SigGenRZ Manual

.

SigGenRZ Manual

Copyright

Licenses and Trademarks

Printed in U. S. A.

Updated: 3/22/2010 4:03 PM

Contents

Organization of the Manual

The *SigGenRZ User's Guide* is divided into two parts:

- Part 1 SigGenRZ Fundamentals
- Part 2 Illustrative Examples

SigGenRZ Fundamentals

Part 1, SigGenRZ Fundamentals, presents all the basic SigGenRZ concepts necessary to build a signal. Part 1 guides you through the process of building a SigGenRZ signal and also serves as a general reference tool. Part 1 provides:

- An introduction to SigGenRZ capabilities, tools, and concepts.
- A detailed explanation of the signal design and preview process.

General Information

An introduction to the purpose and uses of SigGenRZ, SigGenRZ installation, and basic SigGenRZ concepts is presented in:

- *Chapter 1* Introduction

The SigGenRZ interface is described in:

- *Chapter 2* Learning the Basics

The Signal Design Process

The steps required to design and preview a signal are presented below, along with their associated chapter.

Step	**Chapter**
1. Define signal parameters	*Chapter 3* Preparing to Build a Signal – The First Step
2. Define segment parameters	*Chapter 4* Signal Segments – The Building Blocks of a Signal
3. Define components	*Chapter 5* Components – The Building Blocks of Segments
4. View, modify, and/or preview completed segments	*Chapter 6* Working with Segments
5. View, modify, and/or preview completed signals	*Chapter 7* Working with the Complete Signal

Illustrative Examples

In Part 2, Illustrative Examples, various SigGenRZ signals are designed in a step-by-step manner. These sample signals illustrate many powerful SigGenRZ features. The illustrative examples were designed to:

- Provide general examples utilizing various SigGenRZ features.

Part 1 SigGenRZ Fundamentals

~

Chapter 1 Introduction

Welcome to SigGenRZ, the most complete Windows-based signal design package available today. SigGenRZ is designed to work with the RZ6 Multi I/O processor and BioSigRZ for ABR and OAE studies and includes standard stimulus files which can be selected and modified through BioSigRZ. Most researchers using BioSigRZ for animal screening or similar tasks will NOT find it necessary to work directly in SigGenRZ.

Scientists who wish to work with stimuli beyond the scope of the standard stimuli can use SigGenRZ to design, specify, and preview complex auditory stimuli in a fully interactive, high-level signal development environment.

Important!: All BioSigRZ users are encouraged to read the BioSigRZ manual **before** investigating the more comprehensive SigGenRZ capabilities covered in this manual.

What Is SigGenRZ?

Signal generation is an integral part of virtually all tests and experiments involving acoustic stimuli. The SigGenRZ stimulus design application enables scientists to quickly design, specify, and preview the complex auditory stimuli used in their experiments. It offers a fully interactive, high-level signal development environment that can be used to generate sophisticated signals to be used with TDT’s BioSigRZ signal acquisition and processing application.

SigGenRZ Capabilities

SigGenRZ is a complete stimulus design environment with many desirable features. Combined with TDT’s BioSigRZ, it becomes a comprehensive design platform where stimuli for all trials of an experiment are specified. SigGenRZ provides control over the type of stimulus, frequency content, gating function, level, starting time, overall duration, and calibration. SigGenRZ even provides a means for controlling the systematic variation of these parameters from one stimulus interval to another.

Calibration

SigGenRZ allows you to specify all signal levels in terms of a calibrated reference level. This reference level can be specified as 0 dB SPL or some other reference.

Signal Design

SigGenRZ allows you to design a wide variety of signals. Signal components can be specified in the time or frequency domains. You can choose between tones, clicks, noise, bands of noise, harmonics, FM sweeps, and waveforms from disk. SigGenRZ's features enable you to build complex tones and amplitude modulated signals. You can also specify exact signal onset and offset times.

Playback Preview

SigGenRZ allows you to simulate stimulus presentation for an entire experiment by playing through all of the signal intervals. The values of variables are displayed and are reflected in the real-time generation of the signals.

Displays

SigGenRZ's two main displays are the Signal window and the Segment editor. The Signal window displays the overall stimulus signal, showing the placement of segment waveforms along a temporal axis. The Segment editor displays design parameters, the time-domain segment waveform generated by these design parameters, and the magnitude spectrum of the segment waveform. You can *zoom-in* on a specific area of the signal. Values associated with the displayed signal are viewed in an information window. You may place this window anywhere within the screen display.

Hardware Support

SigGenRZ supports TDT's RZ6 Multi I/O processor. The program uses a software emulation of the System II array processor to design signals. The generated waveform is then loaded to a memory buffer on an RPvdsEx circuit. SigGenRZ uses this circuit to control stimulus presentation, sample period, the maximum number of channels that are played out, and also the triggering properties of the system.

SigGenRZ comes with several preconfigured RCO circuit files designed for use with BioSigRZ. See *Device Select* on page 14 for more information.

The Power of SigGenRZ

SigGenRZ provides a simple, yet powerful, means for designing and previewing a wide variety of signals. Signal design, however, is often just the first step in a larger process. The real power of SigGenRZ lies in its ability to generate files that can be used to in BioSigRZ software to present stimuli across multiple intervals or SigGenRZ Indices (SGIs) during an experiment.

What is the SigGen index (SGI)?

In most cases, a signal is built to be presented as an auditory stimulus in an experiment or test. The stimulus may be presented over multiple intervals with signal parameters varied as a function of the interval.

The *SigGen index* (SGI) is simply another name for the interval number. During stimulus presentation, signal parameters are varied as a function of the SigGen index (SGI).

Using SigGenRZ Signals

The process of designing and using SigGenRZ signals is as follows:

1. Design the signal with SigGenRZ.
2. Save the signal to a file.
3. Use the file with BioSigRZ or your own custom signal processing application.
4. Process output.

Signal File Formats

SigGenRZ signals may be saved in the following formats:

- SigGenRZ format
- Binary file format

SigGenRZ File Format

Files saved in the SigGenRZ format contain signal parameter information and the variable information necessary to control the presentation of stimulus signals across multiple intervals, or SigGenRZ Indices (SGIs).

Binary File Format

Files saved in binary format may be easily used with many commercial application software packages or user’s programs. These files are saved in 16-bit integer binary format and consist of the waveform only.

Before You Begin

With a bit of preparation, signal generation with SigGenRZ is quick and easy.

What You Need

You should be familiar with your System 3 hardware and understand the parameters necessary for specification of a signal in the frequency and time domains.

Installing the Software

Requirements

To run SigGenRZ, you must have the following:

- Computer running 32-bit Microsoft Windows XP, Windows Vista, or Windows 7
- TDT Drivers software (version 72 or greater)
- A monitor with at least VGA resolution graphics. Super VGA (1024 x 768) resolution graphics highly recommended
- RZ6 Multi I/O processor (required for playing signals in SigGenRZ)
- Optibit Interface Card in your PC to communicate with RZ6

Installations

To install SigGenRZ:

1. Make sure your TDT hardware is installed and functioning properly. Refer to the *System 3 Installation Guide*.
2. Insert the software CD in to your CD-ROM drive.
3. Click **System 3 Software**, and click **SigGenRZ**.
4. Follow the installation wizard.

~

Chapter 2 Learning the Basics

Getting Started

Starting SigGenRZ

To start SigGenRZ:

➢ Click the **Start** button on the Windows taskbar, point to **Programs**, point to **TDT Sys 3**, and click **SigGenRZ** on the start menu.

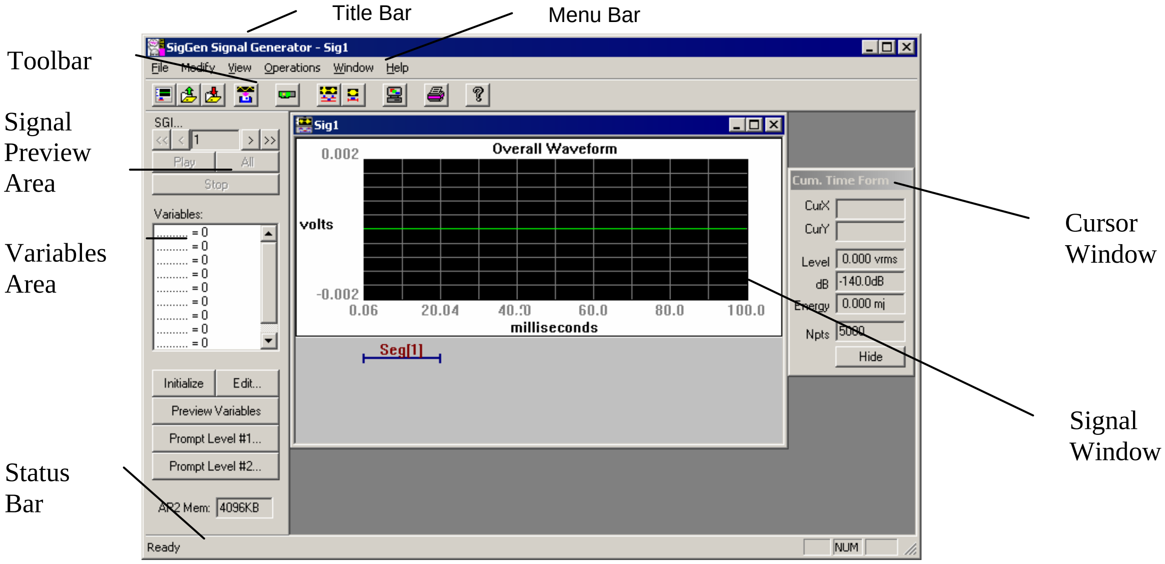
or

➢ Double-click the **SigGenRZ** icon.

The SigGenRZ main window opens. You are now ready to begin designing signals with SigGenRZ.

Getting to Know the Main Window

Upon starting SigGenRZ, you will be presented with the SigGenRZ main window. It is from this window that you will be able to view the complete signal once you have defined all of its segments. The SigGenRZ main window also provides functions that allow you to preview the signal through multiple intervals, or SigGenRZ Indices.



The window contains the following sections:

Title Bar: Displays "SigGenRZ Signal Generator" and the name of the current SigGenRZ file. If the signal has not been saved to a SigGenRZ file, a default name, Sig1 for example, will be displayed.

Menu Bar: Contains a list of menus used for building, manipulating, or saving your signal.

Toolbar: Provides easy-to-use icons for the most commonly used SigGenRZ menu commands.

Signal Window: Displays the signal(s) as a time-domain waveform. More than one signal can be displayed at the same time.

Signal Preview Area: Allows you to preview the stimulus through one specific interval or all intervals.

Variables Area: Variables allow you to specify the manner in which signal parameters will be varied from one interval to another. This area displays the variable names and their current values corresponding to the current SGI.

Cursor Window: Provides a readout of parameter values pertaining to any selected segment or signal.

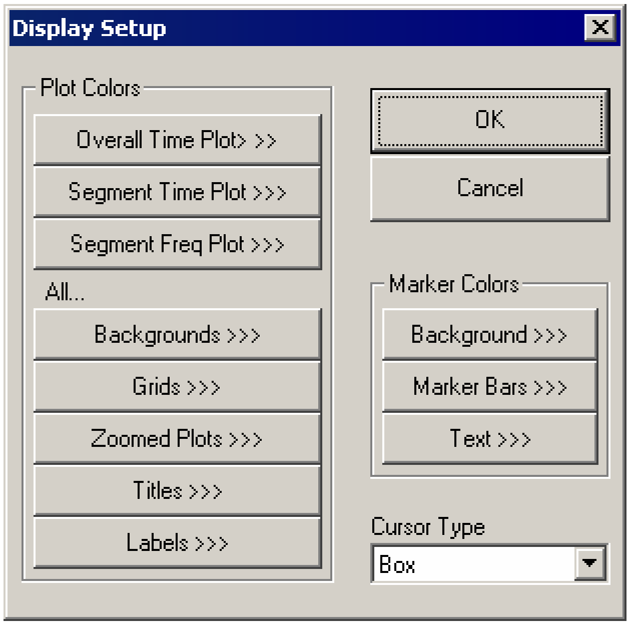
Status Bar: Provides you with information concerning the selected command.

Customizing Your Display

Display settings that can be altered in SigGenRZ can be customized from the Display Setup dialog box.

To access the Display Setup dialog box:

- Choose **Display** from the **Modify** menu.



Cursor Type: Box

From the Display Setup dialog box you may configure the following:

- Plot Colors
- Marker Colors
- Cursor Type

Using the Menus and Toolbar

SigGenRZ provides you with a full set of menu commands and a toolbar buttons. Using these menus, commands, and toolbar buttons, you can create, open, and save signals; build new signals or edit existing signals; print signals; and make choices about your screen display.

The Menus

The File Menu

The File Menu provides standard methods for creating, saving, opening, closing, and printing files.

New: Opens a new Signal Window.

Open: Opens and displays an existing signal file.

Close: Closes the current signal window.

Save and **Save As:** These menu options save signals in SigGenRZ's own format. Files saved in this format use the default extension *.sig*.

Save Wave: Saves a waveform as a 16-bit integer binary file. Files saved in this format use the default extension *.16*.

Target Device: Opens the Setup RP Device dialog box, from which you can configure TDT’s Real-time processors.

Print: Prints the current waveform.

Print Preview: Allows you to preview the waveform as it will be printed.

Print Setup: Allows you to select a printer, define the paper size, and choose a page orientation.

The Modify Menu

The Modify Menu provides commands that allow you to specify and edit signal and segment parameters, configure peripheral devices, and modify SigGenRZ display.

Signal: Opens the Signal Parameters dialog box, from which you can define variables and specify parameters that will affect the entire signal.

Segment: Opens the Edit Signal Segments dialog box, from which you can specify the segment parameters and design the components of the segment.

Display: Opens the Display Setup dialog box, from which you can specify colors for different display areas and select cursor types.

The View Menu

From the View Menu, you can choose to display or hide the Toolbar, Status Bar and Cursor Box. Normalization Curve is not supported.

The Operations Menu

The Operations menu provides commands that calculate the signal, set the value of signal variables, and control signal preview.

Calculate: Causes the signal displayed in the current signal window to be recalculated based on the current SGI and parameter settings. This operation will update the waveform displayed in the Signal Window.

Initialize Variables: Causes the SigGen index (SGI) to be reset to 1, restoring all signal variables to their starting values. Recalculates and updates the waveform displayed in the Signal Window.

Prompt Level #1: When a Prompt Level #1 variable is defined; this operation will display the Input Request dialog box and prompt you for the value of the variable. This box will be repeatedly displayed if there are more than one Prompt Level #1 variables defined. This box will not be displayed if there is no Prompt Level #1 variable defined.

Prompt Level #2: When a Prompt Level #2 variable is defined; this operation will display the Input Request dialog box and prompt you for the value of the variable. This box will be repeatedly displayed if there are more than one Prompt Level #2 variables defined. This box will not be displayed if there is no Prompt Level #2 variable defined.

Note: When there is more than one Prompt variable defined, values entered to Input Request box will be assigned to the variables according to their order in the variable table.

Increment SGI: Increments the SigGen index by 1. This operation will recalculate and update the current waveform.

Advance SGI by 10: Increments the SigGen index by 10. This operation will recalculate and update the current waveform.

Decrement SGI: Decrements the SigGen index by 1. This operation will recalculate and update the current waveform.

Decrease SGI by 10: Decrements the SigGen index by 10. This operation will recalculate and update the current waveform.

Play Waveform: Plays the signal at the current SigGen index.

Play All: Plays the signal for all SigGenRZ Indices beginning at SGI = 1. If there is no boundary control for SGI through a SigGenRZ variable, signal will play until it is stopped manually.

Stop Playing: Halts signal play.

The Window Menu

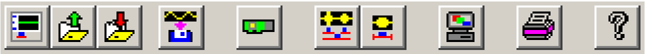
The Window menu provides standard Windows commands that allow you to easily manipulate multiple signal windows.

The Help Menu

From the Help Menu, you can display the About SigGenRZ message box or access the SigGenRZ manual.

The Toolbar

You can find quick point-and-click access to the most common commands on the SigGenRZ Toolbar.



Click		To
	New	Open a new Signal Window
	Open	Open an existing SigGenRZ signal file
	Save As	Save a SigGenRZ signal file
	Save Wave	Save to a 16-bit binary file
	Calculate	Calculate the current signal
	Modify Signal	Edit signal parameters
	Modify Segment	Edit segment parameters
	Display Setup	Customize display colors and cursor types

Print Preview	Preview the waveform as it will be printed
SigGenRZ Help	Access SigGenRZ manual

The Signal Window

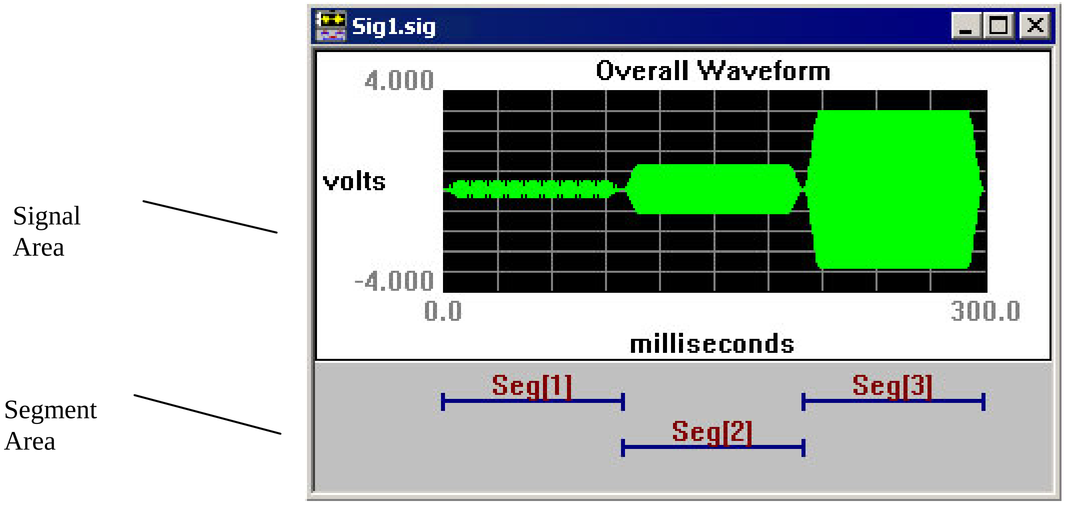
Signals are displayed in child windows known as signal windows.

To open a new signal window:

- Choose **New** from the **File** menu.

  or

- Click on the **New Signal** button on the toolbar.



Signal Area

Segment Area

Each Signal Window consists of:

- **Signal Area** Contains the time-domain waveform display.
- **Segments Area** Displays graphic representations (segment bars) of the durations and temporal positions of each segment in the signal. At any time scale, up to four overlapping segment bars can be displayed.

SigGenRZ builds a signal from mini-signals called segments. When you are designing signals in SigGenRZ, you are actually designing at least one segment, specifying how these segments are related to each other in time, and specifying how the segments will affect the entire signal.

Because each signal must contain at least one segment, "new" signals, by default, contain one segment. The default segment is defined as follows:

- Segment onset is 0 milliseconds.
- Segment length is 20 milliseconds.
- No components defined in the segment.

Working with Signal Files

Opening a SigGenRZ Signal

You can always open an existing SigGenRZ signal file to modify it again, or to view it. When an existing SigGenRZ file is opened in SigGenRZ or an application, SigGenRZ will reset SigGen index to 1 and load all necessary files (including variable files and signal component files; whichever apply).

Note: SigGenRZ or the application loading a signal will look for needed files in the path(s) specified in the SigGenRZ file. An error will be displayed when a file cannot be found or loaded, and an incorrect signal will be generated. You must remember that when you move a variable file or a signal component file to another location or delete it from its current location, related SigGenRZ files should be updated to reflect the change.

To open an existing signal:

1. Select **Open** from the **File** menu, or click on the **Open** button on the toolbar to open the **Open File** dialog box.
2. Click on or type in the desired file name.
3. Click **OK**.

Note: The selected signal file must be in SigGenRZ format. These files are typically saved with the extension *.sig. Trying to open a non-SigGenRZ file will generate an error and crash the software.

To open a recently updated signal:

- Select the desired file name from the **File** menu. SigGenRZ displays the names of the last 4 signal files saved at the bottom of the file menu.

Importing a Non-SigGenRZ Signal

You may import non-SigGenRZ waveforms into SigGenRZ. SigGenRZ signals are built from segments. Segments, in turn, are built from components. When importing a non-SigGenRZ signal, you are actually using this signal as a SigGenRZ component. This means that all non-SigGenRZ signals must be imported as components. See *Importing Components* on page 57 for more information.

Saving Signals

Signals may be saved in the following formats:

- **SigGenRZ format** - Files saved in the SigGenRZ format contain signal, segment, and variable parameter information that can be used to generate the signal through multiple intervals, or SigGenRZ Indices (SGIs) when used in BioSigRZ. The default extension for files saved in SigGenRZ format is *.sig*.
- **Binary file format -** You can also save signals as time-domain waveforms stored in 16-bit integer binary file format. These files are given the default extension of *.16* unless you change it to another one. Since such files are raw data files containing only the current waveform, they do not include any other information in the SigGenRZ files, and each interval must be saved separately. You must display the waveform of another interval before you save that interval. Raw data files can be used as a SigGenRZ component, or with many commercial programs and your own run-time programs.

Saving Files in SigGenRZ Format

To save a new file in SigGenRZ format:

1. Select the signal by clicking anywhere in the signal window.
2. Choose **Save** or **Save As** from the **File** menu, or click the **Save As** button on the toolbar.

The **Save As** dialog box will be displayed. You may assign a name to your new signal.

To save an existing file:

1. Select the signal by clicking anywhere in the signal window.
2. Select **Save** from the **File** menu.

The file will be saved with its current name.

Saving Files in Binary Format

To save a file in binary format:

1. Select the signal by clicking anywhere in the signal window.
2. Choose **Save Wave** from the **File** menu, or click the **Save Wave** button.

The **Save Wave File** dialog box will be displayed. You may assign a name to your new signal. The displayed waveform will be saved as a 16-bit binary file.

The Signal Design Process

SigGenRZ allows you to build a variety of complex signals. More importantly, however, SigGenRZ allows you to specify how these signals will be presented as stimuli.

How Does SigGenRZ Build Signals?

SigGenRZ builds signals in a hierarchical fashion.

- A SigGenRZ signal consists of at least one segment.
- A segment consists of one to three components. Segments without components can also exist.

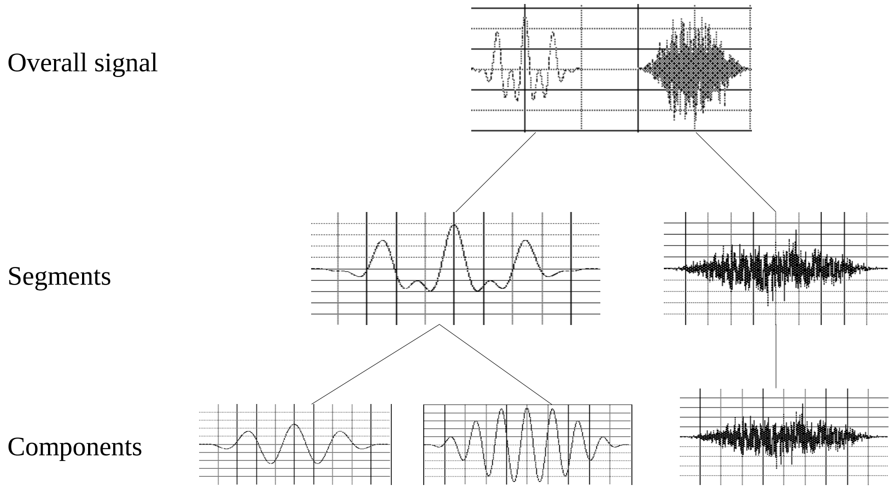
To build a signal with SigGenRZ, all you have to do is specify the parameters of each signal, segment, and component.

Steps

1. Define "signal" parameters including: duration of the entire signal, devices to be used for playback, overall attenuation, calibration, amplitude and phase correction, and variables.
2. Define "segment" parameters including: segment duration, segment onset, gate, component generation method and application method, and segment level.
3. Define "component" parameters. There are various parameters to specify depending on the type of component to generate. These parameters may be in the time or frequency domains. It is these parameters that actually generate the component waveforms that comprise your signal.

Results

Defined components are additively combined to produce a segment. Segments may be added together or multiplied by each other to produce SigGenRZ signals.



The details of this process are explained more extensively in the chapters that follow. Illustrative examples are also provided in the second part of this guide.

Chapter 3 Preparing to Build a Signal - The First Step

Prior to designing the constituent segments of a signal, you should define the parameters that will determine the entire signal, as well as any variables that might be used to change signal, segment, or component parameters as a function of the presentation interval, or SigGen index (SGI).

Device Selection and Setup

When a new SigGenRZ file is created, the default device selection and setup are configured automatically, including:

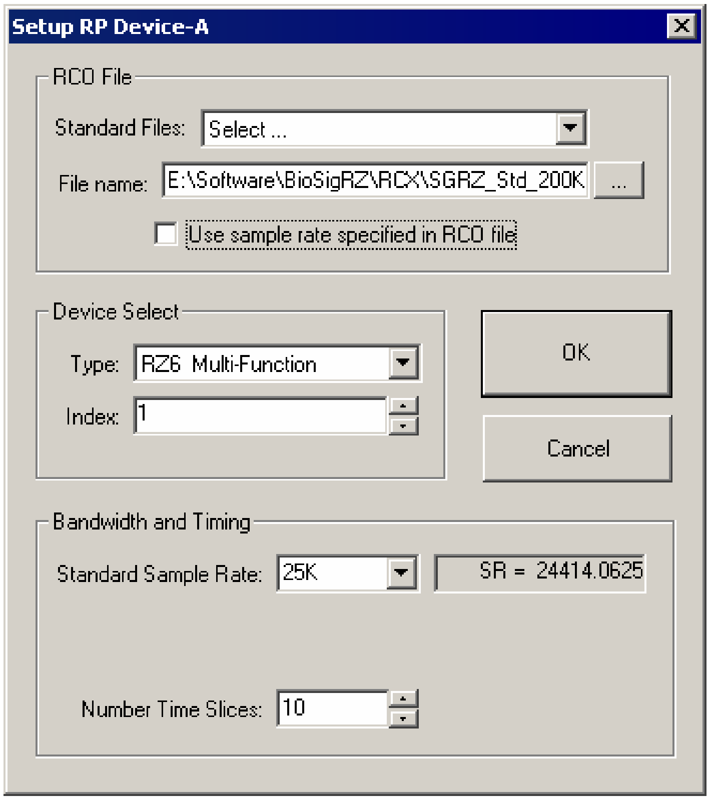
- Device selection
- Stimulus playback circuit
- Sample Period of the device

Device parameters are defined from the Setup RP Device dialog box, which can be accessed from the File menu.

To access the RP device dialog box:

- Click **Target Device** from the **File** menu.

The RP device setup dialog box is divided into three sections. To display all three sections, clear the **Use sample rate specified in RCO file** check box.



Setup RP Device-A

RCO File

Standard Files: Select ...

File name: E:\Software\BioSigRZ\RCX\SGRZ_Std_200K

☐ Use sample rate specified in RCO file

Device Select

Type: RZ6 Multi-Function

Index: 1

Bandwidth and Timing

Standard Sample Rate: 25K

SR = 24414.0625

Number Time Slices: 10

RCO file

Users can choose from the RCO files that come with SigGenRZ or make their own. Two Standard RCO files come with SigGenRZ:

- SGRZ_Std_200K.rcx
- SGRZ_Cal_200K.rcx

The first file is the standard stimulus play circuit. This circuit is compatible with both sweep play and continuous play configurations.

The second file is the standard calibration circuit. This circuit contains the necessary circuit components used for calibration in BioSigRZ.

Because SigGenRZ is targeted to the RZ6 processor, the maximum sample frequency for each RCO file is 200 kHz.

For more information about RCO files see the *RPvdsEx Manual*.

Device Select

Under Device Select, users may select the device type; however, SigGenRZ only supports the RZ6 Multi I/O processor at this time.

Type

All System 3 devices are accessible from the dialog box. By default, the RZ6 Multi-Function processor is selected.

Index

Devices are numbered logically beginning with one. A device index of zero means that the device type has not yet been selected. Logical device numbers may not correspond to the physical device organization. To ensure that the devices are ordered consistently each time the system is used make sure that devices are turned on in the same order each time the system is turned on.

If the hardware set-up includes more than one RZ6, the logical order of devices can be determined using the zBUSmon utility provided with the TDT Driver installation.

Bandwidth and Timing

The Bandwidth and Timing settings are only visible when the Use sample rate specified in RCO file check box has been cleared. It is recommended that you use the sampling rate defined in the standard circuit files provided by SigGenRZ.

Standard Sample Rate

When the Bandwidth and Timing settings are displayed, the standard sample rate of the RCO file can be modified by selecting a new rate from the drop-down list.

The RZ6 allows the user to set the sample period in discrete steps of 6k, 12k, 25k, up to 200 kHz. The RZ6 can deliver excellent waveforms almost up to the Nyquist limit (.85x). Choose the sample rate that is 2.2x greater than the largest frequency.

Signal Setup

Overall signal parameters include:

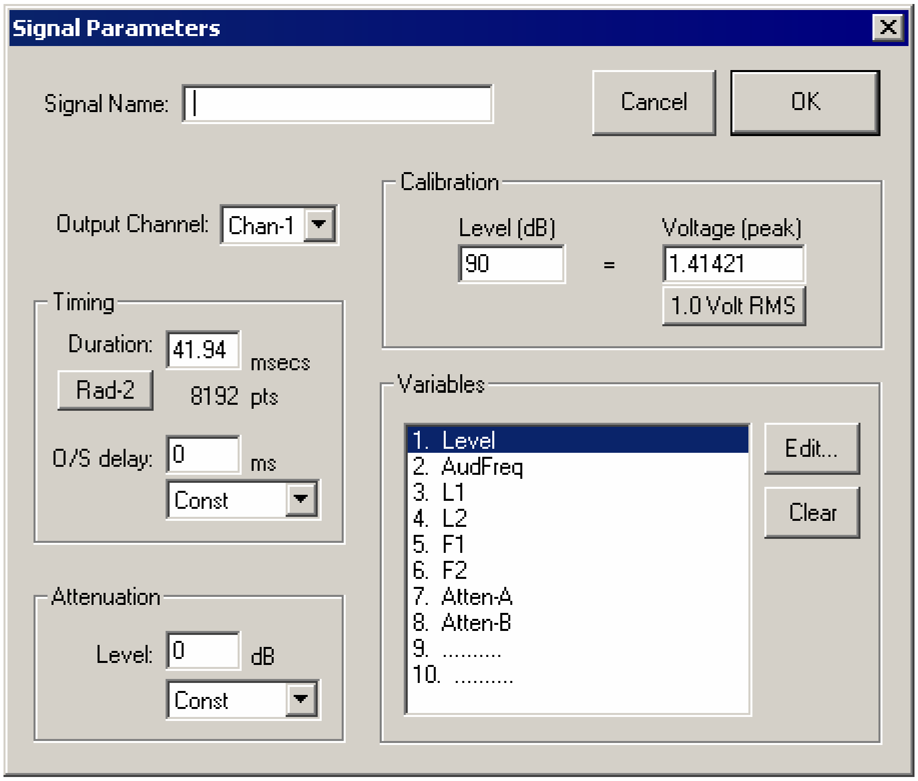
- A name for the signal
- The duration of the signal
- The playback D/A channel

- o The signals attenuation
- o Calibration specifications
- o Variable definitions

Note: Signal parameters are defined from the Signal Parameters dialog box.

To access the Signal Parameters dialog box:

➢ Click **Signal** from the **Modify** menu, or click on the **Modify Signal** button on the toolbar.



Signal Parameters

Signal Name:

Output Channel: Chan-1

Timing

Duration: 41.94 msecs

8192 pts

O/S delay: 0 ms

Const

Attenuation

Level: 0 dB

Const

Calibration

Level (dB) 90 = Voltage (peak) 1.41421

Variables

1. Level
2. AudFreq
3. L1
4. L2
5. F1
6. F2
7. Atten-A
8. Atten-B
9. ....
10. ....

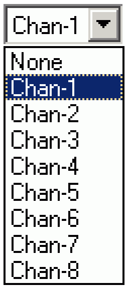
Signal Name

Signals may be named for user reference. Names may not exceed 15 characters in length.

Output Channel

SigGenRZ supports up to eight D/A channels for signal definition, and one channel for signal preview. Prior to signal preview, you must select the playback channel.

Output Channel:



Chan-1

None
Chan-1
Chan-2
Chan-3
Chan-4
Chan-5
Chan-6
Chan-7
Chan-8

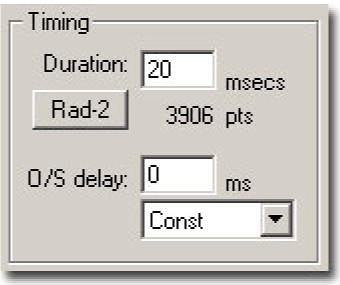
The RZ6 contains two D/A channels, select chan-1 or chan-2 for signal generation.

To select/deselect a D/A Channel for signal preview:

- Click the desired channel.

BioSigRZ reads and uses the D/A channel information stored in the SigGenRZ file.

Timing



Timing

Duration: 20 msecs

3906 pts

O/S delay: 0 ms

Const

Duration

The Duration field of the Signal Parameters box defines the total duration of the signal in milliseconds. The value entered in this field determines the scaling of the Signal Window x-axis.

Radix-2

Sometimes a duration corresponding to a radix-2 number of sampling points is desired.

For example, one scenario could be generating a waveform as short as possible while collecting enough points for Fast Fourier Transformation (FFT) analysis.

SigGenRZ provides a quick function button that truncates the value in the Duration field so that the total number of points in the signal is rounded to the closest radix-2 number. This eliminates the need to calculate a duration with desired radix-2 samples.

To truncate to radix-2:

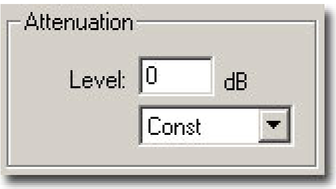
1. Enter the approximate duration in the **Duration** field.
2. Click the **Rad-2** button.

The duration will be recalculated so that the total number of points is defined in terms of radix-2. The new duration and number of data points will be displayed. If you feel a different radix-2 number is what you want, you can double the duration value (or divide it by 2), and then click the Rad-2 button to get the next radix-2 number.

O/S Delay

The O/S Delay field specifies the onset delay of the signal in milliseconds. Onset delay is defined as the delay between the initiation of an SGI and the beginning of signal presentation.

Attenuation



Attenuation

Level: 0 dB

Const

The attenuation field is used to specify the attenuation setting in the RZ6 processor. The drop-down menu allows you to control attenuation through the use of a user defined variable. See the

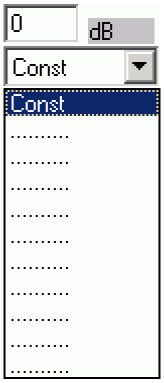
RZ6 Multi I/O Processor in the *System 3 Manual* for more information on the RZ6 programmable attenuation.

When signals are presented to D/A devices, only 16-bit integer samples will be converted to analog signals. As a result, small amplitude signals will lose their details and appear distorted, and the signal to noise ratio will be reduced. To make good use of the dynamic range of the D/A devices, you should make your signal’s amplitude as large as software and hardware allow, and use attenuators to adjust the signal level after D/A conversion to produce desired signal level.

Level

The level field specifies the level of programmable attenuation.

Level:



0 dB

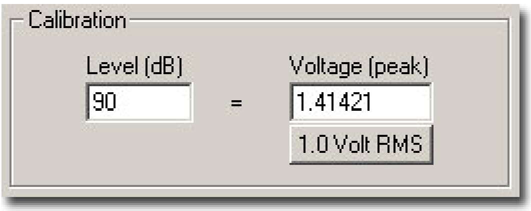
Const

Constant or Variable Level By default, Level is defined as Const, and therefore remains constant, regardless of the current SigGen index value. Level may be varied as a function of the current SigGen index value by assigning a predefined variable. See *Variables* on page 18 for more information.

To set Level to a constant or a predefined variable:

- Click **Const** or the desired variable name in the drop-down menu. If Level is a constant, enter the value in the **Level** textbox.

Calibration



Calibration

Level (dB) 90 = Voltage (peak) 1.41421

In designing signals in SigGenRZ, a decibel (dB) value is used to specify a level of a segment gain, or a component level. Since all signals will eventually be presented in terms of Volts, a dB-Volt relation or calibration must be specified. By default, all measurements of level in SigGenRZ are calculated in dB Volts with a reference of 1 Volt peak. You may calibrate measurements of level by setting a specific Voltage equal to a designated dB level, or using one Volt RMS equal to the dB level.

Level

The Level field specifies a level in dB that equals the specified number of Volts (peak), or one Volt RMS.

Voltage

The Voltage field specifies the number of Volts of the signal peak amplitude that equal the specified number of level units (dB). For example, in the specification (90 dB = 1 Volt (peak)), a tone that is specified to be 90 dB will produce a tone with a 1-Volt peak amplitude.

One Volt RMS

Clicking this button sets Voltage (peak) field equal to 1 Volt RMS (1.41421 (peak) Volts). This is convenient when you want to make your calibration in terms of RMS values.

Example: Calibrating units to reflect SPL

You may wish to calibrate SigGenRZ's level units so that they correspond to the sound pressure level (SPL) output of your system.

To calibrate SigGenRZ:

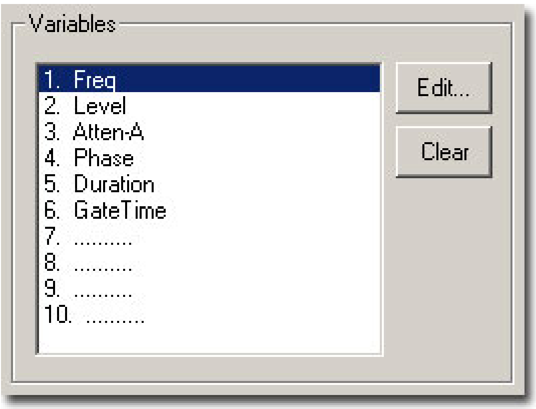
1. Generate a pure tone signal of a specified number of peak Volts.
2. Set Voltage equal to this number of peak Volts.
3. Measure the output of your system in SPL.
4. Set Level equal to the measured output in SPL.

Variables

SigGenRZ allows you to specify variables that may be used to change signal or segment parameters as a function of the presentation interval, or SigGen index (SGI).

Up to ten variables can be defined in each signal. Variable values can be altered systematically, read in from a file, or specified by you prior to each presentation interval.

Variables may be created, edited, and cleared in the Signal Parameters dialog box or from the Variables Area of the main window.



1. Freq
2. Level
3. Atten-A
4. Phase
5. Duration
6. GateTime
7. ....
8. ....
9. ....
10. ....

The Variables Area of the Signal Parameters Dialog Box

Edit

A new variable (except an alternating variable) can be defined at any available place of the Variables box.

To create a new variable:

1. Click on an available variable place (..........) in the Variables box.
2. Click the **Edit** button.

or

- ➢ Double-click on the available variable (..........) in the Variables box.

This will open the Signal Variables dialog box.

To edit an existing variable:

1. Click the desired variable in the Variables box.
2. Click the **Edit** button.

   or

- ➢ Double-click the variable in the Variables box.

This will open the Signal Variables dialog box.

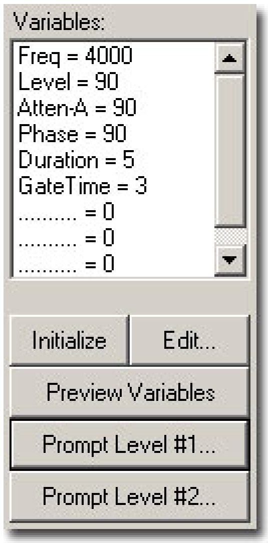
Clear

To delete an existing variable:

1. Click on the desired variable in the Variables box.
2. Click the **Clear** button.

Important note!: When a variable is deleted, SigGenRZ will remove its name from the Variables box, but will not always reset the parameters corresponding to this variable to their default values. Any parameter which uses this variable will still point to the place of this variable, and use zero as the variable value. Consequently, when a new variable is defined at this variable place, those parameters which use the deleted variable will automatically use the new variable instead. You should always remember that when you clear a variable, you must change all the signal/segment/component parameters which use the variable.

Variables Group Box



Freq = 4000
Level = 90
Atten-A = 90
Phase = 90
Duration = 5
GateTime = 3
.... = 0
.... = 0
.... = 0

The Variables group box displays a list of all currently defined variables along with their current values corresponding to the current SGI value.

When more than one signal is used in an experiment, the SigGen index of the channel-1 signal will be the active SGI in the application. As a result, when two signals are created and used in BioSigRZ, variables used in the signal assigned to the second channel should also be defined at the same variable locations in the first signal. For more information about creating variables, see *Creating Variables*, page 18.

Initialize

Clicking the Initialize button sets the SigGen index (SGI) back to 1, causing any variables associated with the current signal to be initialized to their starting values. Clicking Initialize also causes any variable files to be read. When using a variable file, it is a good idea to initialize the variable(s) prior to assigning the variable(s) to parameters.

Edit

Clicking the Edit button will open the Signal Variables dialog box. Clicking any variable or empty variable position will select that variable or position. When the Edit button is clicked, the selected variable (or position) will be edited (or created) in the Signal Variable dialog box. Clicking Edit without selecting any variable or position will cause the first variable to be edited/created. Variables may be deleted in Signal Parameters dialog box.

To edit an existing variable:

1. Click the desired variable in the Variables box.
2. Click the **Edit** button.

This will open the Signal Variable dialog box and you can edit the parameters of this variable.

To create a new variable:

1. Click on one of the available variable positions (.......... = 0) in the Variables group box.
2. Click the **Edit** button.

This will open the Signal Variable dialog box. You can assign a name to this variable and select other parameters.

Preview Variables

Once you have defined the signal variables, it is possible to preview how they will change across intervals. Clicking the Preview Variables button opens the Variable Preview dialog box.

Prompt Level #1

You may wish to specify the value of a variable on the fly during the presentation of a stimulus signal. From the Signal Variable dialog box, you may create user-prompts. When you save your SigGenRZ signal file (see *Working with the Complete Signal*, page 65 and *Saving Signals*, page *10)*, user-prompts will be saved as well and will be available when using the file in BioSigRZ.

When a user-prompt variable is defined, you can click the Prompt Level #1 or Prompt Level #2 button to access the Input Request dialog box and enter a new value for this variable. Current user-prompt variable value(s) will be used for all intervals of the signal preview.

To define Prompt Level #1:

1. Edit an existing variable or create a new variable and select the **Prompted (Level 1)** variable **Method**.
2. Click **OK**.
3. Click on the **Prompt Level #1** button to open Input Request dialog box.
4. Enter the desired value in the Unknown Prompt field.

Note: The Prompt/Comment text box in the Signal Variable dialog allows you to specify your own prompt message for the Input Request dialog box.

From the Input Request dialog, you can view the prompt message and enter any test value you may wish to use for signal preview.

Prompt Level #2

In some cases, you may wish to vary parameters in a nested fashion. For example, you may wish to vary the frequency of a pure tone. For each frequency, you may in turn wish to vary the level.

You may specify a user prompt for a nested variable by using Prompt Level #2. In our example, this would be specified for level.

Note: Current versions of BioSigRZ does not support nested variables or Prompt Level #2.

To define Prompt Level #2:

1. Edit an existing variable or create a new variable and select the **Prompted (Level 2)** variable **Method**.
2. Click **OK**.
3. Click on the **Prompt Level #1** button to open Input Request dialog box.
4. Enter the desired value in the Unknown Prompt field.

If you have multiple Prompt Level #1 or Prompt Level #2 variables defined, the Input Request dialog box will appear repeatedly until enough values are entered for all Prompt Level #1 or Prompt Level #2 variables.

Creating Variables

You may change the value of signal and segment parameters as a function of the SigGen index (SGI) value by defining variables. From the Signal Variable dialog box you can define the names of up to ten variables, as well as specify starting values, minimum values, maximum values, and the manner in which the value will be varied.

You may access the Signal Variable dialog box from the following:

- The Variables area of the main SigGenRZ window.
- The Variables group box of the Signal Parameters dialog box.

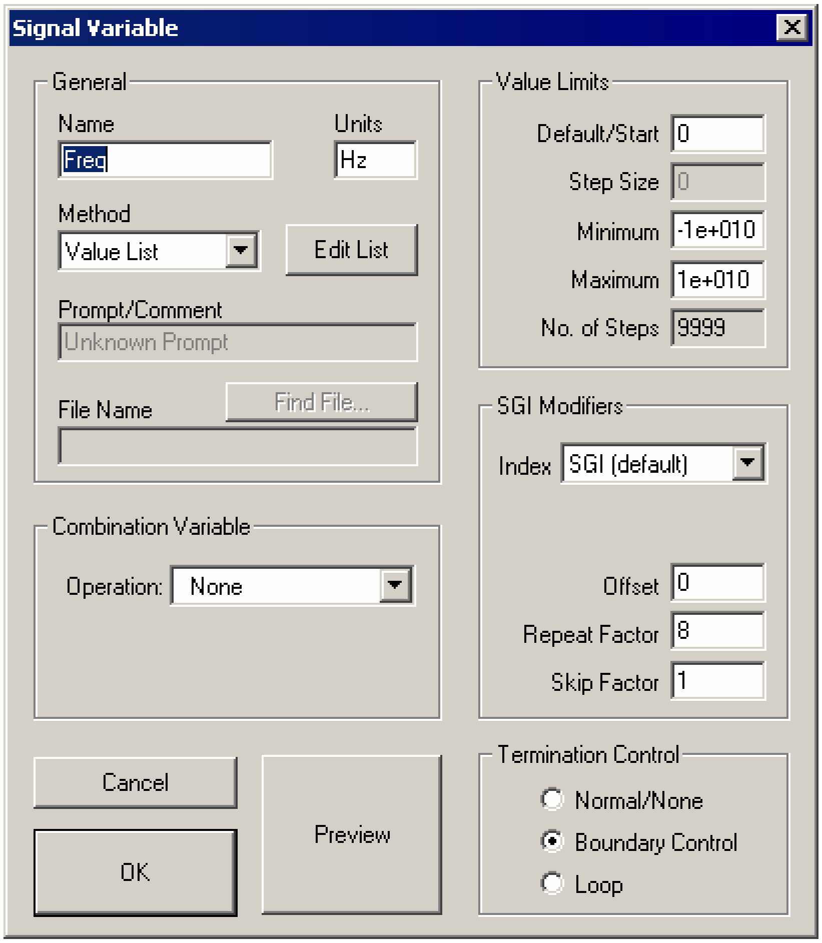
To access the Signal Variable dialog box

Do one of the following:

- Click a variable in the Variables box to edit an existing variable.

  or

1. Click an available variable place (..........) in the Variables box to create a new variable.
2. Click the **Edit** button.



Signal Variable

General

Name: Freq

Units: Hz

Method: Value List

Prompt/Comment: Unknown Prompt

File Name:

Combination Variable

Operation: None

Value Limits

Default/Start: 0

Step Size: 0

Minimum: -1e+010

Maximum: 1e+010

No. of Steps: 9999

SGI Modifiers

Index: SGI (default)

Offset: 0

Repeat Factor: 8

Skip Factor: 1

Termination Control

○ Normal/None

● Boundary Control

○ Loop

General Variable Parameters

General variable parameters include:

- Name
- Units
- Method

Name

Variables are defined using names that may not exceed 12 characters in length. You can enter a variable name by typing in the Name field. When you change a variable name or its parameters, SigGenRZ will also make changes to those parameters using this variable when editing variable is completed. Using a name reflecting the physical nature of the variable is highly recommended.

Units

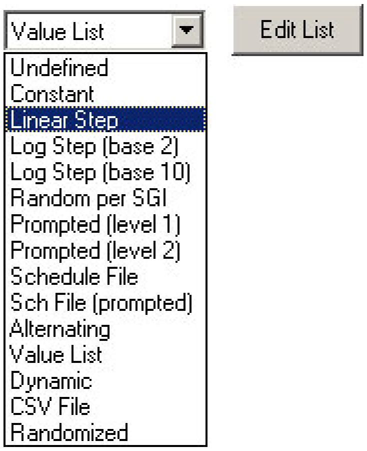
In the Units field, you may enter up to 5 characters to describe the variable's intended unit of measure. The unit value you entered here is only a reminder, not a standard for data conversion.

Three basic units used in SigGenRZ are: decibel (dB) for level, millisecond (ms) for duration, and Hertz (Hz) for frequency. You should use a correct unit for this field.

Note: The actual unit of measure assigned to the variable will be determined by the SigGenRZ parameter with which the variable is associated. For example, you may have defined a variable named "Freq" and described its Units as "Hz." However, if you assign "Freq" to a duration parameter, its unit will be milliseconds.

Method

Method defines the mechanism (function) of altering variable values as a function of the SigGen index (SGI). There are 14 methods available in SigGenRZ.



Variable methods include:

- **Undefined** - Undefined means that no Method has been defined for the current signal variable. The variable will be zero for all the signal intervals.
- **Constant** - This method is most useful when more than one SigGenRZ parameter will be set to the same value. Constant defines a variable that will always have the same value as given in the Default/Start field of the Value Limits box. This value will be a floating-point constant between 1.2E-38 to 3.4E+38.
- **Linear Step -** For each SigGen index (SGI) increment, Linear Step defines a variable that will increment by the Step value specified in the Value Limits box.
- **Log Step (base 2)** - Log Step (base 2) defines a variable that will perform as a base 2 exponential function of SGI:

  $$\text{Variable} = \text{Start x } 2^{\text{Step x (SGI-1)}}$$

  The Start and Step values are defined in the Value Limits group box of the Signal Variable dialog box.
- **Log Step (base 10)** - Log Step (base 10) defines a variable that will perform as a base 10 exponential function of SGI:

  $$\text{Variable} = \text{Start x } 10^{\text{Step x (SGI-1)}}$$

  The Start and Step values are defined in the Value Limits group box of the Signal Variable dialog box.
- **Random per SGI** - The value used for the variable is a random number between the Minimum and Maximum set in the Value Limits group box. The random numbers are generated with a uniform distribution within an SGI.

  A Step Size of zero will give a full floating-point random value within the specified range. Use a non-zero to define the resolution of the result (e.g.: a Step Size of 0.1 might produce values of 3.7, 2.9, 7.2, etc.).

- **Prompted (level 1)** - To specify the value of a variable on the fly during the presentation of a stimulus signal, you may create user prompts. These prompts will be available in TDT's application programs.

  If you have defined a level 1 prompt for the selected variable, the Prompt Level 1 button will be enabled, and you can use this button to enter variable values when you preview the signal.

- **Prompted (level 2)** - Using this method, you can be prompted for a value at run-time. Level 2 prompts are used for 2nd level nested variables.

- **Schedule File** - Variable values are read from a schedule file. The file name is specified in SigGenRZ and is used by BioSigRZ.

  Each variable in a variable file can have up to 100 values.

- **Sch File (prompted)** - Variable values are read from a schedule file. The file name is specified by you during run-time. One usage of this method is to allow user to use more user-defined variable values. For example, when you want to use more than 100 specific variable values, you can put them into more than one variable file, and access them one after another.

  File prompt will be available for applications supporting this feature. While previewing signals in SigGenRZ, you can display this prompt and assign schedule file(s) by clicking Prompt Level #1 button in SigGenRZ Signal Window.

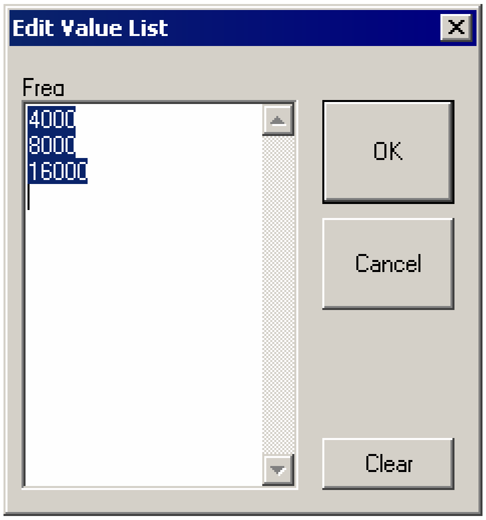
- **Alternating -** The value of the variable alternates between the values defined in the Minimum and Maximum fields. The variable value does not depend on SGI value, but on the number of the continuous presentation of the current signal interval. In the Variables group box, an alternating variable must be defined next to another predefined variable, or as the first variable. See the *BioSigRZ Manual* for more information.

  Note: Due to the nature of this variable, this feature is not available during Play or Play All in SigGenRZ since the "continuous presentation of the current signal interval" condition is not met. However, you can test the alternating effect by repeatedly initializing variables. Alternating variables function when the SigGenRZ signal is used as a stimulus in TDT's BioSigRZ application.

- **Value list -** Variable values are specified in SigGenRZ by you during the signal design process. A maximum of 100 variable values can be specified for the variable.

To enter or modify a value list:

1. Choose **Value List** from the **Method** box.
2. Click the **Edit List** button.
3. Enter values to define the variable.



4000
8000
16000

- **Dynamic** - Dynamic variable values are controlled by the run-time application, usually based on a subject feedback. See the *BioSigRZ Manual* for more information.
- **CSV File** - The values for this variable type are reads from windows standard Comma Separated Value (CSV) files which allows for matrix (two dimensional) arrays where each row is an SGI and each column is a repeat. The CSV files used by this variable type are easily generated and saved using EXCEL or similar programs. A typical CSV file is shown below.

  1, 6,11,16,21,26

  2, 7,12,17,22,27

  3, 8,13,18,23,28

  4, 9,14,19,24,29

  5,10,15,20,25,30

  6,11,16,21,26,31

  7,12,17,22,27,32

  8,13,18,23,28,33

  9,14,19,24,29,34

  A variable using this CSV file would use the current SGI to index down to a specific row. For example, for SGI = 6 the row with numbers 6,11,16,21… would be indexed. If this variable was used in the component of a repeating segment, the first repeat would use the value 6 and the second would use 11 and third 16 etc. This allows the user to get full control of how repeats are computed and where they are applied to the overall waveform
- **Randomize** - The value used for the variable is a random number between the Minimum and Maximum set in the Value Limits group box. The random numbers are generated with a uniform distribution.

  A Step Size of zero will give a full floating-point random value within the specified range. Use a non-zero to define the resolution of the result (e.g.: a Step Size of 0.1 might produce values of 3.7, 2.9, 7.2, etc.).

Prompt/Comment

If the defined method is Prompt, you will be defining the variable values. If the defined method is Sch File (prompted), you will be defining the name of the schedule files. In these cases, a user prompt will be necessary, and the text string(s) entered in the Prompt/Comment field will be used as the Prompt. A prompt may be up to 50 characters in length including spaces.

To define a Prompt/Comment:

1. Edit an existing variable or create a new variable and select **Prompted (level 1 or level 2)** or **Sch File (prompted)** from the **Method** dialog box.
2. Type in the desired text in the Prompt/Comment field.

File Name

If the chosen method is Schedule File, a schedule file must be specified. Schedule files are defined as an ASCII file with the file extension *.sch.

To define a schedule file

- ➢ Type in the file path and name

  or

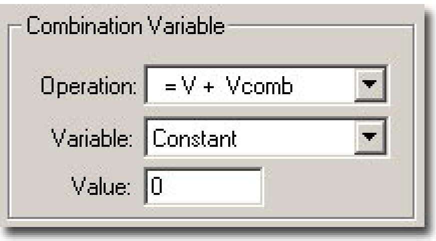
1. Click the **Find File** button.
2. Choose the file name.

When you choose Sch File (Prompted), you do not need to enter a file name at design time; however, you may need to supply the file when you run an application. Specifying a schedule file for a Sch File (Prompted) variable is not required when you preview a signal in SigGenRZ.

Combination Variables

In many cases, a simple variable is not sufficient to define a signal. For example, consider a signal consisting of two segments in which the first segment has a variable duration, and the second segment starts 10 milliseconds after the first segment. You could calculate the start times of the second segment by hand, or you could define a combination variable.

A combination variable is a variable whose value is calculated by performing a mathematical operation on another variable or a constant.



Operation: = V + Vcomb

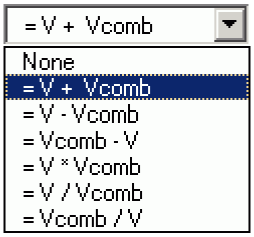
Variable: Constant

Value: 0

To Define a Combination Variable:

1. Open the Signal Variable dialog, either by choosing **Signal** under the **Modify** menu and then clicking the **Edit** button, or by clicking the **Edit** button on the side bar.
2. Name and define the combination variable in the General box.
3. In the **Combination Variable** box select an **Operation** to apply to this variable (**V** in the Operation formula). Combination variables can be defined using addition, subtraction, multiplication, and division.

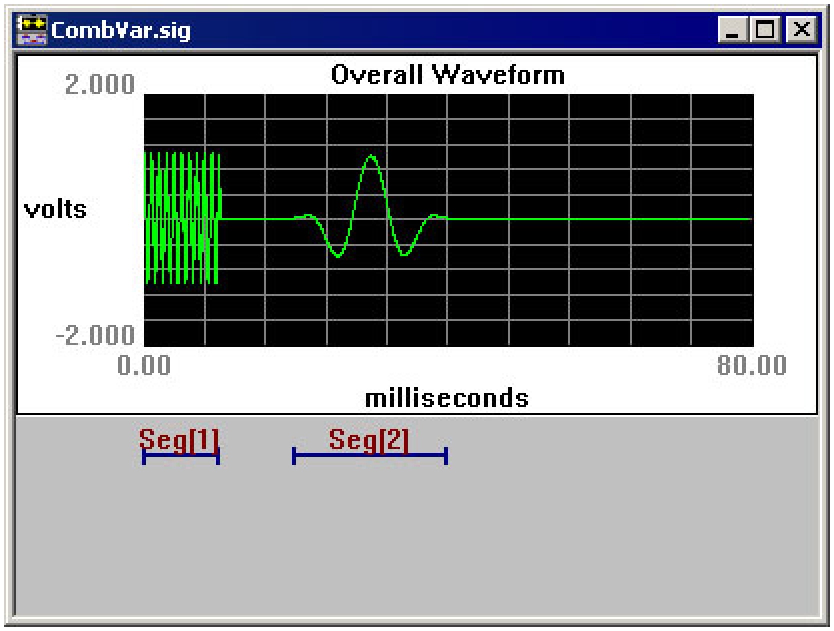
Operation:



The Variable drop down menu in the Combination Variable box is used to define **Vcomb** in the operation. **Vcomb** can be defined as either a constant, and its value entered in the Value box, or it can be defined as another variable.

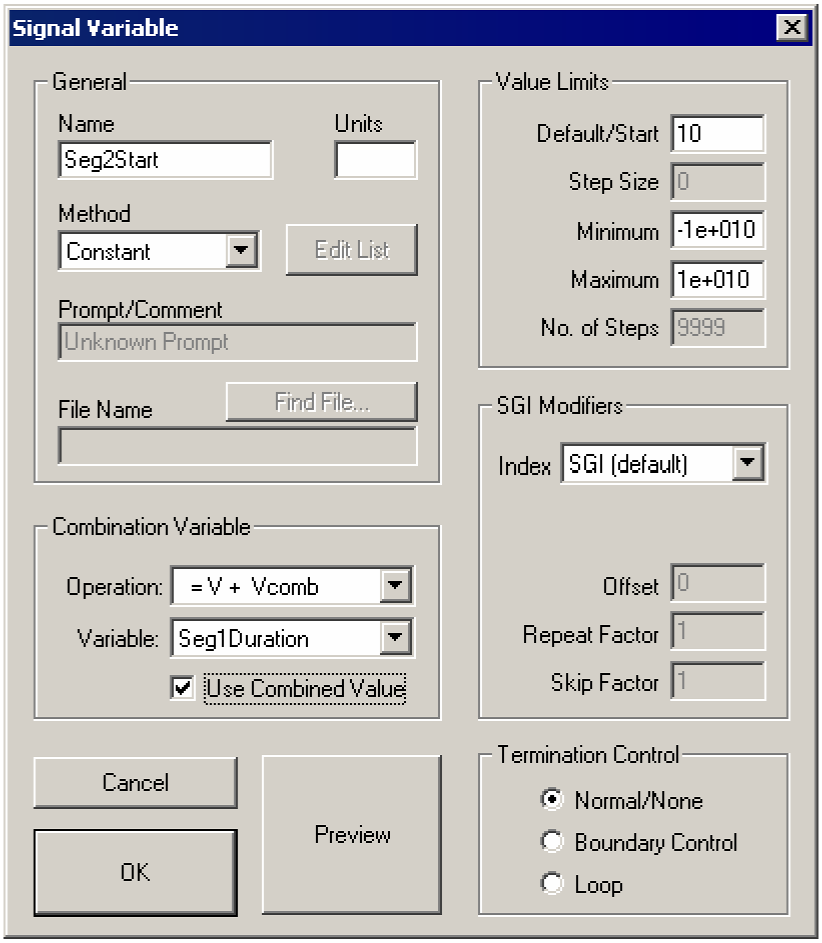
Combination Variable Example

This example illustrates how to use a combination variable to build a signal consisting of two segments in which the first segment has a variable duration, and the second segment starts 10 milliseconds after the first segment.



Seg1Duration is a simple variable that specifies the duration of the first segment. A combination variable, *Seg2start*, is defined so that segment 2 starts 10 msec after segment 1. *Seg2start* is defined with method Constant and Default=10.

In the Combination Variable box **Vcomb** is set equal to the variable *Seg1Duration*. **V+Vcomb** is selected so that *Seg2start* = 10 msec + *Seg1Duration*.



Signal Variable

General

Name: Seg2Start

Units:

Method: Constant

Prompt/Comment: Unknown Prompt

File Name:

Combination Variable

Operation: = V + Vcomb

Variable: Seg1Duration

☑ Use Combined Value

Value Limits

Default/Start: 10

Step Size: 0

Minimum: -1e+010

Maximum: 1e+010

No. of Steps: 9999

SGI Modifiers

Index: SGI (default)

Offset: 0

Repeat Factor: 1

Skip Factor: 1

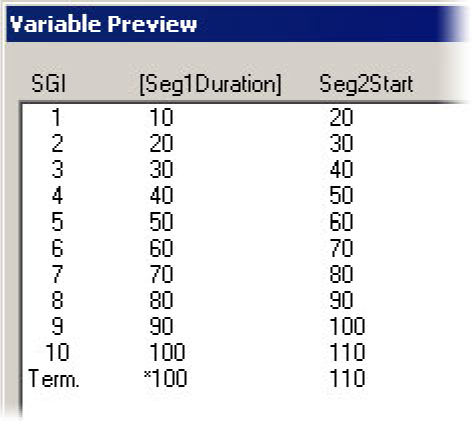
Termination Control

● Normal/None

○ Boundary Control

○ Loop

We can preview the results of the combination variable operation by selecting Preview. The uncombined value of the combination variable *Seg2start* is displayed. To see the results of the combination operation, check the Show Combined Values box. In this case you can see that *Seg2start* begins 10 ms after *Seg1Duration*.



Variable Preview

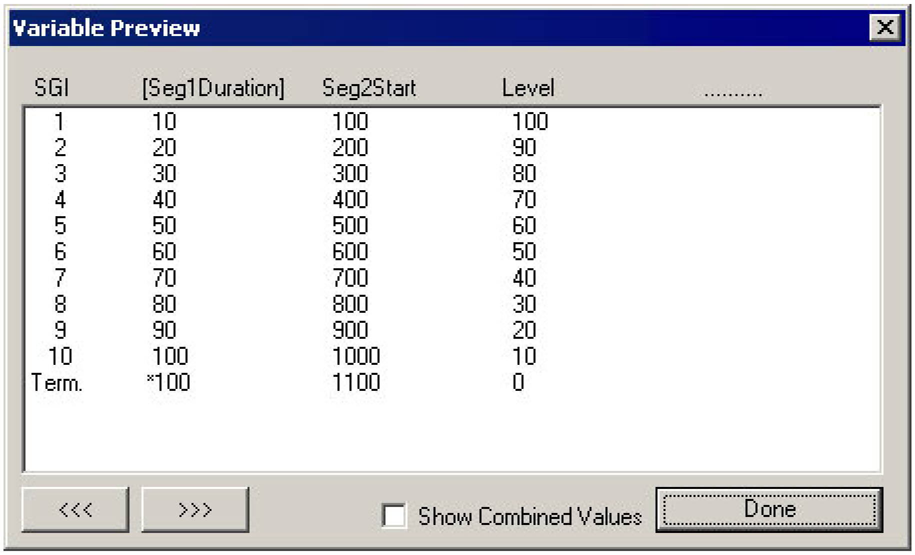
SGI	[Seg1Duration]	Seg2Start
1	10	20
2	20	30
3	30	40
4	40	50
5	50	60
6	60	70
7	70	80
8	80	90
9	90	100
10	100	110
Term.	*100	110

Previewing Variable Values

You may wish to preview the manner in which parameter values will vary as a function of the SGI. Variable values may be previewed from the Variable Preview dialog box.

To open the Variable Preview dialog box:

- Click the Preview button.



SGI	[Seg1Duration]	Seg2Start	Level	....
1	10	100	100	
2	20	200	90	
3	30	300	80	
4	40	400	70	
5	50	500	60	
6	60	600	50	
7	70	700	40	
8	80	800	30	
9	90	900	20	
10	100	1000	10	
Term.	*100	1100	0	

The Variable Preview dialog box displays the current *stimulus schedule*. Variable values are shown for each SGI.

Scrolling

The Variable Preview dialog box can display up to five variables. You may view additional variables by scrolling to the right or the left.

To scroll to the right

- Click on the >>> button.

To scroll to the left

- Click on the <<< button.

To scroll up and down

- Use the vertical scroll bar.

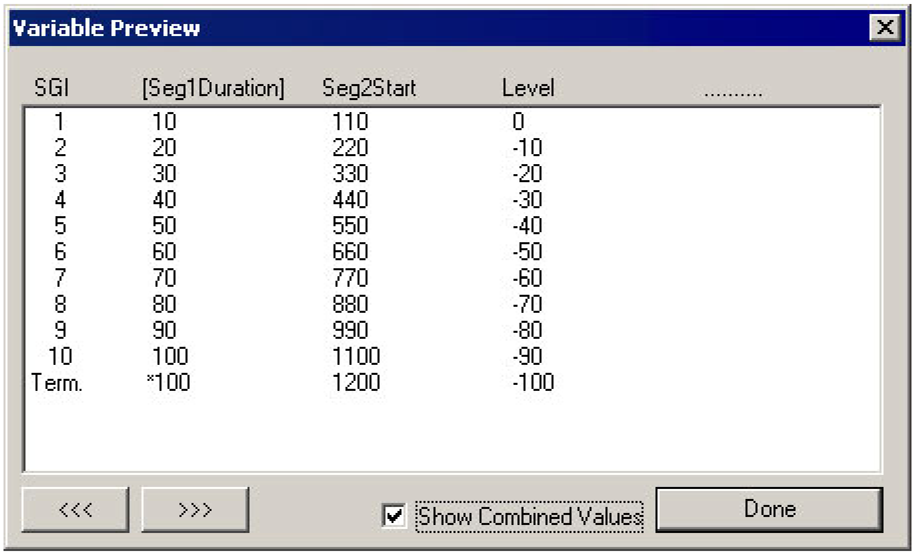
Showing Combined Values

You may have defined a combination variable. By enabling Show Combined Values, you may display the values of this combination variable as a function of the SGI. When Show Combined Values is not enabled, only their primary values will be displayed for combination variables.

To enable Show Combination Values

- Check the Show Combination Values box.

Enabling this feature would cause the previous stimulus schedule to appear as follows:

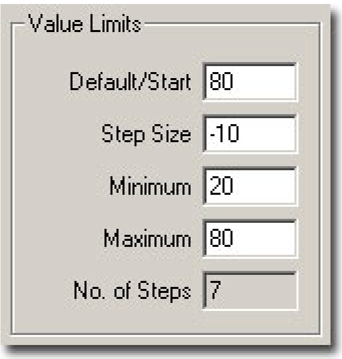


SGI	[Seg1Duration]	Seg2Start	Level	....
1	10	110	0	
2	20	220	-10	
3	30	330	-20	
4	40	440	-30	
5	50	550	-40	
6	60	660	-50	
7	70	770	-60	
8	80	880	-70	
9	90	990	-80	
10	100	1100	-90	
Term.	*100	1200	-100	

☑ Show Combined Values

Value Limits

From the Value Limits group box you can define the manner in which a variable value changes as a function of the SigGen index (SGI) value. You can also define the range of allowable values for the variable. For combination variables, these values will only apply to primary variable(s), and will not limit the combination variable result. However, a zero will be used when the combination variable result becomes a divide-by-zero.



Default/Start	80
Step Size	-10
Minimum	20
Maximum	80
No. of Steps	7

Default/Start

Default/Start defines the value of the variable when the SigGen index = 1. This is also the value for a constant variable. When prompt variables (Prompt Level 1, Prompt Level 2, and Prompted File) and Dynamic variables are defined, this value is also the default variable value before the value is specified.

Step Size

Step Size must be defined when using the following functions:

- Linear step
- Log Step (base 2)
- Log Step (base 10)
- Randomize
- Random per SGI

Step determines how much the variable will be systematically varied as a function of the SigGen index value. Default Step Size for a Randomized variable is 0, which causes SigGenRZ to use the full floating point precision to generate the variable values. A zero Step Size for other functions will define a constant variable.

Minimum

In the Minimum field, you define the minimum allowable value for the variable.

Maximum

In the Maximum field, you define the maximum value allowed for the variable.

Note: Minimum value must be smaller than Maximum, and Default/Start must be within Minimum and Maximum range.

No. of Steps

This field displays the total number of steps between the minimum and maximum variable values, when appropriate. This field is not editable.

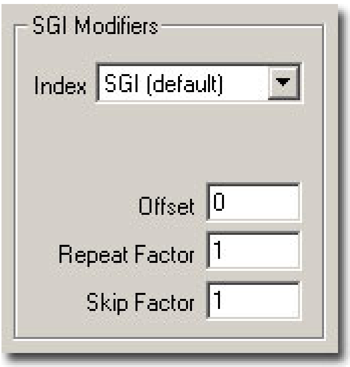
SGI Modifiers

SigGen index (SGI) is a unique number which starts at 1 and increments by 1 when an interval is presented. This SGI applies to every variable by means of SGI Modifiers. These SGI Modifiers become important when SigGenRZ is used to design signals with complicated variables or nested variables.

By default, a stimulus presentation begins with the SGI = 1. Often, the stimulus signal generated at SGI = 1 is presented until a desired number of response signals have been attained. The SGI is incremented by 1 then all variables and the entire signal are updated. This process continues until all SGIs have been presented. Through use of SGI Modifiers, it is possible to customize the process of incrementing the SGI for individual variables.

You may:

- Start variable calculation with an offset to SGI.
- Repeat SGI.
- Skip a specified number of SGIs.
- Create nested variables



SGI Modifiers

Index: SGI (default)

Offset: 0

Repeat Factor: 1

Skip Factor: 1

Index

By default, variables are indexed using the SigGen index or SGI. SGI indexing can also be replaced by another variable with an optional scalar applied. This is useful for a multitude of applications when one or more variables should be calculated based on a single control variable.

Offset

The initial SGI is always 1. When SGI is used to calculate variables, any deviations from this default value are termed offsets. If you wish to start stimulus presentation with a variable calculated based on a modified SGI = *n*, then enter *n* in the Offset field. If the variable's values are 0, 1, 2, 3, 4, 5, ..., corresponding to SGI = 0, 1, 2, 3, 4, 5, ..., an Offset = 2 will result in modified SGI = 2, 3, 4, 5, 6, ..., and so the variable's values 2, 3, 4, 5, 6, ..., correspond to SGI = 0, 1, 2, 3, 4, 5, ...

The default value of Offset is 0, that is, the modified SGI equals to SGI. The value of Offset must be no less than -10,000 and no greater than 10,000.

Repeat Factor

The Repeat Factor field specifies the number of times, *r*, an SGI will be used as modified SGI. For example, when *r* = 2, modified SGI will be 1, 1, 2, 2, 3, 3, ..., corresponding to SGI = 1, 2, 3, 4, 5, 6, ...

By default, Repeat Factor is set to 1 (use SGI once). Repeat Factor may be no less than 1 and no greater than 100.

By using a Repeat Factor in tandem with a Termination Control specification of Loop (see next section), you can create nested variables.

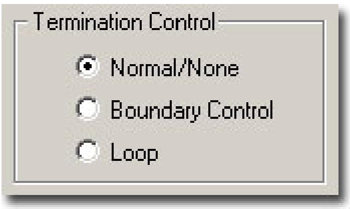
Skip Factor

The Skip Factor field specifies the number of the SGIs is to be ignored in the modified SGI. For example, if Skip Factor is 2, modified SGI will be 1, 3, 5, 7, 9, ..., corresponding to SGI= 1, 2, 3, 4, 5, 6, ...

By default, Skip Factor is set to 1. Skip factor may be no less than 1 and no greater than 100.

Termination Control

You can control the termination of SGI, and thus the signal play by choosing one of three types of termination control.



Termination Control

● Normal/None

○ Boundary Control

○ Loop

Normal/None

When Normal/None is selected, the variable will not be considered in termination control. Thus, if a minimum or maximum variable value is reached, signal play will not terminate, nor will the variable value loop. Continuous signal generation will be based on the final variable value. In this case, signal play will be terminated by other variables or manually.

Boundary Control

Boundary Control provides a method of experiment termination. When Boundary Control is chosen, the SigGen index (SGI) will not be incremented and signal play will halt if such an action would cause the value of the variable to be:

- Less than the Minimum value.
- Greater than the Maximum value.

Such failure to increment the SGI produces differing results, depending on the application.

Boundary Control in SigGenRZ: In SigGenRZ, when a boundary is reached and the SGI is no longer incremented, all signal generation halts.

Boundary Control in BioSigRZ: When a boundary is reached and the SGI ceases to be incremented, signal generation will continue for the current SGI, but no more response data will be collected.

Note: Specification of Minimum and Maximum does not provide boundary control. Minimum and Maximum simply define the limits of parameter variation. The variable will not be allowed to go beyond either of these limits. With Boundary Control disabled, SigGenRZ will continue to increment the SigGen index and produce a signal during preview of multiple intervals, even if a Minimum or Maximum parameter value has been reached.

Loop

When Loop is chosen, parameters continuously loop through the series of variable values as SGI increments.

Using Schedule Files

When other functions cannot generate desired variable values, you may design custom variables using Schedule Files. By default, SigGenRZ looks for files with the *.sch* file extension.

What Is a Schedule File?

A schedule file is an ASCII text file that defines one or more user-customized variables by specifying the following:

- **Variable Name** - The variables in SigGenRZ must match the variables defined in the schedule file.
- **Number of Repetitions for each variable** - Set this number equal to the number of times you want the signal repeated for a given variable value.
- **A list of variable values for each variable** - Up to 100 values may be specified for each variable in the schedule file.

Creating a Schedule File

Schedule Files may be created with any editor that saves ASCII text files such as notepad.

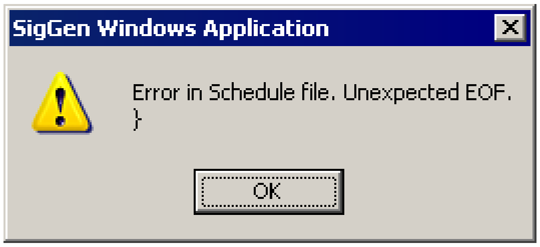
Schedule File Structure

Text Strings: Most users will want to include some text information in their Schedule File that describes the variable(s) defined. Such information can be placed at the top of the file, or between Variable Definition blocks.

Variable Definition: One or more variables may be defined in a Schedule File. Each variable is defined in a variable block. The beginning and end of a variable definition block are marked by a pair of brackets (**{}**). The variable definition block structure is defined on the following page.

{
Variable Name
Number of signal repetitions, r
Variable value 1
Variable value 2
....
Variable value n
}

Important!: This additional empty line under the closing bracket is necessary. Failure to provide this line will result in an Unexpected EOF (End Of File) error.



Error in Schedule file. Unexpected EOF.
}

Sample schedule file

Here is a sample schedule file.

This controls my Delta_T, separation between masker and probe.
{
Delta_T
4
82.0
66.0
58.0
54.0
52.0
51.0
50.0
}

This is a list for intensity of the masker. We'll run each intensity level
for all values of Delta_T.
{
Masker_Level
1
20.0
30.0
40.0
50.0
}

Nesting Variables

In some cases, you may wish to vary parameters in a nested fashion. For example, you may wish to vary the separation between a masker and its probe through the use of a variable such as Delta_T. For each Delta_T, you may in turn wish to vary the masker level.

The variable file shown above is an example of a two-level variable nesting. Please note the following:

- **Variables are defined in the order of nesting.**

  In this case, the separation between a masker and its probe, *Delta_T,* will be varied. For each value of Delta_T, the intensity of the masker, *Masker_Level*, will be varied. Thus, Delta_T is defined first; Masker_Level is defined second.

- **The Number of Repetitions parameter is used to control repetitions during nesting.**

  There are four values for Masker_Level. This means that for each value of Delta_T, Masker_Level level will be varied four times. This is indicated by setting the Number of Repetitions for Delta_T to four.

  For each Masker_Level, the signal will be played only once. This is indicated by setting the Number of Repetitions for Masker_Level to one.

- **Nesting variables can be defined in SigGenRZ without using a schedule file.**

  In the above cases, we can define Masker_Level with proper limits and Loop boundary control so it will loop over 4 values, and define Delta_T with a Repeat Factor of 4, we can obtain the same nested result.

- **Using a schedule file you can define more complicated nesting variables.**

  You can only achieve a two-level nesting if you do not use a schedule file.

Using Variables Defined in a Schedule File

To access variables in a Schedule File:

1. Type the variable name in the **Name** field.
2. Click the **Find File** button.
3. Select the desired file.

   or

- Type in the desired schedule file path and file name.

Note: SigGenRZ uses variable names to access variable values in schedule files. The variable name typed in the Name field must match exactly the variable name assigned in the schedule file.

Initializing Variables in a Schedule File

When a variable is created using a schedule file, variable values will be read as soon as the variable is accepted (clicking the OK button). However, when a file variable is edited (assigned with another schedule file), or the schedule file is altered, variable values may not be read automatically. At this time, when the current signal interval is previewed using Play button, the presented signal may not be the one with new variable values.

In many cases, the value of a file variable is read at the time it is assigned to a parameter. This value will be a function of the current SigGen index (SGI). In some cases, however, the variable value is not automatically read upon assigning the variable to a parameter. This can cause problems if the parameter must fall within a specific range. It is a good idea to manually initialize all file variables prior to assigning them to parameters.

To manually initialize a file variable:

- Click the **Initialize** button of the SigGenRZ main window.

Using Variables to Control System 3 Parameter Tags

When the demands of the required stimulus are beyond SigGenRZ’s basic signal design abilities, SigGenRZ can be used to control device properties through parameter tags. Parameter tags are a powerful component of the RCO file and can be used to control various aspects of a stimulus such as programmable digital filtering, waveform generation, and even hardware.

A good working knowledge of System 3 hardware and RPvdsEx is needed before modifying the SigGenRZ RCO files.

System 3 Hardware and Software

The RZ6 multi I/O processor is an easy-to-use, easy-to-program data acquisition and stimulus presentation device. Each RZ6 contains analog outputs and inputs, digital inputs and outputs and a development program, RPvdsEx. Compiled RPvdsEx files program the signal processor for stimulus generation and data acquisition using a drag-and-drop design environment. Circuits generated in RPvdsEx can be saved for use with other TDT software applications. For more information about creating RCO files, see the *RPvdsEx Manual*.

~

Chapter 4 Signal Segments - The Building Blocks of a Signal

When you are designing signals in SigGenRZ, you are actually designing sub-signals, segments, and specifying how each segment is related to any others temporally. You are also specifying how a segment will affect the entire signal. This chapter introduces Segment design and its characteristics.

Segments, in turn, are composed of one to three components. From the Edit Signal Segments window, you can define segment components.

The Segment Window

The Edit Signal Segments window allows you to specify all the parameters necessary for the definition of a segment.

To access the Edit Signal Segment window:

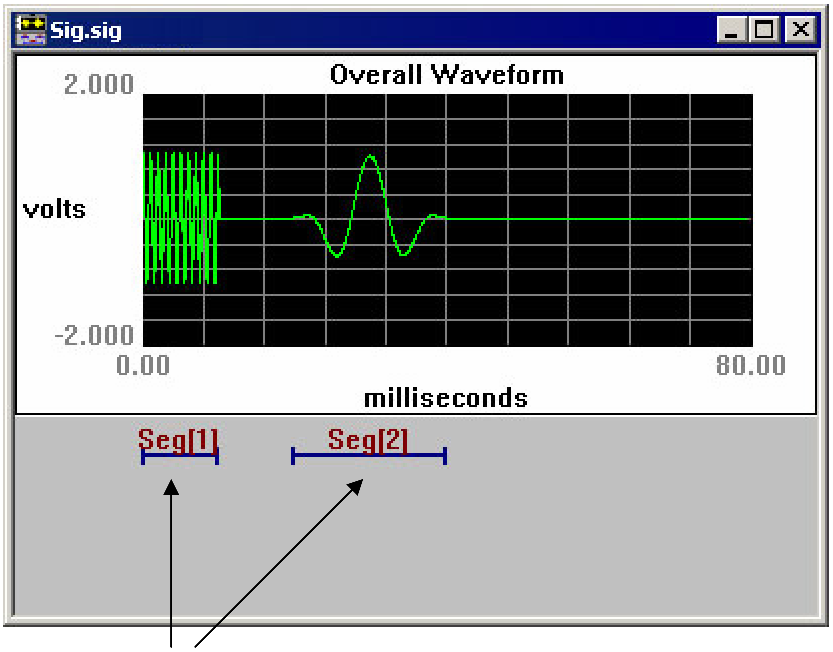
- Select **Segment** from the **Modify** menu of the main window.

  or

- Click the **Modify Segment** button on the toolbar.

  or

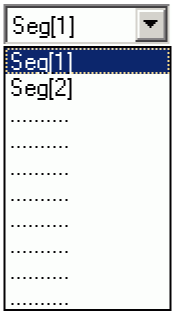
- Double-click the desired segment marker.



Note: Double-clicking a segment marker will prompt the signal segment window for that segment.

Creating/Selecting/Deleting a Segment

Select



Existing segments are listed in the Select drop-down combo box.

To select an existing segment:

- Click the desired segment in the **Select** box.

New

When you start to design a new signal, there is only one segment (empty) for the signal. You may need to create new segment(s) to complete the signal. New segments must be created in the Edit Signal Segment dialog box.

To create a new segment:

- Click the **New** button.

Delete

A segment can be removed (deleted) from the signal.

To delete a segment:

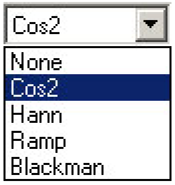
- Click the **Delete** button.

Note: A signal must contain at least one segment. You may not delete the last segment of a signal. If you wish to clear the entire signal, return to the SigGenRZ main window and close the unwanted signal window**.**

Segment Generation Options

Windowing/Gating Parameters

Gate Type



From the Gate Type box, you can select a variety of gating or windowing functions, including:

Gating functions

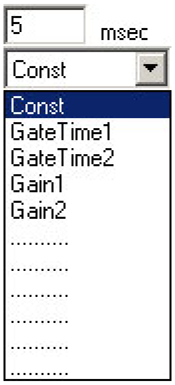
- Cos^2 (10% - 90%)
- Ramp
- Blackman

Windowing functions

- Hanning

In cases where gating or windowing is not necessary (e.g., click generation), choose None**.**

Gate Time



Gate Time specifies the rise/fall times of the gating function in milliseconds.

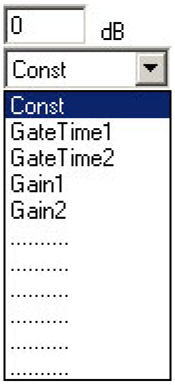
Constant or Variable Gate Time By default, Gate Time is defined as Const, and therefore remains constant throughout the preview process, regardless of the current SigGen index value. Gate Time may be varied as a function of the current SigGen index value by assigning a predefined variable (*see Creating Variables, page 21*).

To set Gate Time to a constant or a predefined variable:

- Click **Const** or the desired variable name in the **Gate Time** box or if Gate Time is a constant, enter the value in the **Gate Time** field.

Signal Parameters

Gain



The Gain field of the Edit Signal Segments window is used to adjust the overall level of the digital segment generated by SigGenRZ. All components of this segment are affected by this value.

Note: Gain is measured in dB. The reference level for dB is user defined. See *Calibration* on page 17 for more information.

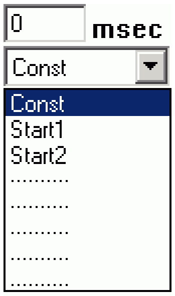
Note: The RZ6 processor limits the Voltage range to ± 10 Volts. Clipping will result during signal play if Voltages exceed this range. If Voltages of the segment, as displayed in the Time Form window of the Edit Signal Segments screen, exceed ± 10 Volts, lower the gain value. You may have to lower the level of one or more of the segment's components to obtain a correct segment level. In contrast, when the Voltage of the segment is too small, signals will not be correctly presented at the D/A converter even if the segment is smoothly displayed on the segment Time Form plot. If this segment is the dominant one in the signal, audible noise may appear. When this happens, you must increase the gain value to suppress noise, and use the on-board attenuator to adjust the signal.

Constant or Variable Gain: By default, Gain is defined as Const, and therefore remains constant throughout the preview process, regardless of the current SigGen index value. Gain may be varied as a function of the current SigGen index value by assigning a predefined variable. See *Variables* on page 18 for more information.

To set Gain to a constant or a predefined variable:

- Click **Const** or the desired variable name in the Gain box or if Gain is a constant, enter the value in the **Gain** field.

Start



The value defined in the Start field defines the delay of the segment onset in the signal in milliseconds.

Constant or Variable Start: By default, Start is defined as Const, and therefore remains constant throughout the preview process, regardless of the current SigGen index value. Start may be varied as a function of the current SigGen index value by assigning a predefined variable (see *Variables*, page 18). You can arrange segments in various manners by assigning different Start values (or variables) for each segment.

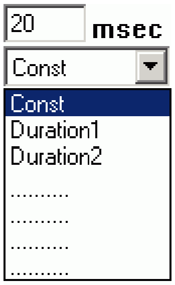
To set Start to a constant or a predefined variable:

- Click **Const** or the desired variable name in the **Start** box or if Start is a constant, enter the value in the **Start** field.

To set the Start value to the beginning of the signal:

- Click the **To Begin** button found in the segment dialog box.

Duration


20 msec

Const

The value of the Duration field defines the total duration of the segment in milliseconds.

Constant or Variable Duration: By default, Duration is defined as Const, and therefore remains constant throughout the preview process, regardless of the current SigGen index value. Duration may be varied as a function of the current SigGen index value by assigning a predefined variable. See *Variables* on page 18 for more information.

To set Duration to a constant or a predefined variable:

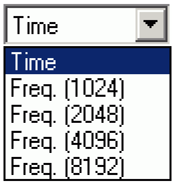
➢ Click **Const** or the desired variable name in the **Duration** box or if Duration is a constant, enter the value in the **Duration** field.

Note: Start and Duration values should not fall outside of the signal duration. If this happens and you do not correct it, SigGenRZ issues a warning, truncates the segment, and continues.

To set the Duration value to the entire length of the signal:

➢ Click the **To Max** button found in the segment dialog box.

Generation Method


Time

Gen. Meth (Generation Method)

You must specify how the segment (its components) is generated by SigGenRZ when you design the signal. SigGenRZ uses two basic generation methods: Time and Frequency. The Time method generates signal components by calculating their time waveform points. Frequency methods specify their frequency characteristics first, and use inverse FFT to obtain the waveform points. Depending on the frequency resolution, there are several Frequency method options.

To choose a Generation Method:

➢ Click on the **Gen. Meth.** Drop-down box and click on the desired **Generation Method** in the **Generation Method** combo box.

When to Use a Frequency Domain Generation Method

Generating signals in the frequency domain allows you to specify the frequency components of your signal. Some signals (tone, click, Gaussian noise, DC shift) can be generated from time domain or frequency domain, while others (band-limited noise, harmonics) can only be generated from frequency domain. Usually when you want to generate band-limited noise, harmonics, or to specify the frequency contents of a signal, you may consider using frequency method.

Choosing Among the Frequency Options

SigGenRZ generates time-domain waveforms from frequency parameters through the use of the inverse Fast Fourier Transform (iFFT). The iFFT builds signals based on a finite number of frequency components (FFT points, which must be a radix-2 number). A list of frequency options is displayed in the Generation Method box.

The number of combined magnitude and phase spectra FFT points is indicated in parentheses. To determine the number of either magnitude or phase spectra FFT points, simply divide this number by 2.

Choice of the number of FFT points depends on the *sample period* you have defined for the time-domain waveform and the desired *binwidth*, or distance between two adjacent points in the signal spectrum in Hz.

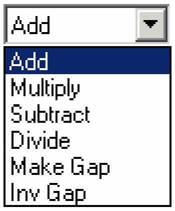
Number of FFT points = 1 / (binwidth x sample period)

To choose a frequency option:

1. Calculate the desired number of FFT points using the above equation.
2. Choose the frequency option with a combined number of magnitude and phase FFT points greater than 2 times the number calculated in (1) to ensure a minimum binwidth. Since there are only a limited number of points for inverse FFT, the signal generated may not be enough to cover the duration of the segment. SigGenRZ repeats the same iFFT result to obtain enough points.

Application Method

App. Meth (Application Method)



Segments may be applied to the existing signal in one of several ways:

- **Add -** A segment may be added to the existing signal.
- **Multiply -** A segment may multiply the existing signal.
- **Subtract -** Subtract a segment from the existing signal.
- **Divide -** Divide the existing signal by the segment.
- **Make Gap -** Creates a gap in the signal, based on the segment.
- **Inv Gap -** Creates an inverse gap in the signal, based on the segment.

Note: The RZ6 D/A converters limit the Voltage range to ± 10 Volts. Clipping will result during signal play if Voltages exceed this range. If this occurs, the overall signal, as displayed in the Signal Window of the SigGenRZ main window, will appear clipped. When this occurs, some modification of segment level or component level will be necessary.

To choose an Application Method:

- Click the desired method from the **Application Method** box.

Using Application Method to Build the Signal

Two Application Method options are designed to help build or modify a signal:

- Add
- Subtract

Add is used primarily to build a signal by adding together two or more segments. For example, a noise segment can be added to a signal to reflect the background noise.

Subtract is used primarily to remove unwanted signal components from the signal. For example, DC offset has to be specified in decibel values, so only positive DC offset can be produced in a segment. A negative DC offset can be generated by subtracting a DC offset from a signal.

Note: Segments consist of one or more components. Components are always additively applied to the segment. As part of component definition, it is necessary to define a component level. When using the Application Methods *Add* or *Subtract*, component level is defined in terms of the system calibration specified during signal parameter definition.

For example, if system calibration is defined as 90 dB equal to 9 Volts, specifying a 90 dB tone will result in a tone whose peak amplitude is 9 Volts.

Using Application Method to Modulate the Signal

Several Application Method options allow you to modulate the current signal:

- Multiply
- Divide
- Make Gap
- Inv Gap

Note: Segments consist of one or more components. As part of component definition, it is necessary to define a component level. When using the modulating Application Methods presented above, component level is NOT defined in terms of the system calibration specified during signal parameter definition, but is instead defined as 0 dB equal to 1 Volt. Thus, specification of a tone with a level of 0 dB will create a tone with a peak amplitude of 1 Volt.

The segment waveform generated will be displayed at the bottom of the Edit Segment Parameters dialog box in both Time Form and Magnitude Spectrum. The waveform can be previewed and updated in this box. See *Working with Segments* on page 59 for detailed information.

Segment Repeating

To support the production of stimuli with multiple pulses, SigGenRZ supports segment repeating. The exact number of repeats and each of the repeats’ temporal position are controlled by three new parameters within the Segment edit dialog box.

The Repeats, RepSep, and Freeze Repeats options are used to control Segment repeating. When Repeats is set to zero or one, a single copy of the generated segment will be applied to the overall signal. Making the Repeats parameter two or more will cause the segment to repeat across the signal with a start-to-start separation of *RepSep* milliseconds between each segment onset.

~

Chapter 5 Components - The Building Blocks of Segments

Components are the foundation of a SigGenRZ signal. Each component is a basic waveform of the signal. It is through the definition of component parameters that the time-varying waveform is defined.

How to Build a Segment - Signal Components

Components are the building blocks of a segment. The manner in which a component is generated is determined by the Generation Method assigned to the segment it is building.

Components may be generated in the following ways:

- Generation Method
- Time
- Frequency
- Parameter specification
- Time-domain parameters
- Frequency-domain parameters

For example, you are building a segment, seg1, composed of two component pure tones. You have chosen a Time as the Generation Method. Both of its components will be defined in the time domain.

How Components are Generated

SigGenRZ generates a time-domain waveform for every component, regardless of which Generation Method, Time or Frequency, has been selected.

Time: When time-domain parameters are specified, SigGenRZ uses these parameters to generate a time-domain waveform. Specifically, SigGenRZ utilizes built-in functions to calculate signal points when time method is used.

Frequency: When frequency-domain parameters are specified, SigGenRZ computes the magnitude spectrum of the signal based on those parameters, generates a phase spectrum, and then converts them into a complex buffer pair (real and imaginary buffer pair). An inverse Fast Fourier Transform (iFFT) will then be applied to the complex buffer pair to generate the time-domain waveform.

Combining Components

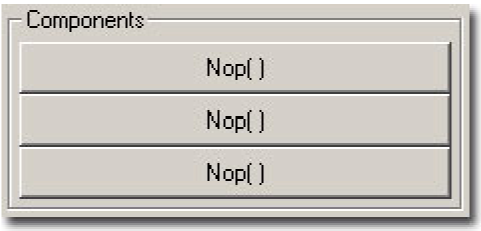
The time-domain component waveforms generated by SigGenRZ are always combined additively.

Note: The RZ6 D/A converters limit the Voltage range to ± 10 Volts. Clipping will result during signal play if Voltages exceed this range. Combining components that are in phase may result in segment Voltages in excess of ± 10 Volts.

Designing Signal Components

Components may be designed from the Component Parameters dialog box. Component Parameter dialog box can be accessed from Edit Segment Parameter dialog box, Component group.

Components



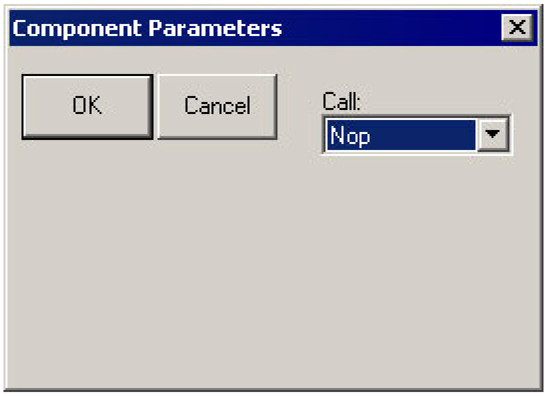
To open the Component Parameters dialog box:

- Click the desired **Component** button.

Component buttons read Nop[ ] (no operation) before a component has been added. After a component has been added, summary information is displayed on the button.

Component Parameters Dialog Box

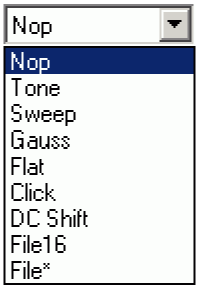
Components of a signal segment are defined in the Component Parameters Dialog Box. Under Call there is a pull-down combo list box which contains all possible signal components SigGenRZ can generate. Clicking on the Drop-down box will display the list. There are different options for Time and Frequency methods.



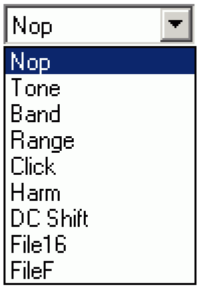
Call

The Call list box displays a list of options you may use in designing your segment. From this list you may choose the type of signal component you wish to define. Depending on which Generation Method you have chosen (Time or Frequency) you will see one of two lists of options:

Time

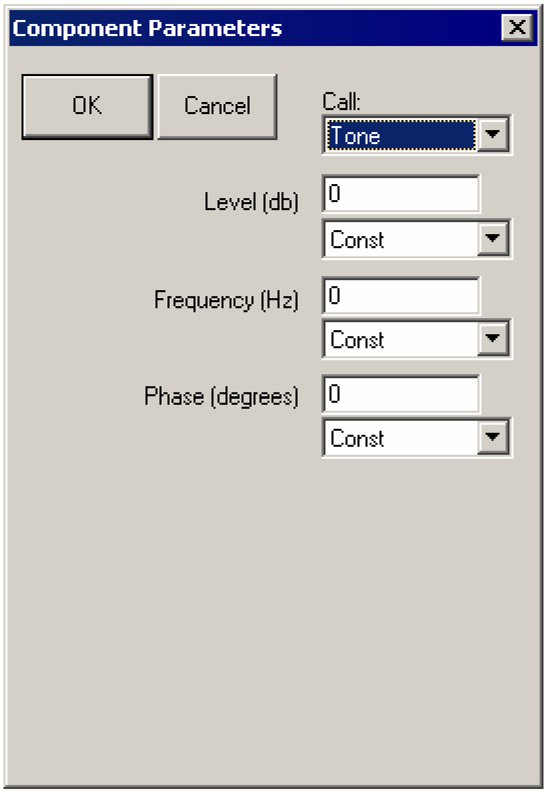


Frequency



Parameter Definition Fields

Once you have chosen the desired type of signal component, corresponding Parameter Definition Fields will be displayed in the dialog box. Each field specifies a parameter required for definition of the component. You must understand and fill the parameter values correctly for SigGenRZ to generate the desired signal.



Call: Tone

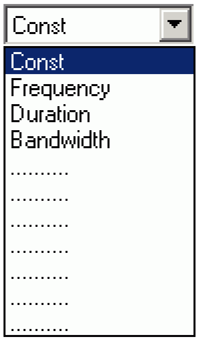
Level (db) 0 Const

Frequency (Hz) 0 Const

Phase (degrees) 0 Const

Each Parameter Definition field is associated with:

- A Value field for entering a constant value
- A Variable box containing the names of all defined variables



To specify a constant value:

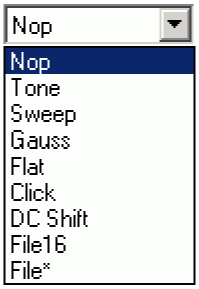
1. Choose **Const** from the **Variables** box (this selection is default).
2. Enter the desired value for the constant in the **Value** field.

To specify a variable value:

- Choose the desired variable from the **Variables** box.

Note: In most cases, component definition requires the specification of a component level. When using the Application Methods, *Add* or *Multiply*, level is defined in terms of the system calibration specified during signal parameter definition. When using the modulating Applications Methods *Multiply*, *Divide*, *Gap*, *Inv Gap*, component levels of this segment are NOT defined in terms of the system calibration specified during signal parameter definition, but is instead defined as 0 dB equal to 1 Volt. For example, specification of a tone with a level of 0 dB will create a tone with a peak amplitude of 1 Volt.

The Time Options



Nop: Causes no operation to be performed, so no component will be defined. Choosing Nop for an existing component causes the component to be removed from the segment.

Tone: Produces a pure tone waveform in the time domain. Parameters include:

- Level in dB
- Frequency in Hz
- Phase (cosine) in degrees

Note: The phase of the Tone component is considered as cosine degrees, that is, a tone with a zero degree phase will begin generating the waveform at the peak of a tone waveform, which is 90 degrees out of phase from a sine waveform.

Sweep: Produces a frequency sweep (a tone with its frequency varying continuously). Parameters include:

- Level in dB
- Starting frequency in Hz
- Sweep rate in Hz/s
- Phase (cosine) in degrees

Sweep rate can be either positive or negative. When sweep rate is positive, the generated signal will have an increasing frequency, or when it is negative, a decreasing frequency. A pure tone will result if the sweep rate is specified to be zero.

Gauss: Produces random noise with a Gaussian distribution. Parameters include:

- Level in dB

Note: The Gaussian distribution has a zero arithmetic mean, and in SigGenRZ its standard deviation is the level specified above. If you generate a Gaussian noise with a specified level, the actual noise peak amplitude will be higher than the specified level.

Flat: Produces random noise with a flat (uniform) distribution. Parameters include:

- Level in dB

Note: The waveform sample points of a flat noise are uniformly distributed between zero and the level specified above.

Click: Produces a two-level (bi-polar) click. There are two peaks (positive and negative) in the bi-level click. Parameters include:

- Level one in dB
- Duration one in ms
- Level two in dB
- Duration two in ms
- Polarity (Enter +1 or -1, other values will affect the signal level.)

Two-level click can be considered as a pair of two opposite polarity clicks. Level one and Duration one specify the level and duration of the first click, and level two and Duration two, the second. When Polarity is a positive number, the first click will be a positive one, the second the negative one, and when Polarity is negative, the first will be negative, and the second positive. A zero Polarity will result in a Nop. The absolute Polarity value will also be used as a scaling factor to be multiplied to the generated component.

For example, a Polarity value of 1.5 will result in a positive-negative click pair, and their peaks are 1.5 times of their levels.

If you want to generate a single-level click, you can specify a zero duration to either first or the second click. Durations must be either positive or zero. A negative duration will be taken as zero.

DC Shift: Produces a DC offset (from 0 Volt). Parameters include:

- Level in dB

Since the Level of DC Shift operation is a dB value, only positive DC shift can be achieved with one DC shift component. To make a negative DC shift, you can generate a positive one first in one segment with desired level, and make another one in a new segment with twice the desired level and subtract it from the signal.

File16: This option allows you to define a component based on one or a group of headerless, 16-bit integer binary files. The file extension **must** be *.000*. When you use only one file, the real file extension must be 000. When you use a group of files, the file extensions must be three-digit numbers (for example, five possible file names: *file.005*, *file.004*, *file.003*, *file.002*, *file.001,* and only *file*.000 should be entered in the parameter field). Parameters include:

- File name (always *filename.000*)
- File extension variable. If there is only one file, Const should be selected. If more than one file will be used, a variable must be selected, and the variable values must be the three-digit numbers of the file extensions.

When a **File16** file is read to SigGenRZ in time domain, the data points will be converted into Voltage values based on 16-bit integer range of values (-32768 to 32767) and the RZ6 D/A hardware range of output levels (±10V).

File*: Allows you to define a component based on one or a group of sound files. The file extension **must** be *.000*. Refer to File16 for using more than one file. In addition to reading raw 16 bit integer and floating point data files, FILE* supports a number of file formats including the sound file format used on PCs, WAV files. In addition, FILE* includes arguments for controlling file seek position and the number of words read from the file, allowing FILE* to excerpt specified portions from files under variable control.

Finally, FILE* supports a versatile file-name access convention whereby the accessed file name is built using any combination of user defined text and a variable value. FILE*'s most powerful feature may be the indirect-file-access control whereby the named file contains a list of files names that are indexed using a specified variable to select the actual data file name.

Parameters include:

- File Type
- File Type variable
- Seek Position

- Seek Position variable
- Excerpt Size
- Excerpt Size variable
- Indirect File Offset
- Indirect File Offset variable
- File name (always *filename.000*)
- File extension variable. If there is only one file, Const should be selected. If more than one file will be used, a variable must be selected, and the variable values must be the three-digit numbers of the file extensions.

File Types

By default, FILE* will load raw floating-point data without offset. When FILE* is selected, the first argument becomes the File Type to be accessed. This parameter can be specified with a variable, but care should be taken that the variable generates legal values that match the actual file type. It is possible to 'hang' the program by mismatching the file type specified with the actual file being accessed. The currently supported file types are:

Raw Float	index = 0
Raw Int16	index = 1
ASCII	index = 2
WAV (Mix)	index = 3
WAV (Left)	index = 4
WAV (Right)	index = 5

Raw Float: Data samples are organized in a sequential floating point format using standard IEEE 32-bit (four byte) data. Numbers should be scaled between +/- 10.0 corresponding to the +/- 10 Volt peak to peak output of the RZ6 D/A converters.

Raw Int16: Data organized as a raw file of 16 bit integers stored in MSB first format. This component call is compatible with the INT16 file format supported by SigGenRZ and includes the other features of FILE* like indexing etc. Data should be scaled between +32767 and –32768 which will be converted to +/- 10 Volts when loaded.

ASCII: This is a standard text list of floating point numbers. These lists can be generated with EXCEL or MATLAB. Data should be scaled between +/- 10.0 as with Raw Float. It should be noted ASCII data files may be quite large with a single data value taking over 20 bytes of disk space.

WAV Files: SigGenRZ supports the Windows WAV file format. While WAV files can be either 8- or 16-bit, SigGenRZ only supports 16-bit files and will fail to load if an 8-bit file is selected. A number of free-ware Windows programs are available for editing and formatting WAV files. These programs typically allow the user to get the file into the proper 16 bit format with the needed sample rate.

Note: the FILE* loader will NOT convert the sampling rate of the loaded file to match that specified within SigGenRZ. It is the user's responsibility to ensure the WAV files being used have the appropriate sample rate.

WAV (mix), WAV (left) and WAV (right): Mono and Stereo WAV files can be read. When mono files are used, any of the WAV component calls will load the same data. If stereo files are used, use the mix, left and right options to load the needed channel. WAV (mix) sums the left and right data and scales down by 6dB.

Seek Position

Seek position and excerpt sizes are used to select the portion of the sound file you wish to load. The SEEK_POS parameter can range from 0 to files_size – points-to-read. Seeking beyond the end of the file will cause garbage to load. It is the user’s responsibility to limit file access to the files size. The SEEK_POS parameter is useful for accessing random segments from a pre-computed noise file. Often masking noise is needed with specific frequency characteristics. A large noise file can be pre-computed and saved to disk. The SEEK_POS argument can then be fed with a RANDOMIZED variable accessing random noise segments generating a pseudo-random noise effect.

Excerpt Size

The EXCERPT SIZE parameter is used to select the number of points to be read from the file. Normally (EXCERPT SIZE = 0) FILE* will read enough points to fill the target segment. That is a 100 ms segment sampled at 50 kHz will load 5000 points. The EXCERPT SIZE parameter can be used to override this behavior, by limiting the number of points read to something less than the 5000 needed to fill the segment. This argument is very useful for reading sound segments from raw data files with variable lengths. Also if one wishes to take excerpts from a larger file, this parameter is useful for controlling the number of points read into the segment.

File Name

FILE* supports a versatile file naming/lookup scheme, allowing the output of a variable to be inserted at any point within a text field specifying the file name. The examples below illustrate how the file name parameter functions.

Simple fixed file access.

File Name	=	“sound.wav”
SG Variable	=	Const
File Accessed	=	‘sound.wav’

*** Wildcard file access Example 1:** Will access file with the name built using the text specified and the current value of the variable named QSPEC.

File Name	=	“snd***.wav”
SG Variable	=	QSPEC = 247.0
File Accessed	=	‘snd247.wav’

*** Wildcard file access Example 2:** Builds file name as specified and truncates the number to the integer portion.

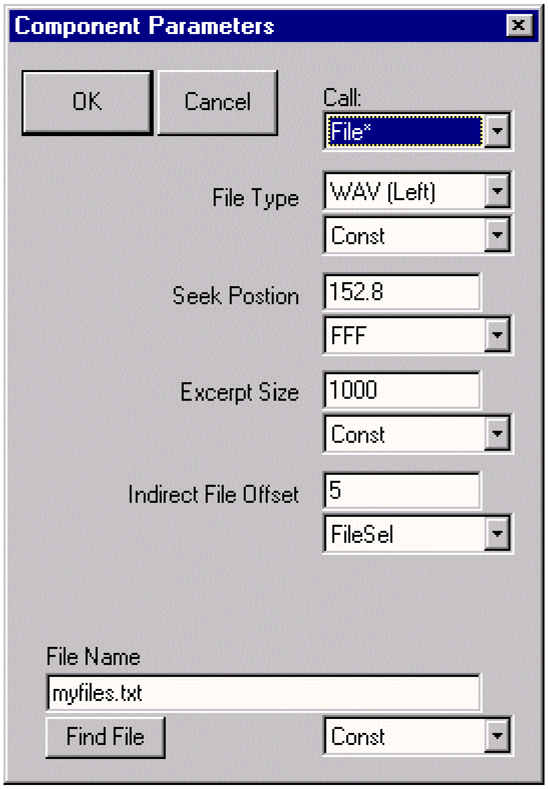
File Name	=	“sound.g**”
SG Variable	=	QSPEC = 12.8
File Accessed	=	‘sound.g12’

*** Wildcard file access Example 3:** Builds file name using first three digits of current value of QSPEC.

File Name	=	“snd***zz.i16”
SG Variable	=	QSPEC = 5437.0
File Accessed	=	‘snd543.i16’

Indirect File Offset

When this parameter is made non-zero, indirect file accessing is enabled. This feature supports indirect file accessing whereby the File Name parameter (described above) accesses a text file containing a list of sound file names each on its own text line. Each line of the file must have the exact same number of characters and the first line is not used (index=0). The value of the Indirect File Offset parameter is used to index to a specified line and get the actual sound file name. Consider the following example.



Call: File*

File Type: WAV (Left), Const

Seek Postion: 152.8, FFF

Excerpt Size: 1000, Const

Indirect File Offset: 5, FileSel

File Name: myfiles.txt, Const

Myfiles.txt contains the following text:

1234567.123

f1k_cat.wav

f1k_dog.wav

f1k_mou.wav

f1k_frg.wav

f1k_bat.wav

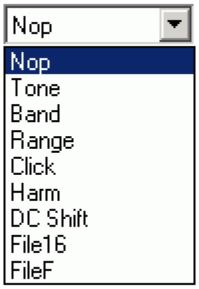
f1k_zea.wav

The WAV file accessed will be f1k_bat.wav. 1000 points will be read starting at point 152 and if the file is stereo, left channel data will be returned. Note that the first line is not used, and all lines have the same number of characters.

File16 and **File*** are very useful ways to generate waveform components which cannot be easily produced with simple mathematical functions. Combined with SigGenRZ variables, you can assign several binary files to the same component, and use a certain file for a certain interval based on the variable values.

When a **File*** file is read to SigGenRZ in time domain, the values of the data points will be taken in terms of Volts and no conversion will be performed.

The Frequency Options



Note: Components generated from Frequency Method are periodic waveforms. Each "period" of the waveform includes the same number of points of inverse FFT (specified as Freq[####] in Edit Segment Parameters dialog box). One "period" of the generated tone has an integer number of complete cycles. So, generated noises may appear periodic due to this inverse FFT technique. You may want to select a different number of iFFT points to produce a satisfactory signal.

Nop: Causes no operation to be performed, so no component will be defined. Choosing Nop for an existing component causes the component to be removed from the segment.

Tone: Produces a pure tone waveform from frequency domain. Parameters include:

- Level in dB
- Frequency in Hz
- Phase (cosine) in radians

Band: Produces a noise of a specific bandwidth around a center frequency. Parameters include:

- Low frequency level in dB: The lower end level of the band. This level is per Hz average level.
- High frequency level in dB: The higher end level of the band. This level is per Hz average level.
- Center frequency in Hz: The geometric center frequency. The high and low frequencies are calculated based on following:

$$F_H = F_C \times 2^{\ BW/2}$$

$$F_L = F_C \times 2^{\ -BW/2}$$

Where: F_H is the high frequency, F_L the low frequency, F_C the center frequency, and BW the bandwidth in Octaves.

- Bandwidth in Octaves: One Octave is a doubling (or halving) usually in frequency measurements. For example, one Octave from 200 Hz is 400 Hz, while two Octaves is 800 Hz, and so forth. This value must be greater than zero. A zero will produce a tone, and a negative value will generate an error message.
- Roll-off in dB/Octave: The roll-offs at the band boundaries. A negative value specifies a roll-off, while a zero or positive value sets the roll-off to be infinite.

Range: Produces a noise of a specific frequency range. Parameters include:

- Low frequency level in dB: Magnitude spectral level at the lower end of the range.
- High frequency level in dB: Magnitude spectral level at the higher end of the range. Levels at other frequencies will be linearly interpolated.
- Low frequency in Hz
- High frequency in Hz
- Roll-off in dB/Octave (see Band)

Click: Produces a bi-level click from frequency domain. Parameters include:

- Low frequency level in dB
- High frequency level in dB
- Low frequency in Hz
- High frequency in Hz
- Roll-off in dB/Octave

Harm: Produces a spectrum composed of a fundamental frequency and several of its harmonics. Parameters include:

- Fundamental frequency level in dB
- Fundamental frequency in Hz
- Frequency separation in Hz: Distance between the frequency components.
- Roll-off in dB/component: The magnitude reduction rate in dB per component. A positive number indicates an increase in magnitude.
- Number of components: Unity means only the fundamental component (tone); zero indicates infinite number of harmonic components. A negative number is invalid, and may crash your system.

DC Shift: Produces a DC offset in frequency domain. Parameters include:

- Level in dB: Since a DC-offset in spectrum is the same level of DC offset in time signal, this level is both time and frequency domain DC offset level. (See *The Time Options,* page 50 if you want to generate a negative DC Shift)

File 16 and FileF: These options allow you to define a component based on one or a group of header less, 16-bit integer binary files or binary floating point files. When you use a group of files, the file extensions must be three-digit numbers (for example, five possible file names: *file.005, file.004, file.003, file.002, file.001,* and only *file.*000 should be entered in the parameter field). Parameters include:

- File name (*filename.000 for more than one file*)
- File extension variable. If there is only one file, Const should be selected. If more than one file will be used, a variable must be selected, and the variable values must be the three-digit numbers of the file extensions.

The frequency and phase data should be **interleaved** (freq, phase, freq, phase …). The level of each frequency bin is specified in dBVolts and the phase in degrees (0 to 360).

The values of the frequency bins should be set to something like -250 if you don't want sound, and 0 if you want sound (for example, a 0 in one bin would produce a 1 V tone). It is important that you have the same number of frequency points and phase points as you have points in your FFT. For example, if you use a 1024-pt FFT, your file should contain 2048 points (1024 freq points, and 1024 phase points). If it is too short, SigGenRZ will include a bunch of zeroes at the end, which results in a large unwanted signal. If you have too many points, SigGenRZ will use the number of points of the specified FFT, and ignore the remaining values. Note, this will ignore the calibration you set in SigGenRZ (e.g. 9 V = 180dB).

Note: Entering 0 for Roll-off in dB/Octave will result in the computation of infinite roll-off.

Note: In both time domain and frequency domain, data points in **File16** and **FileF** files must have the same sampling period as the signal you are designing.

Importing Components

SigGenRZ signals can be created using existing non-SigGenRZ digital signals. SigGenRZ signals are built from segments. Segments are built from components. A non-SigGenRZ signal, therefore, must be imported as a component.

Imported waveforms may be specified in either the time or frequency domains. Refer to the above sections for designating file extensions.

Importing Time-Domain Waveforms

Time-domain waveforms may be imported for a variety of reasons.

- Including the waveform as one segment of a multi-segment signal.
- Adding the waveform to another component or segment.
- Multiplying another segment by the imported waveform.

To import a time-domain component:

1. If you want to import more than one file, define a variable which will have values equal to the file extension digits while SGI is increasing.
2. Access the **Edit Signal Segments** window.
3. Choose **Time** from the **Generation Method** list box.
4. Click one of the three **Component** buttons to access the **Components Parameters** dialog box.
5. Choose the desired **File** command from the **Call** Drop-down box.

To import a 16-bit integer binary file:

- Choose **File16**.

To import another file type:

1. Choose **File***.
2. Type in the file name in the **File** field. The file extension must be *.000*.
3. If you want to import more than one file, select the file extension variable in the variable list.
4. Click the **OK** button.

Importing Frequency-domain Waveforms

You may wish to import a component defined in the frequency domain. An imported frequency domain waveform may be used to:

- Generate a time-domain component waveform with spectral characteristics that remain constant over time.
- Apply spectral shaping to other SigGenRZ components.

To import a frequency-domain component:

1. If you want to import more than one file, define a variable which will have values equal to the file extension digits while SGI is increasing.
2. Access the **Edit Signal Segments** window.
3. Choose **Time** from the **Generation Method** list box.

4. Click one of the three **Component** buttons to access the **Components Parameters** dialog box.
5. Choose the desired **File** command from the **Call** Drop-down box.

To import a 16-bit integer binary file:

- Choose **File16**.

To import a floating point binary file:

1. Choose **FileF**.
2. Type in the file name in the **File** field. The file extension must be *.000*.
3. If you want to import more than one file, select the file extension variable in the variable list.
4. Click the **OK** button.

Chapter 6 Working with Segments

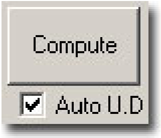
After you have defined component parameters and created a segment, you may:

- Modify segment parameters.
- Obtain information about the segment's time-domain waveform.
- Obtain information about the segment’s frequency spectrum.

You may do these things from the Edit Signal Segments window.

Computing Segments

As you modify segment parameters, you affect the characteristics of the segment waveform. SigGenRZ does not automatically regenerate the waveform every time you change a parameter. A recalculation and updating will be performed when you click OK to finish editing segments. It may be necessary to recalculate the segment while you modify the parameters. This can be done at various points in the modification process.



Manual Update

You may elect to re-compute segments manually. This allows you to edit segment parameters without recalculating the segment.

Compute

Clicking the Compute button recalculates the segment based on the currently specified parameters.

Auto Update

You may wish to have SigGenRZ automatically recalculate segments when a parameter value is changed.

Auto U.D.

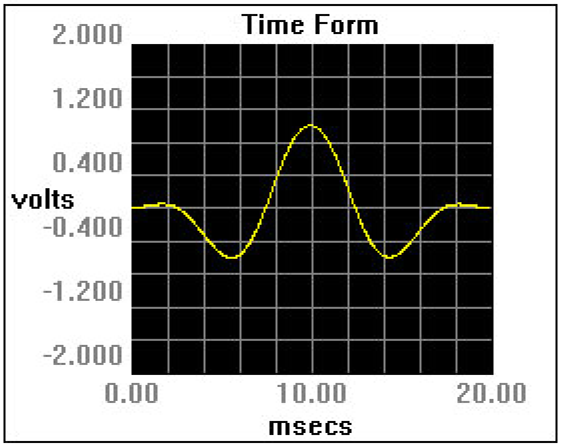
When Auto Update is checked, the segment is automatically recalculated whenever you move the focus from a field that has been updated.

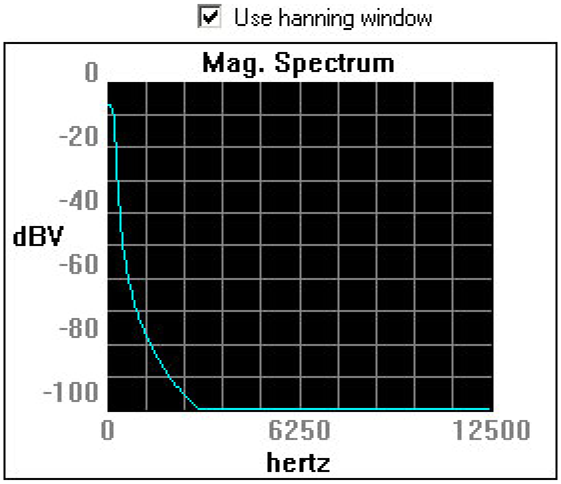
The following are examples of situations where recalculation would occur:

- You edit a parameter and then TAB to another field.
- You edit a parameter and then click in another field.
- You select a new parameter from a list.

Displaying Segments

After you have created a segment, SigGenRZ displays the time-domain waveform in the Time Form window. The magnitude spectrum of the segment is displayed in the Magnitude Spectrum window.




☑ Use hanning window

The Time Form Window

The Time Form window displays the time-varying waveform of the current segment. The x-axis is measured in milliseconds. The y-axis is measured in Volts.

Default Display

X-Axis: By default, values of the x-axis extend from 0 to the number of milliseconds defined in the Duration field of the Edit Signal Segments window. You may use cursor to zoom-in to display details.

Y-Axis: SigGenRZ automatically scales the y-axis so that the peak Voltage of the signal is displayed. The x-axis is always centered at the y-axis (0 Volt horizontal line). The y-axis is always placed at the left boundary of the time plot.

The Magnitude Spectrum Window

The Magnitude Spectrum window displays the frequency domain values of the segment. The x-axis is measured in Hertz. The y-axis is measured in dBV (decibels in Volts). The reference value for dBV is user defined. See *Calibration* on page17 for more information.

Default Display

X-Axis: The x-axis extends from 0 Hz to the Nyquist frequency based on the current sample period.

Nyquist frequency (Hz) = 1/sample period (seconds) x 1/2

Y-Axis: SigGenRZ automatically scales the y-axis so that the maximum value in dBV can be displayed clearly.

Using a Hanning Window

You may apply a Hanning window to the frequency domain signal by checking the Use Hanning Window box.

Using Cursors

You may wish to focus on a portion of either the Time Form window or the Magnitude Spectrum window. SigGenRZ provides two types of cursor functions that you can use to zero-in on specific areas of the display.

Zoom-In

You can zoom-in on a section of either the Time Form window or the Magnitude Spectrum window.

To zoom in:

1. Place the mouse at the right or left edge of the desired area of either the **Time Form** window or the **Magnitude Spectrum** window.
2. Hold down the left mouse button.
3. Drag the mouse over the desired area.
4. Release the left mouse button.

The window will now display the desired section.

Note: When you zoom-in on a Time Form window, the Magnitude Spectrum window is automatically recalculated to reflect the new spectrum of the zoomed-in section.

To return to the default display:

- ➢ Click the right mouse button anywhere in the display window.

Crawling

You can "crawl" through a zoomed-in time-domain waveform. Crawling will scroll the display through the waveform.

To crawl to the right:

- ➢ Click the left mouse button in the right portion of the display.

To crawl to the left:

- ➢ Click the left mouse button in the left portion of the display.

Note: The magnitude spectrum will be automatically recalculated to reflect the energy represented in the portion of the signal that is displayed in the window.

Placing a Cursor

You can view information about a specific point in either the Time Form window or the Magnitude Spectrum window by placing a cursor within the desired display. The information will be displayed in the Segment Time Form, described on the next page.

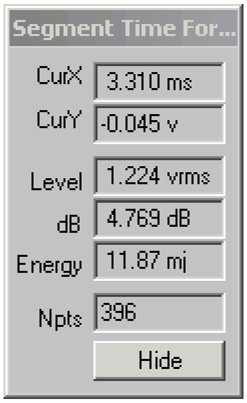
To place a cursor:

Place the mouse pointer in the desired display window.

1. Hold down the **Shift** key.
2. Hold down the left mouse button.
3. Drag the cursor to the desired location.
4. Release the **Shift** key and left mouse button.

The Segment Time Form

The Segment Time Form displays information about the current segment or portion of a segment and may display information about either the Time Form window or the Magnitude Spectrum Window.



Segment Time For...

CurX 3.310 ms
CurY -0.045 v
Level 1.224 vrms
dB 4.769 dB
Energy 11.87 mj
Npts 396

CurX

CurX is the x-axis (time) value at the cursor, in milliseconds (ms).

CurY

CurY is the current y-axis value of the signal at the point of any cursor, in Volts.

Level

Level is the root mean squared (RMS) value of the signal as measured in Volts.

dB

dB is the level in dB Volts of the currently displayed signal.

Energy

Energy is the total energy of the displayed signal, in millijoules (mJ).

Npts

Npts is the number of sampling points contained in the displayed time signal waveform.

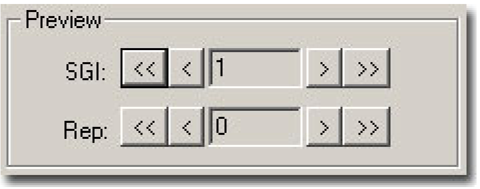
Hide

Clicking the Hide button causes the Segment Time Form to be hidden. You may display it again by selecting Cursor Box from the View menu of the main window.

Previewing Segment Intervals

Individual segment can be previewed in the Time Form and Magnitude Spectrum windows of the Edit Signal Segments window. Using SGI you can preview the segment in any interval of the signal.

Specifying the SigGen index



SGI: 1

Rep: 0

SigGen index

The SigGen index (SGI) field specifies the specific interval you wish to preview. Since variable SGI is defined in Edit Signal Parameters window, boundary control is not applied to the SigGen index here. In other words, you can increment SGI beyond its actual value. In this case, the last variable values will be used to compute the waveform.

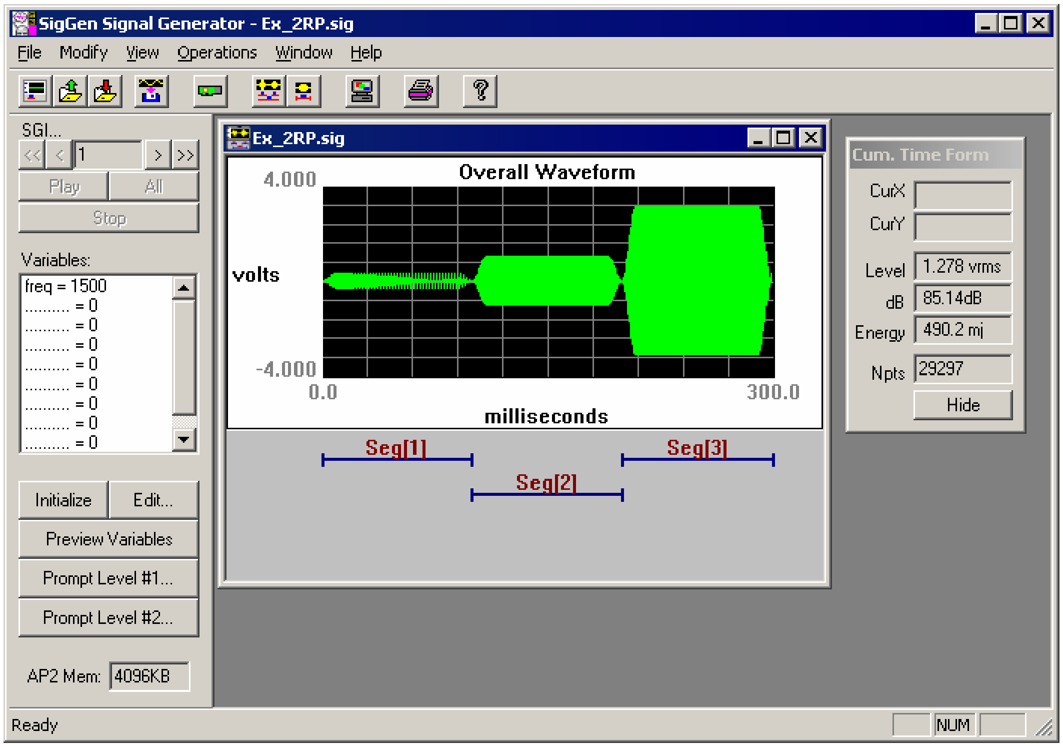
< >	Lowers/raises the trigger tick in increments of 1.
<< >>	Lowers/raises the trigger tick in increments of 10.

Note: The Rep. Field is not supported at this time.

~

Chapter 7 Working with the Complete Signal

Once you have defined all signal-level, segment-level, and component-level parameters, you can view the final waveform and preview the signal.



SGI... 1

Variables:
freq = 1500
.... = 0
.... = 0
.... = 0
.... = 0
.... = 0
.... = 0
.... = 0
.... = 0

AP2 Mem: 4096KB

Level 1.278 vrms
dB 85.14dB
Energy 490.2 mj
Npts 29297

Displaying the Signal

The overall signal is displayed in the Signal window.

To view the overall signal:

- Return to the main SigGenRZ window.

When you click OK to return to the current Signal window from either Signal Parameters or Edit Signal Segments window, SigGenRZ will automatically update all parameters and the overall signal waveform. The Signal window displays the time-varying waveform of the current segment and the x-axis is measured in milliseconds. The y-axis is measured in Volts. If you have defined/modified a variable file, you may click on the Initialize button to read in all variable values and recalculate the waveform.

Default Display

X-Axis: By default, values of the x-axis extend from 0 to the number of milliseconds defined in the Duration field of the Signal Parameters window.

Y-Axis: SigGenRZ automatically scales the y-axis so that the peak Voltage of the signal is displayed. The origin of the y-axis is always 0 Volts.

Note: The RZ6 D/A converters limit the Voltage range to ± 10 Volts. The Signal window limits the displayed Voltage to this range. If there are points of which the calculated values are greater than 10 Volts, or smaller than -10 Volts, you will spot clipping from this display. When this happens, if you are concerned, you may change either the segment levels or component levels.

Using Cursors

It may be desirable to look at the details of a small signal portion, or obtain information on specific point. You can zoom-in on a portion of the overall signal or place a cursor on a specific point in the overall signal.

Zoom-In

To zoom in:

1. Place the mouse at the right or left edge of the desired area of the Signal window.
2. Hold down the left mouse button.
3. Drag the mouse over the desired area.
4. Release the left mouse button.

The window will now display the desired section.

To return to the default display:

- Click the right mouse button anywhere in the Signal window.

Crawling

You can "crawl" through a zoomed-in time-domain waveform. Crawling will scroll the display through the waveform.

To crawl to the right:

- Click the left mouse button in the right portion of the display window.

To crawl to the left:

- Click the left mouse button in the left portion of the display window.

Placing a Cursor

You can obtain information about a specific point in the Signal window by placing a cursor with the desired display. This information is displayed in the Cumulative Time Form, as described on the next page.

To place a cursor:

1. Place the mouse in the desired display window.
2. Hold down the Shift key.
3. Hold down the left mouse button.
4. Drag the cursor to the desired location.
5. Release the Shift key and left mouse button.

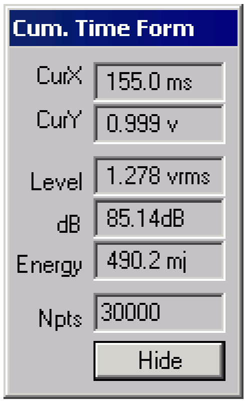
The Cumulative Time Form

The Cumulative Time Form (or Cursor Box) displays information about the current signal or portion of a signal. It is displayed by default when SigGenRZ opens, but may be hidden at any time.

When there is no cursor in the signal waveform, CurX and CurY in the Cumulative Time Form will be empty.

To display the Cumulative Time Form:

- Select **Cursor Box** from the **View** menu.



CurX 155.0 ms
CurY 0.999 v
Level 1.278 vrms
dB 85.14dB
Energy 490.2 mj
Npts 30000

CurX

SigGenRZ allows you to place a cursor within the displayed signal. CurX is the x-axis (time) value of the cursor, in milliseconds (ms)

CurY

CurY is the current y-axis value at the cursor, in Volts.

Level

Level is the root mean squared (RMS) value of the displayed signal measured in Volts.

dB

dB is the average level, in dB Volts, of the currently displayed signal.

Energy

Energy is the total energy of the displayed signal, in millijoules (mJ).

Npts

Npts is the number of sampling points contained in the displayed signal waveform.

Hide

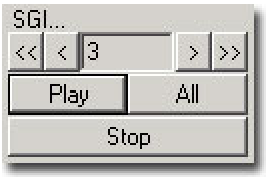
Clicking the Hide button causes the Segment Time Form to be hidden. You may display it again by selecting Cursor Box from the View menu of the main window. You can also hide the Cumulative Time Form by clicking Cursor Box on the View menu.

Previewing the Signal

While SigGenRZ can design signals for a variety of purposes, many users will wish to design stimulus signals for experiments. Often, these experiments will require the repeated presentation of the stimulus signal through multiple intervals. Signal parameters may be varied from one interval to another during the experiment. SigGenRZ provides a means for previewing (listening to) the signal through all intervals, or SigGen indexes (SGIs).

The SigGen Index

By setting the SigGen index (SGI), you can play one selected interval of your signal or play all intervals of your signal.



Note: When SigGenRZ is started and it cannot find the RZ6 device (the hardware may not be turned on), it will still run but the above three buttons will be dimmed and not be available for signal preview.

You can preview a specific interval, *n*, of the signal by setting the value of the SigGen index equal to *n*. This is most useful if at least one signal parameter will vary from one interval to another. Boundary control applies to this SGI display, so if your variables are limited by a boundary, you can only change SGI in the range to view the signal.

< >	Lowers/raises the SigGen Index in increments of 1.
<< >>	Lowers/raises the SigGen Index in increments of 10.

Previewing

Play

Clicking the Play button plays interval *n* of the signal where *n* is the number specified in the SigGen index field (the currently displayed signal).

All

Clicking the All button plays all specified intervals of the displayed signal. When All is clicked, SigGenRZ will initialize all variables, recalculate the entire waveform, and present every signal interval as SGI increments. A next interval is calculated when the previous interval is played, so the signal intervals will not be presented continuously.

The delay between intervals is determined by the time SigGenRZ takes to calculate the next interval.

Stop

Clicking the Stop button stops the signal preview (either the Play or play All mode).

~

Part 2 Illustrative Examples

~

Chapter 8 General Examples

In future releases this chapter will provide examples to illustrate the use of a variety of SigGenRZ features.