "ready"
For n,1,50
RefreshProbeVars status
temperature:=meter.temperature
Disp "Temperature: ",temperature
If temperature>30 Then
Disp "Too hot"
© Play a tone on the Hub
Send "SET SOUND 440 TIME 2"
EndIf
© Wait for 1 second between samples
Wait 1
EndFor
Else
Disp "Not ready. Try again later"
EndIf
EndPrgm
```

What's New

What's New in Version 4.2

To report issues and help us improve our applications, please email us at ti-cares@ti.com.

New View Option for using Capture Class

• You can capture handheld screens automatically at 15-second intervals. This interval is the new default setting for Auto Refresh.

Prepare Handhelds Remotely

• Teachers can change the settings of connected handhelds from their computer.

• The handhelds can be connected through wired or wireless connections.

• You can remotely clear the Scratchpad, apply document settings, and enter and exit Press-to-Test sessions on the handhelds.

Variables - Sliders

• In the Graphs application, you are automatically offered sliders when you enter certain functions, equations, or sequences that refer to undefined variables.

• You can now add sliders to a Notes page.

Relation Graphing

• You can graph relations using ≤, <, =, >, and ≥.

• Relations can be:

  - Equations and inequalities equivalent to y = f(x), such as y = sqrt(x)

  - Equations and inequalities equivalent to x = g(y), such as x-sin(y) ≥ 1/2

  - Conic equations and inequalities, such as x^2-y^2 ≥ 1/2+y

• Relation graphing is available in the ScratchPad, the Graphs application, and the Analytic Window of the Geometry application.

Math Actions

• Math Actions are available on Notes, Scratchpad, and Calculator pages.

• When you display the context menu for a selected expression or equation, you might see available Math Actions, such as Solve Numerically, Try to Factor, and Complete the Square.

• Each action may prompt you for any needed parameters.

Graphing from Notes and Calculator

• You can graph a function or relation directly from its context menu.

• When possible, the graph appears on the same page as the selected function or relation.

- Available for many functions and relations on Notes, Scratchpad, and Calculator pages.

Customizing the Graph in Data Collection

- You can press **Esc** to remove the selection line from graphed data.

Setting View Options in Capture Class

You have several options for optimizing the view of captured class screens, including:

- Toggling between a tiled view and a gallery view.

- Comparing selected screens in a side-by-side view.

- Refreshing the view as needed or setting the auto-refresh feature to automatically refresh the view at timed intervals.

- Zooming in or out to increase or decrease the size of the captured screens in the window. You can also select the percentage at which the size of the screen will be increased or decreased.

- Showing or hiding student names.

- Accessing Live Presenter options. For more information, see *Using Live Presenter*.

Toggling Between Tiled and Gallery Views

When you capture student screens, the default view for the Class Screen Capture window is Tiled. Switch to Gallery view to show an enlarged view of a selected screen. Thumbnail views of all the captured screens are shown in the left pane, enabling you to find and select screens.

To toggle between Tiled and Gallery views:

▶ Click **View** 🔳, and then click **Gallery** or **Tiled**.

—or—

▶ From the **View** menu, select **Gallery** or **Tiled**.

—or—

▶ For Gallery view, press Ctrl + G. (Mac®: ⌘ + G). For Tiled view, press Ctrl + T (Mac®: ⌘ + T)**.**

In Gallery view, thumbnail images of captured screens are displayed in a column in the left pane of this view. An enlarged view of a selected screen is displayed in the right pane.
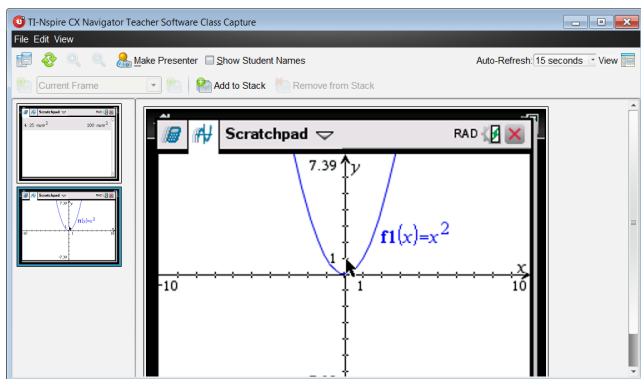
- If you select multiple thumbnails to view in the left pane, they are displayed in the right pane.

- If you select more than four thumbnails, view the additional screens by scrolling down the right pane.

- If you selected to show student names, names are centered under each student's captured screen.

- Zoom functions are disabled in Gallery View.

- You can change the order of the screens in the Gallery View.

- The width of the thumbnail pane cannot be changed.

Rearranging Captured Screens

Changing the order of captured screens in one view (Tiled or Gallery) automatically changes it in the other view.

▶ In Tiled view, drag the screen to its new position.

▶ In Gallery view, drag the screen's thumbnail up or down to its new position.

As you drag, a vertical bar shows the point at which the screen will be inserted.

Note: To cancel the move before releasing the mouse button, press **Esc**.

Refreshing Captured Screens

As students log in to class and work on their screens, you might want to update the captured screens.

Refreshing Screens Manually

▶ Click **Refresh** 🔄.

> **Note:** If a student has lost connection and a screen cannot be refreshed, it is outlined in yellow.
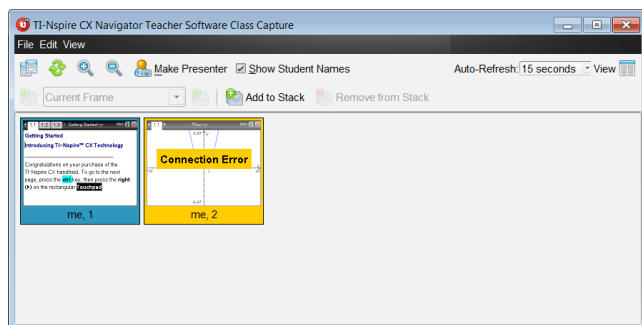
Refreshing Screens Automatically

The auto-refresh feature recaptures class screens at timed intervals. The default setting for the auto-refresh feature is Off. To enable auto-refresh:

▶ On the **View** menu, click **Auto Refresh**, and select a time interval.
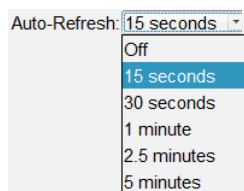
> **Note:** The minimum interval for TI-Nspire™ Navigator™ NC Teacher Software is 30 seconds.

Zooming in and Zooming out

In the Tiled view only, you can magnify or shrink the views of the captured screens.

▶ To zoom to levels between 50% and 400% in 50% increments:

Click Zoom In 🔍 or Zoom Out 🔍 on the toolbar.
–or–
Click **View > Zoom In** or **View > Zoom Out** from the menu.

▸ To zoom to a selected level, click **View > Zoom Percentage**, and select the level.
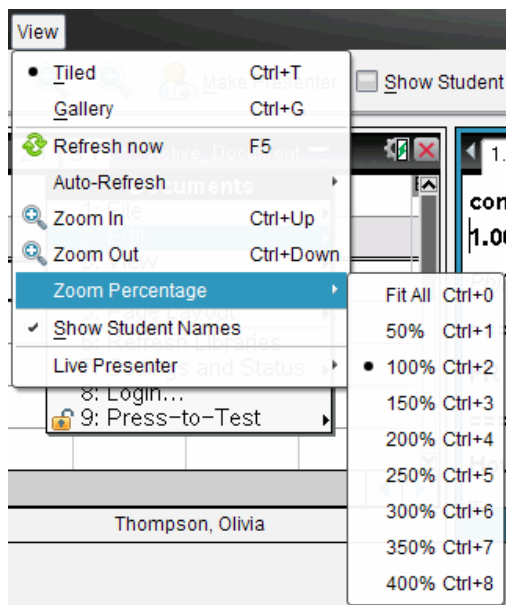
Preparing Handhelds Remotely

TI-Nspire™ CX Teacher Software and TI-Nspire™ CX Navigator™ Teacher Software let you change settings on connected handhelds from your computer. The handhelds can be connected through wired or wireless connections.

You don't need to start a class session or require students to log in. You can perform these operations while other transfer activities, such as Quick Poll or Live Presenter, are in progress.

Clearing the Scratchpad

This removes all Scratchpad data from the connected handhelds, including the Scratchpad Calculator pages and Graphs & Geometry pages, the Scratchpad history and variables, and the contents of each handheld's clipboard. The handhelds can be in or out of Press-to-Test mode.

WARNING: This action clears the Scratchpad and Clipboard data from *all currently connected handhelds*. You cannot limit it to a selected set of handhelds.

1. From the **Tools** menu, select **Prepare Handhelds > Clear Scratchpad**.

A Clear Scratchpad confirmation message appears.

2. To proceed, click **Clear Scratchpad**.

   A status log reports the progress. A notification message appears on each handheld.

Configuring Document Settings

This lets you apply document settings, such as Angle unit and Calculation Mode, to the connected handhelds. It gives you a quick way to put all the handhelds into a known state. You can choose to save configurations for reuse. The handhelds can be in or out of Press-to-Test mode.

Note: The document settings are applied immediately to handheld Scratchpads and are saved as defaults for the Scratchpad and all new documents. Existing documents, whether open or closed, retain their prior settings.

1. From the **Tools** menu, select **Prepare Handhelds > Set Default Document Settings**.

   The Document Settings dialog box opens, showing the setting that will be applied.

2. Click each option that you want to change, and select the desired setting.

   - To apply a previously saved configuration, click the **Saved Configuration** list, and select the configuration.

   - To restore factory default settings, click the **Saved Configuration** list, and select **TI Factory Default**.

3. (Optional) To save the current settings as a named configuration, click **Save**, type a **Name** into the Save Configuration dialog box, and click **Save**.

4. Click **Send**.

   A status log reports the progress. A notification message appears on each handheld.

Entering Press-to-Test Mode on the Handhelds

This lets you start a Press-to-Test session on the connected handhelds and apply the same set of restrictions to each. You can choose to save configurations for reuse.

WARNING: *All currently connected handhelds* will restart, including any handhelds that are already in a Press-to-Test session. You cannot limit this action to a selected set of handhelds.

1. From the **Tools** menu, select **Prepare Handhelds > Enter Press-to-Test**.

The Press-to-Test Configuration dialog box opens, showing the restrictions that will be applied.

2. Click the check box for each restriction to select or deselect it.

3. (Optional) To save the current settings as a named configuration, click **Save**, type a **Name** into the Save Configuration dialog box, and click **Save**.

4. Click **Enter Press-to-Test**.

   A warning message lists the effects of entering a Press-to-Test session.

5. To proceed, click **Enter Press-to-Test**.

   A status log reports the progress. Each handheld preserves its current files and settings and then restarts in Press-to-Test mode.

Exiting Press-to-Test mode

This closes the Press-to-Test session on the connected handhelds and restarts them.

WARNING: *All currently connected handhelds* will exit Press-to-Test and restart. All data and files created while in Press-to-Test will be deleted. You cannot limit this action to a selected set of handhelds.

1. Notify the students that you are about to close the Press-to-Test session on their handhelds.

2. From the **Tools** menu, select **Prepare Handhelds > Exit Press-to-Test**.

   A warning message lists the effects of exiting a Press-to-Test session.

3. To proceed, click **Exit Press-to-Test**.

   A status log reports the progress. Each handheld restarts with its previous files and settings restored.

Adjusting Variable Values with a Slider

A slider control lets you interactively adjust or animate the value of a numeric variable. You can insert sliders in the Graphs, Geometry, Notes, and Data & Statistics applications.
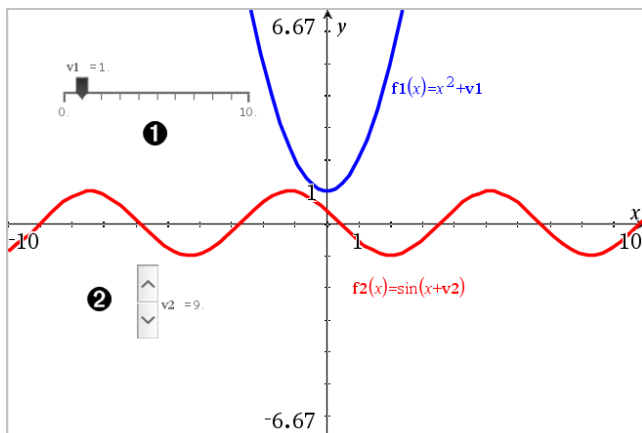
❶ Horizontal slider for adjusting variable $v1$.

❷ Minimized vertical slider for adjusting variable $v2$.

Note: TI-Nspire™ version 4.2 or higher is required for opening .tns files containing sliders on Notes pages.

Inserting a Slider Manually

1. From a Graphs, Geometry, or Data & Statistics page, select **Actions > Insert Slider**.
   —or—
   From a Notes page, make sure the cursor is not in a math box or chem box, and then select **Insert > Insert Slider**.

   The Slider Settings screen opens.

2. Enter the desired values, and click **OK**.

   The slider is displayed. On a Graphs, Geometry, or Data & Statistics page, handles are displayed to let you move or stretch the slider.
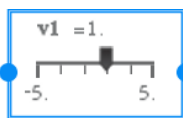
To remove the handles and use the slider, click an empty space in the work area. You can show the handles anytime by selecting **Move** from the slider's context menu.

3. To adjust the variable, slide the pointer (or click the arrows on a minimized slider).

   - You can use the **Tab** key to move the focus to a slider or to move from one slider to the next. The color of the slider changes to show you when it has the focus.
   - When a slider has the focus, you can use the arrow keys to change the value of the variable.

Working with the Slider

Use the options on the context menu to move or delete the slider, and to start or stop its animation. You can also change the slider's settings.

1. Display the slider's context menu.
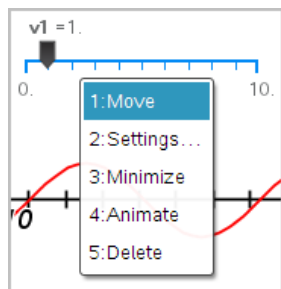
2. Click an option to select it.

Automatic Sliders in Graphs

Sliders can be created for you automatically in the Graphs application and in the analytic window of the Geometry application. You are offered automatic sliders when you define certain functions, equations, or sequences that refer to undefined variables.

Graphing Relations

Relation graphing is available on Graphs pages and in the Analytic Window of Geometry pages.

You can define relations using ≤, <, =, >, or ≥. The inequality operator (≠) is not supported in relation graphing.

Relation type	Examples
Equations and inequalities equivalent to y = f(x)	y = sqrt(x) y-sqrt(x) = 1/2 -2*y-sqrt(x) = 1/2 y-sqrt(x) ≥ 1/2 -2*y-sqrt(x) ≥ 1/2
Equations and inequalities equivalent to x = g(y)	x = sin(y) x-sin(y) = 1/2 x-sin(y) ≥ 1/2
Conic equations and inequalities	x^2+y^2 = 5 x^2-y^2 ≥ 1/2+y

Note: Restrictions imposed by an active Press-to-Test session may limit the types of relations you can graph.

To Graph a Relation:

1. From the **Graph Entry/Edit** menu, select **Relation**.

2. Type an expression for the relation.

3. Press **Enter** to graph the relation.

Tips for Graphing Relations

▶ You can quickly define a relation from the Function entry line. Position the cursor to the immediate right of the = sign, and then press the **Backspace** key. A small menu appears with the relation operators and a **Relation** option. Choosing from the menu places the cursor in the Relation entry line.

▶ You can type a relation as text on a Graphs page and then drag the text object over either axis. The relation is graphed and added to the relation history.

Using Math Actions

Math Actions are available on Notes, Scratchpad, and Calculator pages.

When you display the context menu for a selected expression or equation, the menu may include a **Math Actions** submenu that lists the available actions. Each action might prompt you for any needed parameters.

The specific math actions listed depend on:

• The type of expression or relation.

• The operating system in use (numeric or CAS).

• Any restrictions imposed by an active Press-to-Test session.

Example of Math Actions in Notes

1. Insert a math box, and type the equation $x^2+3x+1=0$, but don't press **Enter** yet.

Numeric OS

CAS OS

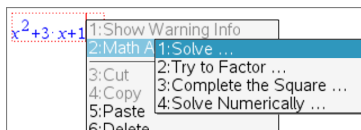2. Display the context menu of the equation, and select **Math Actions**.
   Windows®: Right-click the equation.
   Mac®: Hold ⌘, and click the equation.
   Handheld: Point to the equation, and press ⌈ctrl⌉ ⌈menu⌉ .
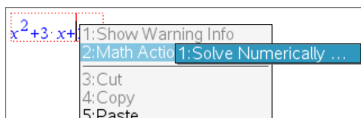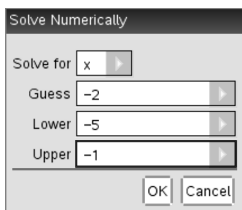

Numeric OS

CAS OS

3. Select the action to perform:

   - **Solve Numerically** for numeric OS.

   - **Solve** for CAS OS.

   You are prompted to enter parameters. For example, Numeric Solve prompts for the variable of interest, initial guess, lower bound, and upper bound.

4. Type a value for each parameter. When options are available, you can click an arrow to make a selection.

Numeric OS

CAS OS

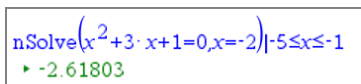5. Click **OK** to construct the completed expression and place it in the math box.
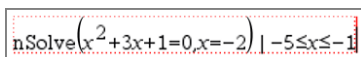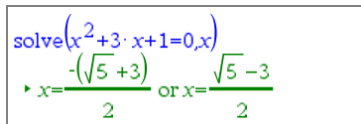

Numeric OS
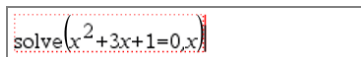
CAS OS

6. Press **Enter** to complete the action.

Numeric OS

CAS OS

7. As a further exploration, drag through the math box to select $x^2+3\cdot x+1$. Do not include the "=0" portion.

Numeric OS

CAS OS

8. Display the context menu for the selected text, select **Math Actions > Find Roots of Polynomial**, and press **Enter** to complete the action.

    The action and its result are shown in a new math box.

Numeric OS

CAS OS

Tips for Using Math Actions in Notes

▶ For a previously evaluated expression, click in the expression and then display its context menu.

    When you select an action, it replaces the expression.

▶ For a displayed result, click in the result and then display its context menu.

    When you select an action, it appears in a new math box.

▶ For a portion of an expression or result, select the portion, and then display the context menu.

    When you select an action, it appears in a new math box.

Graphing from Notes and Calculator

You can graph a function or relation directly from its context menu. This feature is available for many functions and relations on Notes, Scratchpad, and Calculator pages.
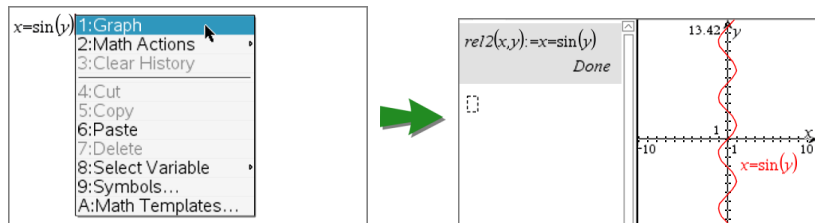
If page layout options allow, the graph appears on the same page as the function or relation. Otherwise, the graph appears on a separate Graphs page.

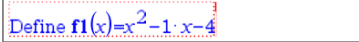The type of graph created depends on:

- The type of function or relation.
- Any restrictions imposed by an active Press-to-Test session.

Example of Graphing from Notes

This example uses a Notes page to explore a quadratic function interactively.

1. Insert a math box on a new Notes page, and enter the following function definition:

   ```
   Define f1(x)=x^2-1·x-4
   ```
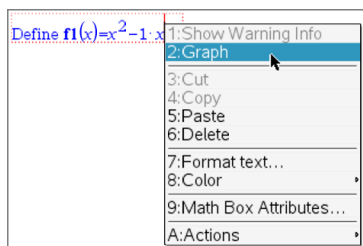
   $$\text{Define } \mathbf{f1}(x) = x^2 - 1 \cdot x - 4$$

2. Display the context menu of the Define statement.
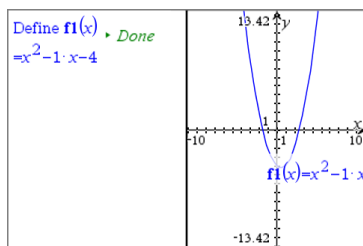   Windows®: Right-click the statement.
   Mac®: Hold ⌘, and click the statement.

   Handheld: Point to the statement, and press [ctrl] [menu].

3. Select **Graph** from the context menu.

   The graph appears. The graph and the math box are linked so that any adjustment to one affects the other.

4. Explore the relationship between the defined function and its graph:

   - Drag the ends or center of the graph to manipulate it, and observe the changes to the function definition.

     —or—

- Edit the defined function in the math box, and observe the changes to the graph.
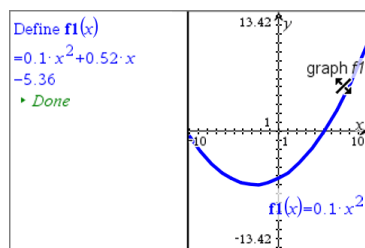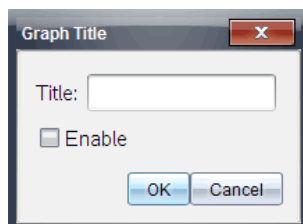

Customizing the Graph of Collected Data

You can customize the Graph view by adding a title, changing colors, and setting ranges for the axis.
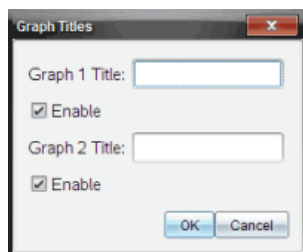
Adding a Title

When you add a title to a graph, the title is displayed in the View Details area. When you print the graph, the title prints on the graph.

1. Click **Graph > Graph Title**.

   The Graph Title dialog box opens.

   If there are two graphs in the work area, the dialog box has two title options.

2. Type the name of the graph in the Title field.

    —or—

    a) Type the name of the first graph in the Graph 1 field.
    b) Type the name of the second graph in the Graph 2 field.

3. Select **Enable** to show the title.

    Note: Use the Enable option to hide or show the graph title as needed.
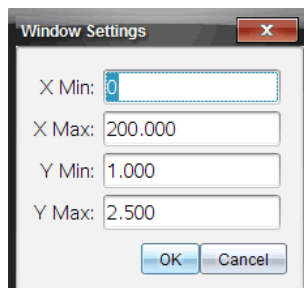
4. Click **OK**.

    The title is shown.

Setting Axis Ranges

Setting Axis Ranges for One Graph

To modify the minimum and maximum range for the x and y axis:

1. Click **Graph > Window Settings**.

    The Window Settings dialog box opens.

2. Type the new values in one or more of these fields:

    -   X Min
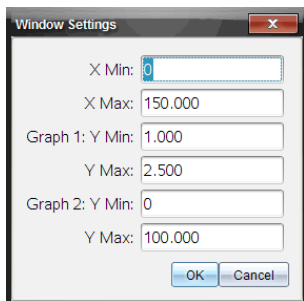   -   X Max
    -   Y Min
    -   Y Max

3. Click **OK**.

The application uses the new values for the graph visual range until you modify the range or change data sets.

Setting Axis Ranges for Two Graphs

When working with two graphs, enter two y axis minimum and maximum values, but only one set of minimum and maximum values for the x axis.

1. Click **Graph > Window Setting**.

    The Window Setting dialog box opens.

2. Type the new values in one or more of these fields:

    -   X Min

    -   X Max

    -   Graph 1: Y Min

    -   Y Max
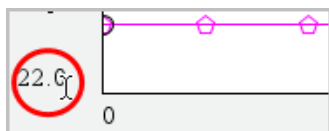
    -   Graph 2: Y Min

    -   Y Max

3. Click **OK**.

    The application uses the new values for the graph visual range until you modify the range or change data sets.

Setting the Axis Range on the Graph Screen

You can modify the minimum and maximum range for the x and y axes directly on the graph screen.

▶   Select the axis value that you want to change, and type a new value.

The graph is redrawn to reflect the change.

Selecting which Data Sets to Plot

1. In the Detail view on the left, click the tab immediately below the view selection buttons.
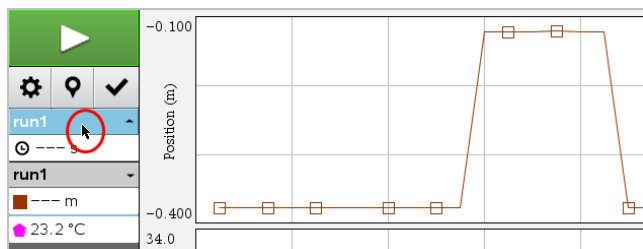
2. The Detail view shows a list of available data sets.

3. Use the check boxes to select the data sets to plot.
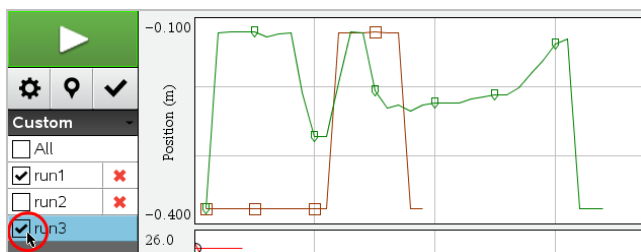
Autoscaling a Graph

Use the autoscale option to show all the points plotted. Autoscale Now is useful after you change the x and y axis range or zoom in or out of a graph. You can also define the automatic autoscale setting to use during and after a collection.

Autoscale Now Using the Application Menu

▶ Click **Graph > Autoscale Now**.

The graph now displays all the points plotted.

Autoscale Now Using the Context Menu

1. Open the context menu in the graph area.

2. Click **Window/Zoom > Autoscale Now**.

    The graph now displays all the points plotted.

Defining Autoscale During a Collection

There are two options for using the automatic autoscaling that occurs during a collection. To choose an option:

1. Click **Options > Autoscale Settings**.

    The Autoscale Settings dialog box opens.
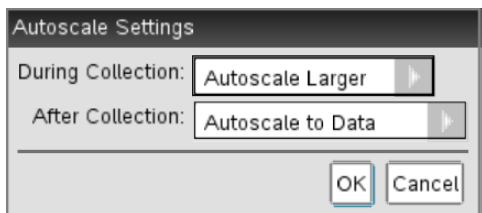
2. Click ► to open the During Collection drop-down list.

3. Select one of these options:

    • **Autoscale Larger** - Expands the graph as needed to show all points as you collect them.

    • **Do Not Autoscale** - The graph is not changed during a collection.

4. Click **OK** to save the setting.

Defining Autoscale After a Collection

You have three options for setting the automatic autoscaling that occurs after a collection. To set your choice:

1. Click **Options > Autoscale Settings**.

    The Autoscale Settings dialog box opens.

2. Click ► to open the **After Collection** drop-down list.

3. Select one of these options:

- **Autoscale to Data.** Expands the graph to show all data points. This option is the default mode.
- **Autoscale From Zero.** Modifies the graph so all data points including the origin point are displayed.
- **Do Not Autoscale.** The graph settings are not changed.

4. Click **OK** to save the setting.

Selecting a Range of Data

Selecting a range of data on the graph is useful in several situations, such as when zooming in or out, striking and unstriking data, and examining settings.

To select a range:

1. Drag across the graph.

   The selected area is indicated by gray shading.

2. Perform one of these actions.

- Zoom in or out
- Strike or unstrike data
- Examine settings

To deselect a range:

▶ Press the **Esc** key as necessary to remove the shading and the vertical trace line.

Zooming In on a Graph

You can zoom in on a subset of the collected points. You can also zoom out from a previous zoom or expand the graph window beyond the data points collected.

To zoom in on a graph:

1. Select the area you want to zoom into, or use the current view.

2. Click **Graph > Zoom In**.

   The graph adjusts to display only the area you selected.

The x range selected is used as the new x range. The y range autoscales to show all graphed data points in the selected range.

Zooming Out of a Graph

▶ Select **Graph > Zoom Out**.

The graph is now expanded.

If a Zoom In precedes a Zoom Out, the graph displays the original settings prior to the Zoom In.

For example, if you Zoomed In twice, the first Zoom Out would display the window of the first Zoom In. To display the full graph with all data points from multiple zoom ins, use Autoscale Now.

Setting Point Options

To indicate how often marks show on the graph and whether to use a connecting line:

1. Click **Options > Point Options**.

   The Point Options dialog box opens.
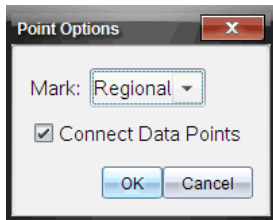
2. Select a **Mark** option from the drop-down list.

   • **None.** No point protectors.
   • **Regional.** Periodic point protectors.
   • **All.** Every data point as a point protector.

3. Select **Connect Data Points** to display a line between points.

   —or—

   Clear **Connect Data Points** to remove the line between points.

The following graphics show examples of some of the Point Mark options.

Regional

Connect Data Points - Selected

All

Connect Data Points - not selected

None

Connect Data Points - Selected
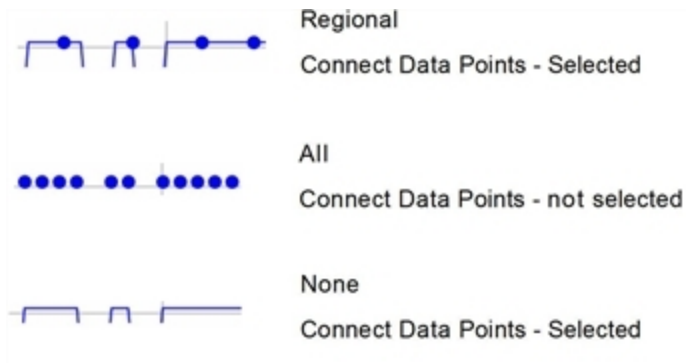
Changing a Graph's Color

1. Click the point indicator for the graph whose color you want to change.
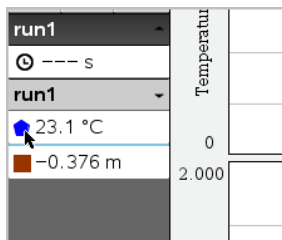
2. In the Column Options dialog box, select the new **Color**.

Selecting Point Markers

1. Right-click in the graph to open the menu.

2. Click **Point Marker**.

    Note: If there is only one dependent variable column, the Point Marker option is preceded by the data set name and column name. Otherwise, the Point Marker option has a menu.

3. Select the column variable to change.

4. Select the point marker to set.

    The Point Marker changes to the option selected.

Selecting an Independent Variable Column

Use the option Select X-axis Column to select the column used as the independent variable when graphing the data. This column is used for all graphs.

1. Click **Graph > Select X-axis Column**.

2. Select the variable you want to change.

    The x-axis label on the graph changes and the graph is reordered using the new independent variable for graphing the data.

Selecting a Dependent Variable Column

Use the option Select Y-axis Column to select which dependent variable columns to plot on the displayed graph(s).

1. Click **Graph > Select Y-axis Column**.

2. Select one of the following:

    • A variable from the list. The list is a combination of dependent variables and the number of data sets.

    • **More.** Selecting More opens the Select dialog box. Use this when you want to select a combination of data set variables to graph.
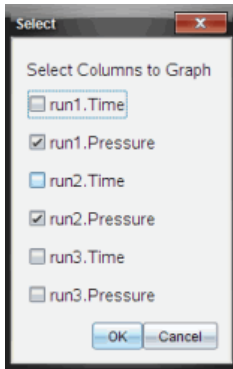
Showing and Hiding Details

You can hide or show the Details view on the left side of the screen.

▶ Click **Options > Hide Details** or **Options > Show Details**.

What's New

What's New in Version 4.0

To report issues and help us improve our applications, please email us at ti-cares@ti.com.

Working with Documents - Angle Mode Indicator

• An indicator now shows the angle mode (Degrees, Radians, or Gradians) in effect for the current application. On handhelds, the indicator resides at top of the document screen. In desktop software, the indicator resides in the document status bar.

Working with Documents - Zoom to Fit

• When using the handheld preview of a document, you can make the document zoom automatically to fit the window size.

Working with Documents - Set default page size

• You can now set the default page size as either Computer or Handheld for new documents.

Graphs - Exact Window Settings

• You can use expressions such as 7/3 or 2•π for exact input of custom window settings. The values appear in exact form on the axes and in the Window Settings dialog the next time you display it.

  - On TI-Nspire™ products, fractional input is preserved as-is. Other exact inputs are replaced with the evaluated results.
  - On TI-Nspire™ CAS products, fractional and other exact inputs are preserved.

Geometry - Force Triangle Angles to Integers

• A new setting lets you restrict the angles of a geometric triangle to integer values as you create or edit the triangle. This helps to minimize instances of triangles whose angles do not add up "visually" to 180 degrees.

3D Graphing - Orthographic View

• You can customize the 3D environment to show 3D graphs in either Orthographic Projection or Perspective View.

Graphs and Geometry - Math Draw

- A new tool, Math Draw, lets you use touchscreen or mouse gestures to create points, lines, circles, and other shapes. For example, if you draw an approximately circular shape, the tool interprets it as a circle.

Graphs and Geometry - Automatically Label Points

- A new setting lets you automatically apply labels (A, B, ..., Z, A_1, B_1, and so on) to points, lines, and vertices of geometric shapes as you draw them.

Graphs and Geometry - Directed Angle Measurements

- A new measurement tool lets you create directed angle measurements. You can set clockwise or counterclockwise orientation.

Graphs and Geometry - Bounded Area Selection

- When prompted to select two curves while defining a bounded area, you can optionally click one curve and the x axis.

Graphs, Geometry, and 3D Graphing - Views Preserved

- When you switch among the Graphs, Geometry, and 3D Graphing views, the most recently used window/view settings for the selected view are applied. You can also undo and redo view changes.

Data & Statistics - Display Digits and Diagnostics

- A new Display Digits setting lets you select the display format for numeric labels in the current document.

- A new Diagnostics setting lets you display the value of the r^2 or R^2 statistic (when available) under certain regression equations.
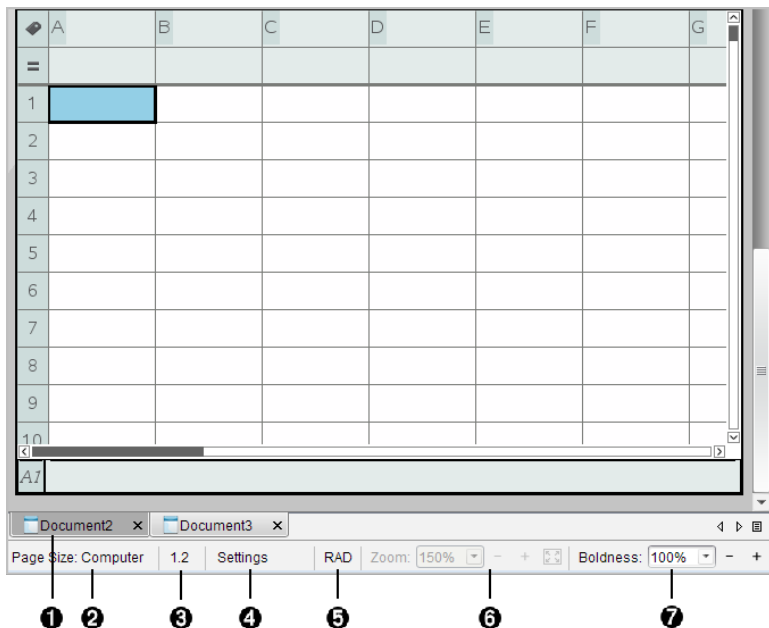
Lua Script Editor - Zoom Text

- You can now adjust the size of the text in your script and in any of the tool panels.

Working with Applications

When you first open a new document or add a new problem to a document, you select an application from a menu.

The following illustration shows how a document containing the Lists & Spreadsheet application appears in the work area.

❶ **Document name**. Tabs show the names of open documents. Click a name to make it the active document.

❷ **Page Size.** Shows the document's page size as Handheld or Computer. You can use the TI-Nspire™ **File** menu to convert a document from one page size to the other.

❸ **Problem/Page counter**. Labels the problem number and page number of the active page. For example, a label of **1.2** identifies Problem **1**, Page **2**.

❹ **Settings**. Double-click to view or change the Document Settings for the active document or to change the default Document Settings.

❺ **Angle Mode**. Shows an abbreviation of the angle mode (Degrees, Radians, or Gradians) in effect. Hover the pointer over the indicator to see the full name.

❻ **Zoom**. Enabled in Handheld preview only (click **Document Preview** on the toolbar and select **Handheld**). Click ▼ and select a magnification value, or click the zoom-to-fit button 🔲 to make the preview adapt automatically to window size.

➐ **Boldness**. Enabled in Computer preview only (click **Document Preview** on the toolbar and select **Computer**). Click ▼ and select a value to increase or decrease the boldness of text and other items.

Working with Multiple Applications on a Page

You can add up to four applications to a page. When you have multiple applications on a page, the menu for the active application is displayed in the Documents Toolbox. Using multiple applications involves two steps:

• Changing the page layout to accommodate multiple applications.

• Adding the applications.

You can add multiple applications to a page even if an application is already active.

Adding Multiple Applications to a Page

By default, each page contains space to add one application. To add additional applications to the page, complete the following steps.

1. Click **Edit > Page Layout > Select Layout**.
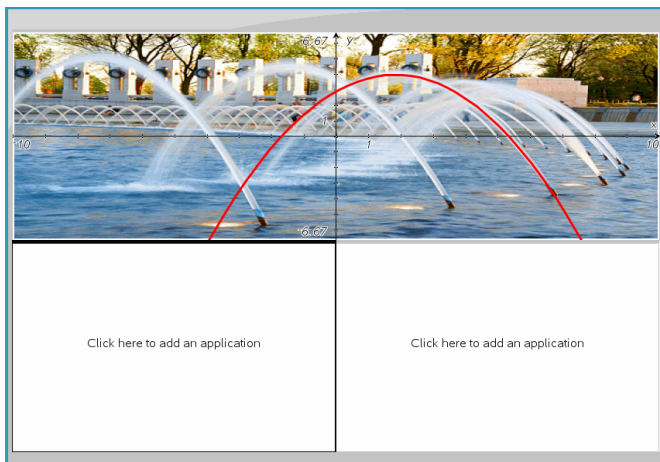
   –or–

   Click .

   The page layout menu opens.

There are eight page layout options available. If an option is already selected, it is dimmed.

2. Highlight the layout you want to add to the problem or page, and then click to select it.

The new layout is displayed with the first application active.

3. In Handheld preview, click **Press menu** to select an application for each new section in the problem or page. In Computer view, select **Click here to add an application**.

Swapping Applications

To change the position of applications on a page with multiple applications, "swap" the positions of two applications.

1. Click **Edit > Page Layout > Swap Application**.

    Note: The last active application you worked on is automatically selected as the first application to be swapped.

2. Click the second application to swap.

    This action performs the swap.

    Note: When there are only two work areas, the selected application automatically swaps position with the other application in the work area.

To cancel a swap, press **Esc**.

Setting Page Size and Document Preview

When you ceate a document, you specify its page size as Handheld or Computer, depending on how you expect the document to be used. Documents of both page sizes can be opened on either platform, and you can

convert the page size anytime.

- **Handheld.** Size: 320 × 217 pixels, fixed. Handheld documents can be viewed on all platforms. You can magnify (zoom) the content when viewing it on a tablet or larger screen.
- **Computer.** Size: 640 × 434 pixels, minimum. Computer documents scale up automatically to take advantage of higher resolution screens. The minimum size is 640 × 434, so some content may be clipped on handheld devices.

Note: You can view documents of either page size using Handheld or Computer preview.

Converting the Current Document's Page Size

▶ On the main TI-Nspire™ **File** menu, select **Convert to**, and then select the page size.

The software saves the current document and creates a copy that uses the requested page size.

Viewing the Document in Handheld Preview

1. On the application toolbar, click **Document Preview**, and select **Handheld**.

   The preview changes. This does not change the document's underlying page size.

2. (Optional) Adjust the viewing magnification:

   - Click the **Zoom** tool beneath the work area, and select a magnification value.

     –or–

   - Click the **Zoom to Fit** button ⛶ to make the handheld preview adjust automatically to the window size.

Viewing the Document in Computer Preview

1. On the application toolbar, click **Document Preview**, and select **Computer**.

   The preview changes. This does not change the document's underlying page size.

2. (Optional) Click the **Boldness** tool beneath the work area, and select a value to increase or decrease the boldness of text and other items.

Setting the Default Page Size for New Documents

1. On the main TI-Nspire™ **File** menu, select **Settings > Page Size Settings**.

2. Select a default page size, either Handheld or Computer.

    The new size applies to documents that you create (Windows®: **Ctrl+C**, Mac®: **Cmd+C**) after setting the default, including the blank document created automatically each time you open the software. Changing the default setting does not convert any currently open documents or other existing documents.

Setting a Default Preview

By default, when you open a document, it is automatically displayed using the preview that matches its page size. You can override this rule and specify a preview that you prefer.

1. On the main TI-Nspire™ **File** menu, select **Settings > Preview Settings**.

2. Select the preview that you want documents to use when you open them.

Zooming/Rescaling the Graphs Work Area

Rescaling in the Graphs application affects only the graphs, plots, and objects that reside in the Graphing view. It has no effect on objects in the underlying Plane Geometry view.

Rescaling by Dragging Along an Axis

▸ To rescale the x and y axes proportionally, drag a tic mark on either axis.

▸ To rescale only one axis, hold down **Shift** and drag a tic mark on the axis.

Zooming Using a Zoom Tool

▸ From the **Window / Zoom** menu, select one of the tools.

   -   **Zoom - Box** (Click two corners of a box to define the area to show.)
   -   **Zoom - In**
   -   **Zoom - Out**

Zooming to Predefined Settings

▸ From the **Window / Zoom** menu, select one of the predefined Zoom settings.

   -   **Zoom - Standard**
   -   **Zoom - Quadrant 1**
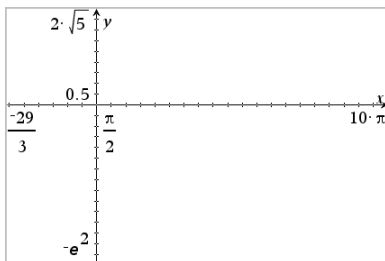
- **Zoom - Standard User**
- **Zoom - Standard Trig**
- **Zoom - Standard Data**
- **Zoom - Fit**

Entering Custom Window Settings

1. From the **Window / Zoom** menu, select **Window Settings**.

2. Enter a value for each setting. You can use expressions for exact input, as shown below.
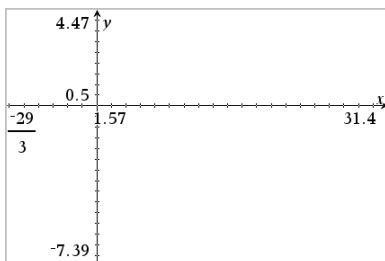
On TI-Nspire™ products, fractional input is preserved as-is. Other exact inputs are replaced with the evaluated result.

On TI-Nspire™ CAS products, fractional and other exact inputs are preserved.

What You Must Know

Changing the Graphs and Geometry Settings

1. From the **Settings** menu in the Documents Toolbox, select **Settings**.

2. Select the settings that you want to use.

    -   **Display Digits.** Sets the display format for numbers as Floating or Fixed decimal.

    -   **Graphing Angle.** Sets the angle unit for all Graphs and 3D Graphing applications in the current docuument. The default setting is Radian. Set this to Auto if you want graphing angles to follow the Angle setting in the main **File > Settings** menu. An angle mode indicator shows the resulting mode in Graphs and 3D Graphing applications.

    -   **Geometry Angle.** Sets the angle unit for all Geometry applications in the current document. The default setting is Degree. Set this to Auto if you want geometry angles to follow the Angle setting in the main **File > Settings** menu. An angle mode indicator shows the resulting mode in Geometry applications.

    -   **Automatically hide plot labels.** In the Graphs application, hides the label that normally appears next to a graphed relation.

    -   **Show axis end values.** Applies only in the Graphs application.

    -   **Show tool tips for function manipulation.** Applies only in the Graphs application.

    -   **Automatically find points of interest.** In the Graphs application, shows zeros, minima, and maxima while tracing function graphs.

    -   **Force Geometric Triangle Angles to Integers.** Restricts the angles of a triangle to integer values as you create or edit the triangle. This setting applies only in the Geometry View with the Geometry Angle unit set to Degree or Gradian. It does not apply to analytic triangles in Graphing View or to analytic triangles in the Analytic Window of the Geometry View. This setting does not affect existing angles, and it does not apply when constructing a triangle based on previously inserted points. By default, this setting is deselected.

    -   **Automatically Label Points.** Applies labels (A, B, ..., Z, A_1, B_1, and so on) to points, lines, and vertices of geometric shapes as you draw

them. The labeling sequence starts at A for each page in a document. By default, this setting is deselected.

Note: If you create a new object that uses existing unlabeled points, those point are not automatically labeled in the completed object.

- Click **Restore** to restore all settings to their factory defaults.
- Click **Make Default** to apply the current settings to the open document and save them as the default for new Graphs and Geometry documents.

Using Context Menus

Context menus provide quick access to commonly used commands and tools that apply to a specific object. For example, you can use a context menu to change an object's line color or to group a set of selected objects.

▶ Display the context menu for an object in one of the following ways.

- Windows®: Right-click the object.
- Mac®: Hold ⌘ and click the object.
- Handheld: Move the pointer to the object, and then press [ctrl] [menu].

Finding Hidden Objects in the Graphs or Geometry Application

You can hide and show individual graphs, geometric objects, text, labels, measurements, and axis end-values.

To temporarily view hidden graphs or objects or to restore them as shown objects:

1. From the **Actions** menu, select **Hide/Show**.

   The Hide/Show tool 👁 appears in the work area, and all hidden objects become visible in dimmed colors.

2. Click a graph or object to toggle its Hide/Show state.

3. To apply the changes and close the Hide/Show tool, press **ESC**.

Inserting a Background Image

You can insert an image as a background for a Graphs or Geometry page. The file format of the image can be .bmp, .jpg, or .png.

1. From the **Insert** menu, click **Image**.

2. Navigate to the image you want to insert, select it, and then click **Open**.
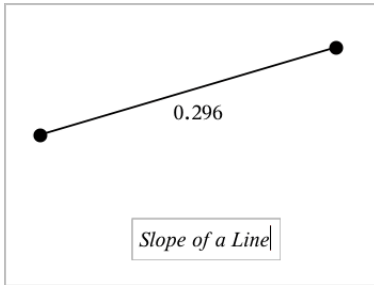
For information on moving, resizing, and deleting a background image, see *Working with Images in the Software*.

Adding Text to the Graphs or Geometry Work Area

1. From the **Actions** menu, select **Text**.

    The Text tool appears in the work area.

2. Click the location for the text.

3. Type the text in the box that appears, and then press **Enter**.

4. To close the Text tool, press **ESC**.

5. To edit the text, double-click it.

Deleting a Relation and its Graph

1. Select the relation by clicking its graph.

2. Press **Backspace** or **DEL**.

    The graph is removed from both the work area and the graph history.

Customizing the 3D Viewing Environment
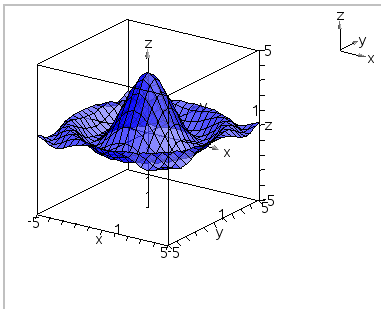
Setting the Background Color

▸   Display the context menu for the work area, and then click **Background Color**.
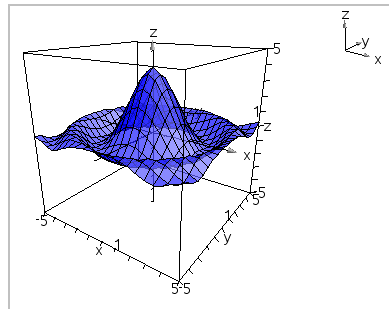
Showing or Hiding Specific View Elements

▸   From the **View** menu, click the item to show or hide. You can choose items such as the 3D box, axes, box end values, and legend.

Changing the 3D Projection

▸ From the **View** menu, click **Orthographic Projection** or **Perspective View**.

Orthographic Projection (default)	*Perspective View*

Setting the Visual Attributes of the Box and Axes

1. Display the context menu for the box, and then click **Attributes**. You can set the following attributes.

   - Show or hide tic labels
   - Show or hide end values
   - Show or hide arrows on axes
   - Show 3D or 2D arrow heads

2. Set the attributes as you like, and then press **Enter** to accept the changes.

Shrinking or Magnifying the 3D View

▸ From the **Range/Zoom** menu, click **Shrink Box** or **Magnify Box**.

Changing the 3D Aspect Ratio

1. From the **Range/Zoom** menu, click **Aspect Ratio**.

2. Enter values for the x, y, and z axes. The default value for each axis is **1**.

Changing the Range Settings

▸ On the **Range/Zoom** menu, click **Range Settings**. You can set the following parameters.

   - XMin (default=**-5**)
     XMax (default=**5**)
     XScale (default=**Auto**) You can enter a numeric value.

- YMin (default=**-5**)

  YMax (default=**5**)

  YScale (default=**Auto**) You can enter a numeric value.

- ZMin (default=**-5**)

  ZMax (default=**5**)

  ZScale (default=**Auto**) You can enter a numeric value.

- eye $\theta°$ (default=**35**)

  eye $\phi°$ (default=**160**)

  eye distance (default=**11**)

Creating Shapes Using Gestures (Math Draw)

The Math Draw tool lets you use touchscreen or mouse gestures to create points, lines, circles, and other shapes.

Math Draw is available in:

- Geometry view without the analytic window displayed.

- Graphing view when the x scale and y scale are identical. This avoids non-circular ellipses and non-square rectangles appearing as circles and squares.

Math Draw is not available in the 3D Graphing view or in the Geometry view with the analytic window displayed.

Activating Math Draw

1. If using the Geometry view with the analytic window visible, use the **View** menu to hide the window.

2. On the **Actions** menu, select **Math Draw**.

   The Math Draw icon ⬡ appears. You can begin using the tool.

Canceling Math Draw

▸ When you have finished using the Math Draw tool, press **Esc**.

   The tool also closes if you select a different tool or change views.

Creating Points

To create a labeled point, tap or click in an open area.

- If the point is close to an existing line, segment, ray, geometric conic (including circles), or polygon, the point snaps to that object. You can also place a point on the intersection of any two of those types of objects.

- If the point is close to a visible grid location in a Graphs view or the analytic window of a Geometry view, it snaps to the grid.

Drawing Lines and Segments

To create a line or segment, touch or click the initial position, and then drag to the end position.

- If the drawn line passes near an existing point, the line snaps to the point.

- If the drawn line starts close to an existing point and ends next to another existing point , it becomes a segment defined by those points.

- If the drawn line is nearly parallel or perpendicular to an existing line, segment, or side of a polygon, it aligns to that object.

  Note: The default tolerance for detecting parallel/perpendicular lines is 12.5 degrees. It is defined in a variable named **ti_gg_fd.angle_tol**. You can change the tolerance in the current problem by setting this variable to a value in the range 0 through 45 (0=no parallel/perpendicular detection).

Drawing Circles and Ellipses

To create a circle or ellipse, use the touchscreen or mouse to draw the approximate shape.

- If the drawn shape is sufficiently circular, a circle is created.

- If the shape is elongated, an ellipse is created.

- If the virtual center of the drawn shape is near an existing point, the circle or ellipse is centered on that point.

Drawing Triangles

To create a triangle, draw a triangle-like shape.

- If a drawn vertex is close to an existing point, the vertex snaps to the point.

Drawing Rectangles and Squares

To create a rectangle or square, use the touchscreen or mouse to draw the perimeter.

- If the drawn shape is nearly square, a square is created.

- If the drawn shape is elongated, a rectangle is created.
- If the center of a square is close to an existing point, the square snaps to that point.

Drawing Polygons

To create a polygon, tap or click a succession of existing points, ending on the first point you tapped.

Using Math Draw to Create Equations

In the Graphs view, Math Draw attempts to recognize certain gestures as equations, particularly analytic parabolas of the form **y=**.

Note: The default step value for quantization of the parabola coefficients is 1/32. The denominator of this fraction is defined in a variable named **ti_gg_fd.par_quant**. You can change the step value in the current problem by setting this variable to a value greater or equal to 2. A value of 2, for example, produces a step value of 0.5.

Using Math Draw to Measure an Angle

To measure the angle between two existing lines, use the touchscreen or mouse to draw a circle arc from one of the lines to the other.

- If the intersection point between the two lines does not exist, it is created and labeled.
- The angle is not a directed angle.

Using Math Draw to Find a Mid-point

To create a point midway between two points, tap or click point 1, point 2, and then point 1 again.

Using Math Draw to Erase

To erase objects, use the touchscreen or mouse to drag left and right, similar to the motion of erasing a whiteboard.

- The erasure area is the bounding rectangle of the erasure gesture.
- All point objects and their dependents inside the erasure area are removed.

What You Must Know

Changing the Graphs and Geometry Settings

1. From the **Settings** menu in the Documents Toolbox, select **Settings**.

2. Select the settings that you want to use.

   - **Display Digits.** Sets the display format for numbers as Floating or Fixed decimal.

   - **Graphing Angle.** Sets the angle unit for all Graphs and 3D Graphing applications in the current docuument. The default setting is Radian. Set this to Auto if you want graphing angles to follow the Angle setting in the main **File > Settings** menu. An angle mode indicator shows the resulting mode in Graphs and 3D Graphing applications.

   - **Geometry Angle.** Sets the angle unit for all Geometry applications in the current document. The default setting is Degree. Set this to Auto if you want geometry angles to follow the Angle setting in the main **File > Settings** menu. An angle mode indicator shows the resulting mode in Geometry applications.

   - **Automatically hide plot labels.** In the Graphs application, hides the label that normally appears next to a graphed relation.

   - **Show axis end values.** Applies only in the Graphs application.

   - **Show tool tips for function manipulation.** Applies only in the Graphs application.

   - **Automatically find points of interest.** In the Graphs application, shows zeros, minima, and maxima while tracing function graphs.

   - **Force Geometric Triangle Angles to Integers.** Restricts the angles of a triangle to integer values as you create or edit the triangle. This setting applies only in the Geometry View with the Geometry Angle unit set to Degree or Gradian. It does not apply to analytic triangles in Graphing View or to analytic triangles in the Analytic Window of the Geometry View. This setting does not affect existing angles, and it does not apply when constructing a triangle based on previously inserted points. By default, this setting is deselected.

   - **Automatically Label Points.** Applies labels (A, B, ..., Z, A_1, B_1, and so on) to points, lines, and vertices of geometric shapes as you draw

them. The labeling sequence starts at A for each page in a document. By default, this setting is deselected.

Note: If you create a new object that uses existing unlabeled points, those point are not automatically labeled in the completed object.

- Click **Restore** to restore all settings to their factory defaults.

- Click **Make Default** to apply the current settings to the open document and save them as the default for new Graphs and Geometry documents.

Using Context Menus

Context menus provide quick access to commonly used commands and tools that apply to a specific object. For example, you can use a context menu to change an object's line color or to group a set of selected objects.

▶ Display the context menu for an object in one of the following ways.

- Windows®: Right-click the object.

- Mac®: Hold ⌘ and click the object.

- Handheld: Move the pointer to the object, and then press ctrl menu.

Finding Hidden Objects in the Graphs or Geometry Application

You can hide and show individual graphs, geometric objects, text, labels, measurements, and axis end-values.

To temporarily view hidden graphs or objects or to restore them as shown objects:

1. From the **Actions** menu, select **Hide/Show**.

   The Hide/Show tool 👁 appears in the work area, and all hidden objects become visible in dimmed colors.

2. Click a graph or object to toggle its Hide/Show state.

3. To apply the changes and close the Hide/Show tool, press **ESC**.

Inserting a Background Image

You can insert an image as a background for a Graphs or Geometry page. The file format of the image can be .bmp, .jpg, or .png.

1. From the **Insert** menu, click **Image**.

2. Navigate to the image you want to insert, select it, and then click **Open**.
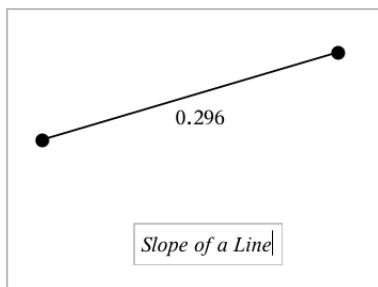
For information on moving, resizing, and deleting a background image, see _Working with Images in the Software_.

Adding Text to the Graphs or Geometry Work Area

1. From the **Actions** menu, select **Text**.

    The Text tool ![Abl] appears in the work area.

2. Click the location for the text.

3. Type the text in the box that appears, and then press **Enter**.

4. To close the Text tool, press **ESC**.

5. To edit the text, double-click it.

Deleting a Relation and its Graph

1. Select the relation by clicking its graph.

2. Press **Backspace** or **DEL**.

    The graph is removed from both the work area and the graph history.

Measuring Objects

Measurement values update automatically as you manipulate the measured object.

Note: Measurements of objects created in the Graphs application are displayed in generic units named u. Measurements of objects created in the Geometry application are displayed in centimeters (cm).
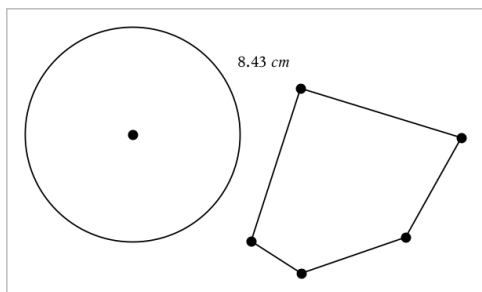
Measuring Length of a Segment, Circle Arc, or Vector

1. From the **Measurement** menu, select **Length**. (In the Graphs application, click **Geometry > Measurement > Length**.)

2. Click the object to display its length.
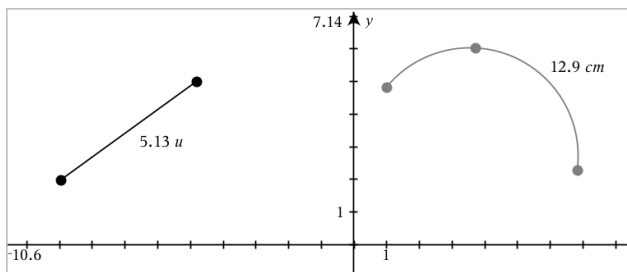
Measuring Distance Between Two Points, a Point and a Line, or a Point and a Circle

1. From the **Measurement** menu, select **Length**. (In the Graphs application, click **Geometry > Measurement > Length**.)

2. Click the first point.

3. Click the second point or a point on the line or circle.

In this example, length is measured from the center of the circle to the upper left vertex of the polygon.

Measuring Circumference of a Circle or Ellipse or the Perimeter of a Polygon, Rectangle, or Triangle

1. From the **Measurement** menu, select **Length**. (In the Graphs application, click **Geometry > Measurement > Length**.)

2. Click the object to display its circumference or perimeter.
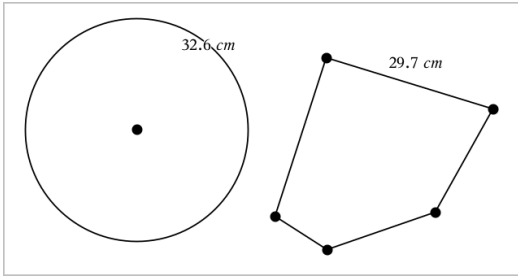
Measuring a Side of a Triangle, Rectangle, or Polygon

1. From the **Measurement** menu, select **Length**. (In the Graphs application, click **Geometry > Measurement > Length**.)

2. Click two points on the object that form the side you want to measure.

   Note: You must click *two points* to measure a side. Clicking the side measures the entire length of the object's perimeter.
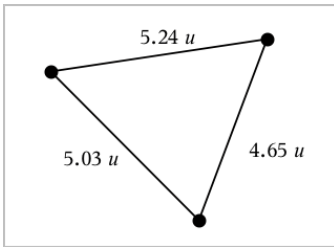
## Measuring Area of a Circle, Ellipse, Polygon, Rectangle, or Triangle

Note: You cannot measure the area of a polygon constructed using the Segment tool.

1. From the **Measurement** menu, select **Area**. (In the Graphs application, click **Geometry > Measurement > Area**.)

2. Click the object to display its area.
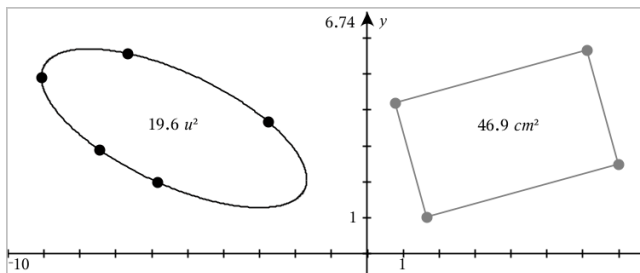
Measuring Slope of a Line, Ray, Segment, or Vector

1. From the **Measurement** menu, select **Slope**. (In the Graphs application, click **Geometry > Measurement > Slope**.)

2. Click the object to display its slope.
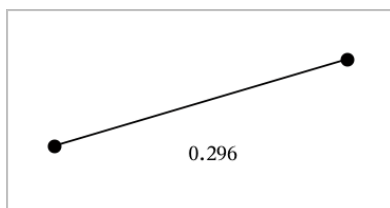
The value is updated automatically when you manipulate the object.

Measuring Angles

Measured angles in the Geometry application range from 0° to 180°. Measured angles in the Graphs application range from 0 radians to π radians. To change the angle unit, use the **Settings** menu.

1. From the **Measurement** menu, select **Angle**. (In the Graphs application, click **Geometry > Measurement > Angle**.)

2. Click three locations or points to define the angle. The second click defines the vertex.
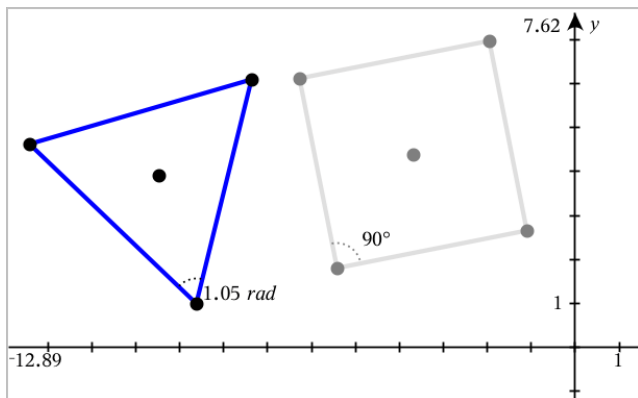
Measuring Angles using the Directed Angle Tool

1. From the **Measurement** menu, select **Directed Angle**. (In the Graphs application, click **Geometry > Measurement > Directed Angle**.)

2. Click three locations or existing points to define the angle. The second click defines the vertex.
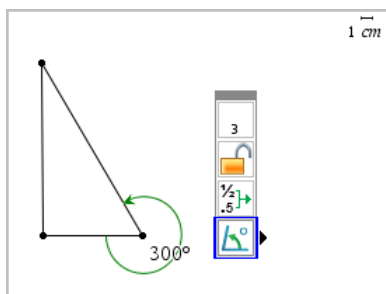
3. To reverse the measurement orientation,

   a) On the **Actions** menu, select **Attributes**.

   b) Click the angle text. For example, click **300°**.

   c) Select the orientation attribute, and use the right or left arrow key to change it.

   d) Press **Esc** to close the Attributes tool.
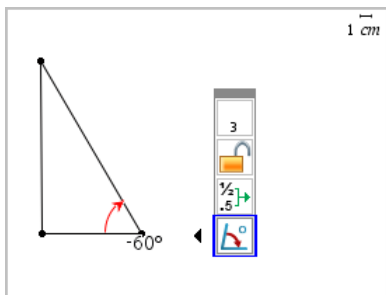
Moving a Measured Value

▶ Drag the measurement to the desired location.

Note: If you move a measurement too far from its object, it stops following the object. However, its value continues to be updated as you manipulate the object.

Editing a Measured Length

You can set the length of a side of a Triangle, Rectangle, or Polygon by editing its measured value.

▶ Double-click the measurement, and then enter the new value.

Storing a Measured Value as a Variable

Use this method to create a variable and assign a measured value to it.

1. Display the item's context menu, and select **Store**.

2. Type a variable name for the stored measurement.

Linking a Measured Length to an Existing Variable

Use this method to assign a measured length value to an existing variable.

1. Display the measurement's context menu, and select **Variables > Link to**.

   The menu shows the list of currently defined variables.

2. Click the name of the variable you want to link to.

Deleting a Measurement

▶ Display the measurement's context menu, and select **Delete**.

Locking or Unlocking a Measurement

1. Display the measurement's context menu, and select **Attributes**.

2. Use the up/down arrow keys to highlight the Lock attribute.

3. Use the left/right arrow keys to close or open the lock.

    As long as the value remains locked, manipulations are not allowed that would require the measurement to change.

Calculating a Bounded Area

Note: To avoid unexpected results when using this feature, make sure the document setting for "Real or Complex Format" is set to **Real**.

When you calculate the area between curves, each curve must be:

*   A function with respect to x.

    - or -

*   An equation in the form y=, including y= equations defined through a text box or a conic equation template.

Defining and Shading the Area

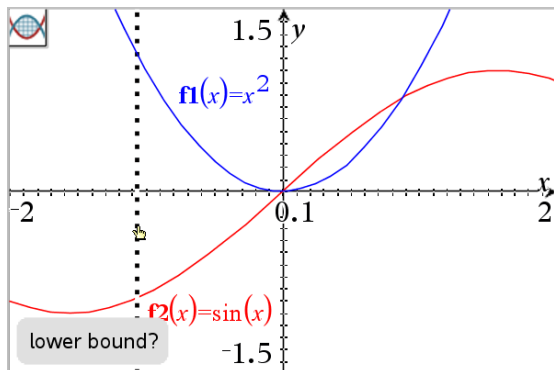1. From the **Analyze Graph** menu, select **Bounded Area**.

    If exactly two appropriate curves are available, they are selected automatically, and you can skip to step 3. Otherwise, you are prompted to select two curves.

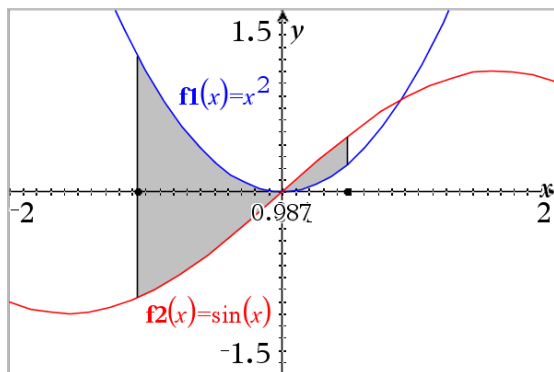2. Click two curves to select them.
    - or -
    Click one curve and the x axis.

    You are prompted to set the lower and upper bounds.

3. Click two points to define the bounds. Optionally, you can type numeric values.

The area becomes shaded, and the area value is displayed. The value is always non-negative, regardless of the interval direction.


Working with Shaded Areas

As you change the bounds or redefine the curves, the shading and the area value are updated .

- To change the lower or upper bound, drag it or type new coordinates for it. You cannot move a bound that resides on an intersection. However, the point moves automatically as you edit or manipulate the curves.

- To redefine a curve, either manipulate it by dragging or edit its expression in the entry line.

If an endpoint resided originally on an intersection, and the redefined functions no longer intersect, the shading and area value disappear. If you redefine the function(s) so that there is an intersection point, the shading and area value reappear.

- To delete or hide the shaded area, or to change its color and other attributes, display its context menu.

  - Windows®: Right-click the shaded area.

  - Mac®: Hold ⌘ and click the shaded area.

  - Handheld: Move the pointer to the shaded area and press [ctrl] [🖱].

Basic Operations in Data & Statistics

The Data & Statistics application lets you explore and visualize data and graph inferential statistics. The Lists & Spreadsheet application can work in conjunction with the Data & Statistics application. The Lists & Spreadsheet Summary Plot and Quick Graph tools automatically add a Data & Statistics application to show plots. A list that you create in a problem (using the Lists & Spreadsheet or Calculator applications) can be accessed as a variable in any TI-Nspire™ application in that problem.

Changing Data & Statistics Settings

1. From the **Settings** menu, select **Settings**.

2. Select the settings that you want to use.

   - **Display Digits.** Lets you select the display format for numeric labels in the current document. Select **Auto** to automatically follow the setting in the Document Settings dialog box.

   - **Diagnostics.** Displays the value of the r^2 or R^2 statistic (when available) under certain regression equations.

     - r^2 is displayed for Linear (mx+b), Linear (a+bx), Power, Exponential, and Logarithmic regressions.

     - R^2 is displayed for Quadratic, Cubic, and Quartic regressions.

Using the Default Caseplot

The Data & Statistics application plots numeric and string (categorical) data from variables. When you add a Data & Statistics application to a problem that includes lists, a default caseplot displays on the work area.

The caseplot is like having a stack of cards with information on them and scattering the cards randomly on a table. You can click a dot to see the information on that "card." You can drag a dot to "group" the "cards" by the caption variable.
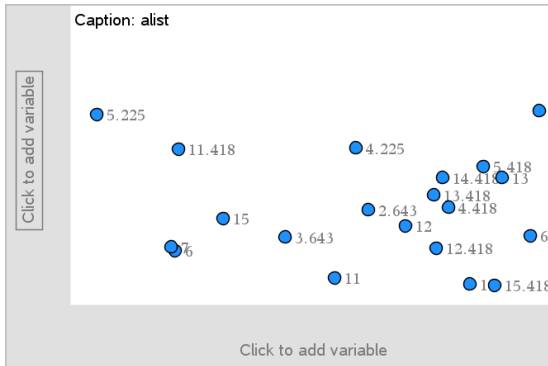
▶ Click the variable name displayed after **Caption** to use the caseplot.

- Choose <None> to remove the default caseplot.

- Choose the name of a variable to have it replace the current caseplot variable.

- Hover over any data point to see the summary information.

- Drag any data point toward an axis to see how the points group.

- Activate the Graph Trace tool and press ◄ or ► to move across points.

When you add a variable to either axis, the plot for that variable replaces the default caseplot. The default caseplot redisplays if you remove the plotted variable from each axis.

Using the Context Menu

The context menu provides access to the tools most commonly used with the selected object. The context menu displays different options depending on the active object and the task you are performing.

▶ To open the context menu for an object.

Windows®: Right-click the object.

Mac®: Hold ⌘ and click the object.

Handheld: Point to the object and press [ctrl] [menu].

The context menu includes the **Color** option. You can use the Color option to change the data to the color of your choice.

Other options that are appropriate for various plots also appear on the context menu.

## Selecting Data and Displaying Summary Information

When you hover over part of a plot, the Data & Statistics application displays summary information for the data it represents.

1. Hover at an area of interest in a plot to display data values or summary information. For example, you can hover over the center of a box plot to display the median summary data.

2. Click once to select a representation of data in a plot.

    Data points are shown with a bold outline to indicate selection. You can click a point a second time to deselect it, or click additional points to add to the selection.

Plotting Variables

To plot variables, start with a problem that includes a Data & Statistics application and lists created in the Lists & Spreadsheet application or the Calculator application.

1. Click the Add Variable region near the center of an axis.

    If no variable is plotted on the axis, the tooltip **Click or Enter to add variable** displays.

2. Click the tooltip **Click or Enter to add variable**.

   A list displays the names of available variables.
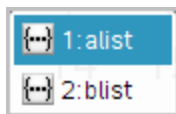
3. Click the name of the variable to plot.

    Note: By convention, the independent variable is shown on the x-axis.

    The default plot for one variable is a dot chart. The data points in the default caseplot reposition to represent the elements of the selected variable in a dot chart.

4. (Optional) Click the Add Variable region near the center of the remaining axis to plot a second variable.

    The default plot for two variables is a scatter plot. The data points shift to represent the elements of both variables as a scatter plot.

5. (Optional) Repeat Steps 1-3 to choose additional variables to plot on the vertical axis.
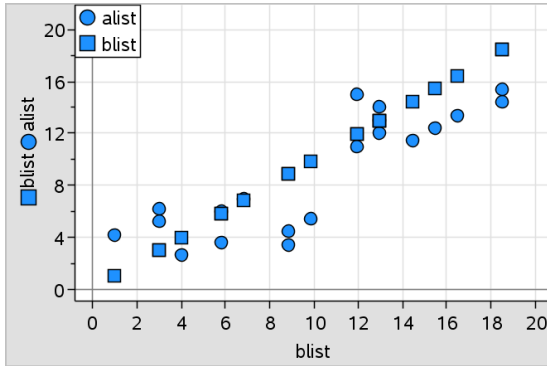
The name of each variable that you add is appended to the label on the axis. The default data point shape changes to help you distinguish data, and a legend is displayed to identify the shapes.

6. Change, analyze, or explore the plotted data.

    -   Remove or change the variable on an axis by clicking the Add Variable region again.

    -   View the plotted data in another supported plot type by selecting a tool from the **Plot Types** menu.

    -   Choose the Graph Trace tool on the **Analyze** menu and press ◄ or ► to move across the data points in the plot.

    -   The lists that you plot as variables can include incomplete or missing cases. (A case is the data contained in a row of cells in the Lists & Spreadsheet application.) The Lists & Spreadsheet application displays a void as an underscore ("_"), and Data & Statistics plots no data point for a void cell.

Manipulating Plotted Data

You can manipulate data points on the Data & Statistics work area to explore their effects. For example, you could explore how a specific group of values affects the median.

You can move a data point only in directions allowed by its definition. If a list is defined with a formula in Lists & Spreadsheet, the points in Data & Statistics may not move because of the formula's restrictions. For example, you can manipulate a plot that represents the result of y=x, but you can only move along a line.

You cannot move points that represent data in a locked variable or data that represents a categorical value.

1. On the Data & Statistics work area, click a representation of data–such as a histogram bin or a whisker of a box plot–that is not locked or restricted by a formula.
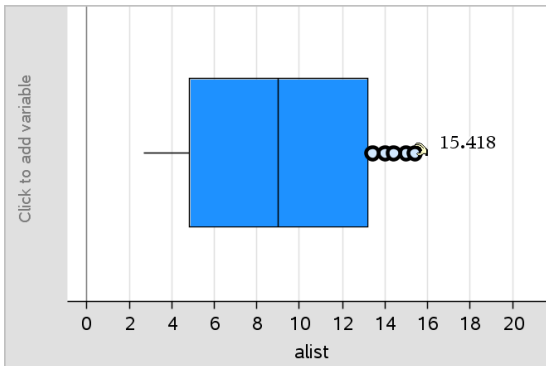
The pointer changes to an open hand to show that the data can be moved.

2. Drag the selection to explore how different values of the point affect the plot.

    Handheld: Press $\boxed{\text{ctrl}}$ $\boxed{\text{⬚}}$ to grab, and then swipe or use the arrow keys to drag.

    As you drag, the changing value displays on the work area.

Inserting New Scripts

To insert a new script application and script, follow these steps.

1. Open the document where you want to insert the script. It can be a new or existing document.

2. Click **Insert > Script Editor > Insert Script**.

   A script application is inserted, and the Script Title dialog box opens.

   Note: The TI-Nspire™ CX Student Software and the TI-Nspire™ CX CAS Student Software open automatically within the Documents Workspace.

3. Type a script title. (The maximum number of characters is 32.)

4. Click **OK**.

   The Script Editor window opens showing a blank script.

5. Type your text in the script lines.

   Note: Some nonstandard UTF-8 wide characters may not be displayed correctly. For these characters, it is highly recommended that you use the string.uchar function.

6. When the script is complete, click **Set Script** to execute it.

   • In a TI-Nspire™ document, the script application is inserted in a new page. When the page containing the script application is active, the Documents Toolbox is empty.

   • In a PublishView™ document, a frame containing the script application is added to the active page. You can move or size this frame just as you would any other PublishView™ object, and you can add other PublishView™ objects to the page.

7. To view the script application, click **Focus Script**.

Zooming Text in the Script Editor

You can adjust the size of the text in your script and in any of the tool panels.

Menu Method

1. Click in the area of the text to zoom.

2. On the **View** menu, select **Zoom**, and then select **Zoom In**, **Zoom Out**, or **Restore**.

   Note that the menu also displays keyboard shortcuts for the Zoom commands.

Mouse Method

1. Postion the mouse pointer over the area to zoom.

2. Hold **Ctrl**, and roll the mouse wheel forward or back.