Welcome to Z-Tool

Z-Tool is a development application which can be used to communicate with TI ZigBee target device development system. Communication takes place using a standard RS-232 serial communication port on a Windows based PC. It is used for development, testing, and debugging purposes.

What's new in this release

There are many new exciting features in this Z-Tool 2.0 release. We have completely redesigned Z-Tool and have it built on top of Z-Architect SDK. This allows us to make Z-Tool 2.0 much more robust and easy to use. Yet, we managed to keep the user interface the same as many of the users may have already become acquainted with it. Z-Tool 2.0 includes the following new features.

Ability to connect to multiple devices simultaneously property scripting engine property.

Message panel enhancements property supports property to connect to a device via a socket connection property property

System requirements

Hardware

The minimum hardware requirements are:

CPU: Pentium III or later

Memory: 128MB or greater

Available disk space: 60MB

Development boards and RS-ZigBee devices:

232 cables

Software

The minimum software requirements are:

Windows XP with Service Pack Operating System:

Microsoft .NET

Version 2.0 Framework

Installing Z-Tool

Here are installation instructions:

- 1. Close all programs.
- 2. Unzip the installation package
- 3. Run Setup.exe
- 4. Follow the instructions on the screen.

By default, Z-Tool is installed in **C:\Program Files\Texas Instruments\Z-Tool**.

Basic features of Z-Tool

Key features of Z-Tool include the following:

- **Easy to use Message and Log panels** Send and receive messages to multiple devices simultaneously. Enumeration fields are displayed properly.
- **☑ Easy to use Script Window** A variety of features keep your scripts organized, improve your productivity, and save time.
- **™ Configuration Window** You can now read and write configuration parameters on devices using Configuration Window.
- Support for Socket Connection You can now connect to a real or virtual ZigBee device via a socket connection.
- Support for both old and new TI LPW Serial Port Interfaces
 Automatically detects devices with old and new interfaces.

Basic steps to use Z-Tool

These are the basic steps to begin using Z-Tool

- 1. Run Z-Tool
- 2. Open Settings and set running environment and preferences
- 3. Save Settings and exit Z-Tool
- 4. Connect devices
- 5. Run Z-Tool.
 - Click "Scan for Devices" if not all devices were detected on initial auto scan
 - Click "Scan for Devices" when new devices are attached to the system
- 6. You can now send/receive messages, run scripts, and run Configuration on the devices.

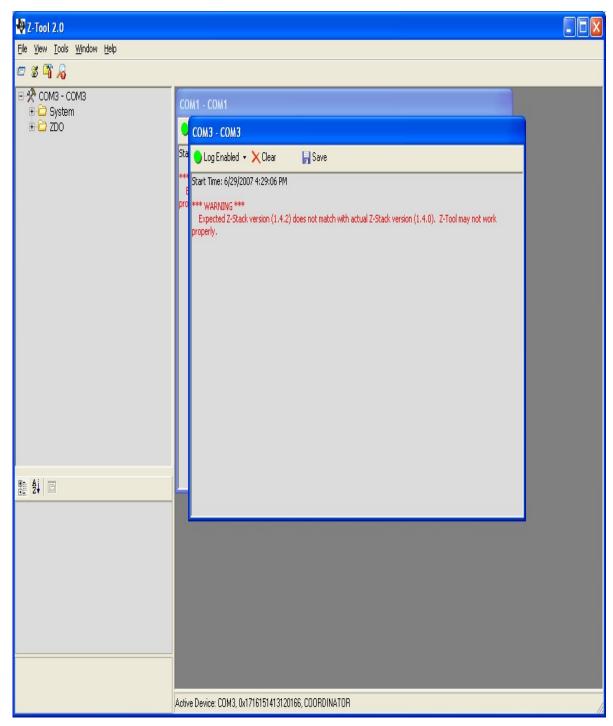
Steps 1 to 3 are only required when Z-Tool settings need to be changed.

Starting Z-Tool

Once installation is complete, you will see a Z-Tool folder under Start menu.

To start Z-Tool

• From the **Start** menu, select **Programs -> Texas Instruments -> Z- Tool.**

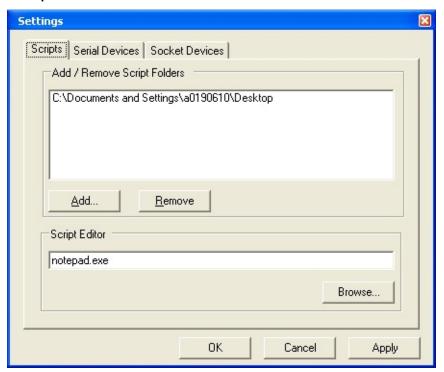


Copyright © 2007 Texas Instruments Inc. All rights reserved.

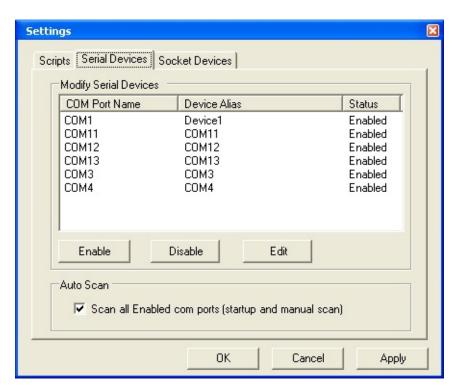
Settings in Z-Tool

Before using Z-Tool, it's recommended to run Settings and set your preferences and operating environments. To open Settings dialog box, select Settings under Tools from the MenuBar. Alternatively, you can click on Settings button on the ToolBar. The available settings are:

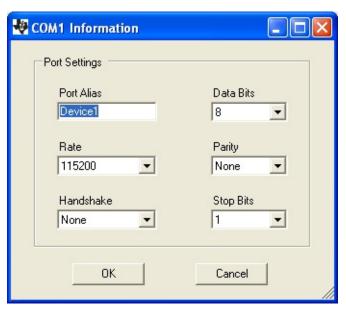
1. Scripts settings. Use this tab to add script folders and set the default Script editor.



2. Serial Devices settings. Use this tab to set COM Port information in Device Aliases. Select Auto Scan checkbox. Disable any COM Ports that are used for other applications. Ex: COM1 usually used by PDA sync. applications.

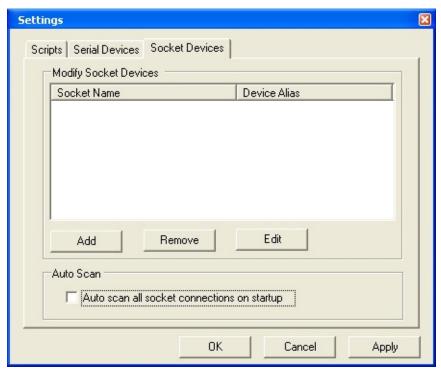


Double click on the desired COM Port to open COM Port setting dialog box. Set alias name, Data Bits, Baud Rate, Parity Bit, Handshake, and Stop Bits for each COM Port. Use the default if you are not sure what to enter.



3. Socket Devices settings. Select Auto Scan if you want Z-Tool to scan for socket devices. Use this tab to add socket connections if

you have any.

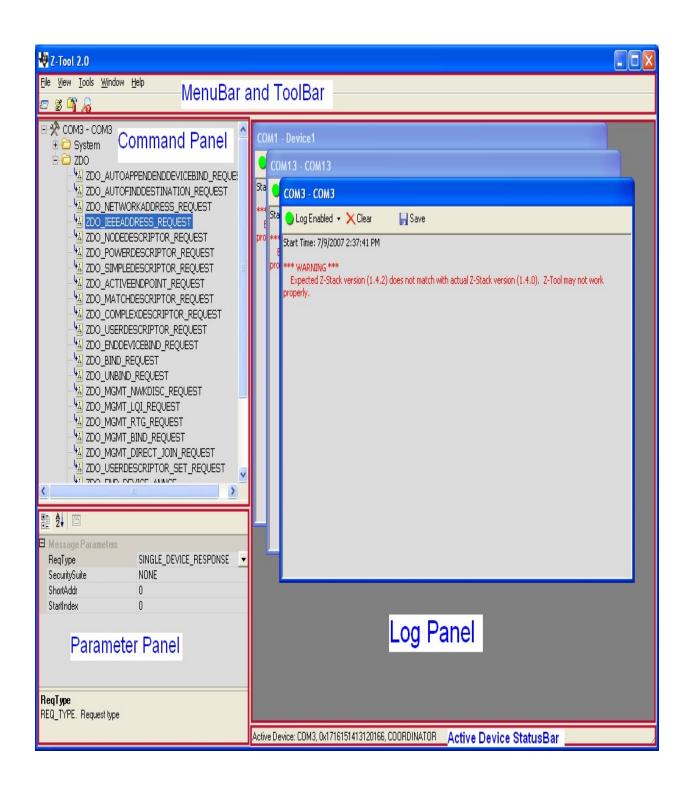


Enter socket information.



Moving around in Z-Tool

- Z-Tool consists of these main areas:
 - MenuBar and ToolBar contains tool and menu options
 - Command Panel contains all messages supported by the associated device
 - Parameter Panel contains all parameters associated with the selected message
 - Log Panel contains all log messages
 - Active Device StatusBar contains information of the active device



Copyright © 2007 Texas Instruments Inc. All rights reserved.

Scanning for Devices

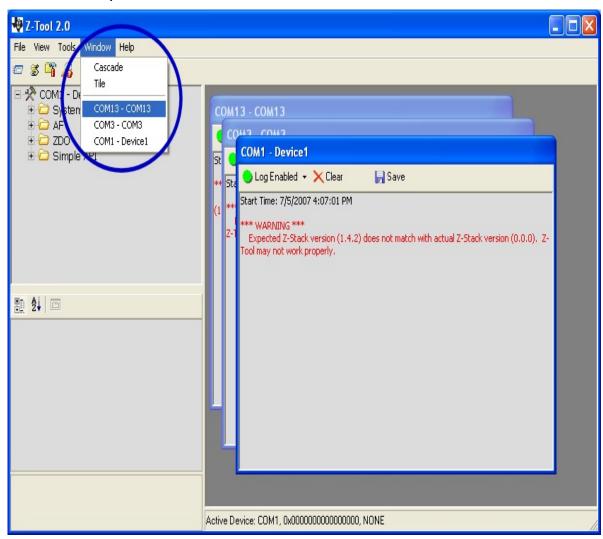
Though scanning for devices is very straightforward, it's good to understand how Z-Tool performs scanning. The followings are the expected behaviors:

- The list of all connections, serial and socket, are generated at the beginning of a scan. Z-Tool gathers the options selected under Settings window. If Auto Scan option is selected in Serial Devices tab then all Enabled serial connections are added to the list. Settings of each serial connection are also read at this time. If Auto Scan option is selected in Socket Device tab then all socket connections are added to the list.
- Z-Tool then sends out SYS_PING and SYS_GET_DEVICE_INFO commands to all of those connections. Notice that it also re-scans the devices that are already detected. A device must response to both of those commands to get detected. There are total of 4 retries and 3-second timeout. There's also a backup 10-second timeout to counter problems with system's resources running low. Scanning can be as fast as under 1 second if all devices respond immediately.
- Scanning for socket devices usually takes longer than scanning for COM Port devices. If you don't have any socket connections, it's recommended to deselect Auto Scan option under Socket Devices.
- Z-Tool automatically creates 1 log for each detected device. If you don't see the log for your device, chances are the device didn't get detected.
- Z-Tool also clears all logs after scanning completes. If you need to save your logs, be sure to save them before doing a scan.
- Scanning is performed automatically when Z-Tool first starts up.
- There are two ways to perform a scan
 - 1. Press Scan Devices button on the ToolBar or
 - 2. Click on Tools -> Scan for Devices on the MenuBar

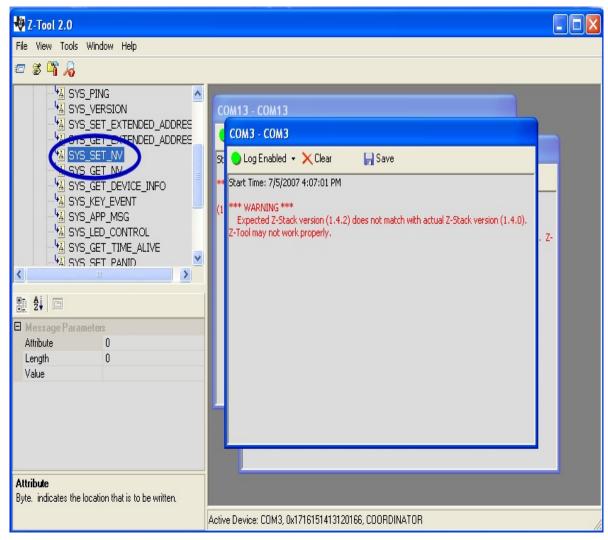
Sending Commands

To send a command to a device, simply follow these steps:

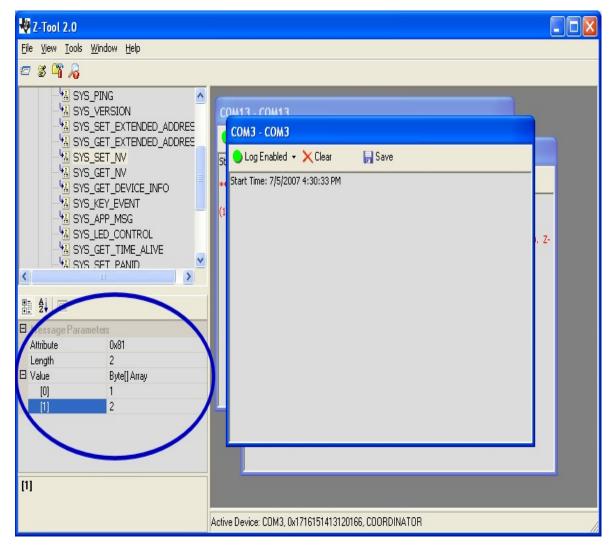
1. Select the device by clicking on the device's log panel or from Window dropdown menu



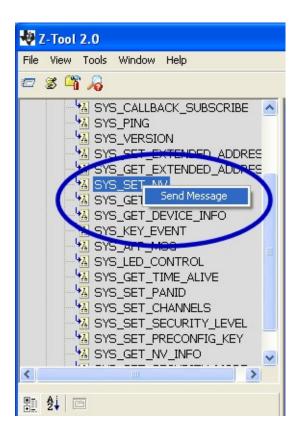
2. From the message browser, select the message to be sent



3. Enter parameters' values



Right click on the message and select Send Message



Working with Configuration Window

Configuration window allows user to work with configuration values. It allows reading, modifying and writing parameter values on a device. User can also open and save the values to files. There are two types of configuration parameter: Device Specific and Network Specific. Device Specific parameters are parameters that are only associated to a device. Network Specific parameters are parameters that are associated to the network. Network Specific parameters should be set to the same values for all devices across the network.

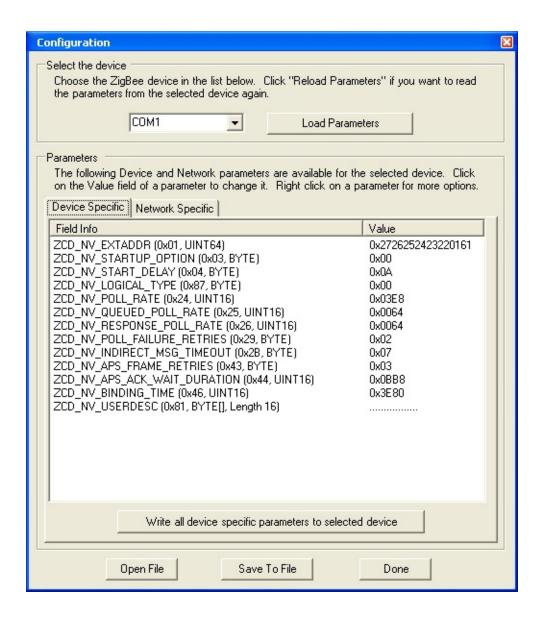
Working with Scripts

Script Browser and Control can be opened by selecting Script Window under Tools drop down menu. Alternatively, you can click on Script Window button on the ToolBar.

- Browse for your script using script browser
- Click Run to run the selected script
- Active script is shown in the status bar on the bottom of the window. Script status is also shown in the status bar.
- To edit a script using your default Script Editor, right click on a script and click Edit
- To refresh the script browser: right click anywhere in the browser then select Refresh
- Script developers should be aware that there's a 10 seconds timeout built-in the script engine. If the script engine doesn't see any traffic coming in or going out to ZigBee devices, then it would terminate the script execution with TIMEOUT status.
- See Script Control for more information

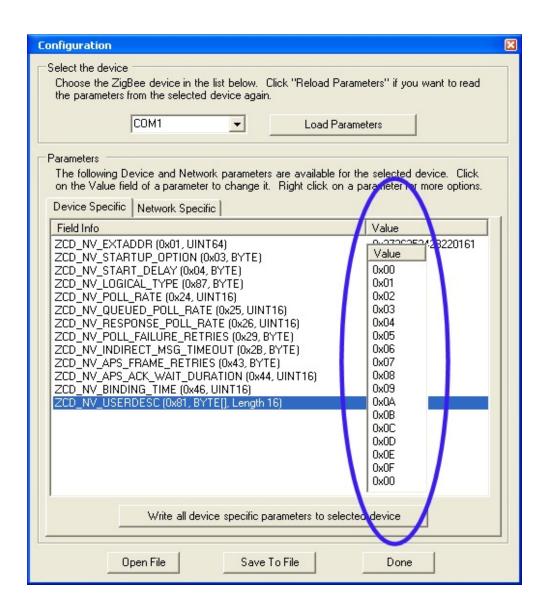
Configuration Main

The main Configuration Window shown below.



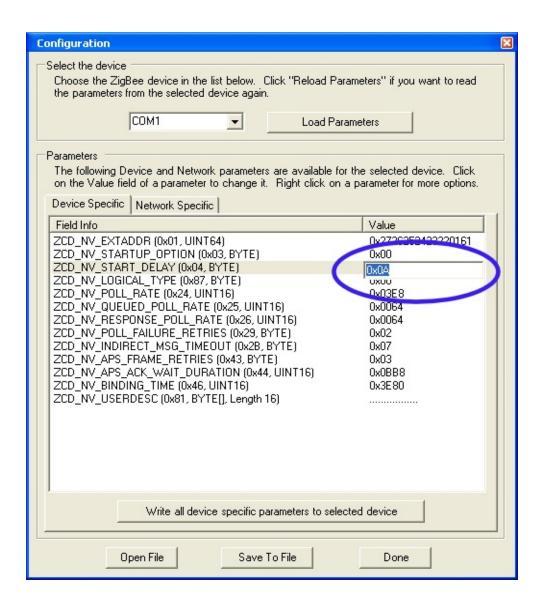
Edit Array Field

To edit an array, click on the value cell of that array parameter. Then edit the items in the array. All values are in Hexadecimal display mode. No data validations are done at this point. Be sure to enter the correct values. Click anywhere outside the editable box to close it.



Edit Nonarray Field

To edit a regular/nonarray parameter, click on the value cell of that parameter. Then edit the item in that field. All values are in Hexadecimal display mode. No data validations are done at this point. Be sure to enter the correct value. Click anywhere outside the editable box to close it.



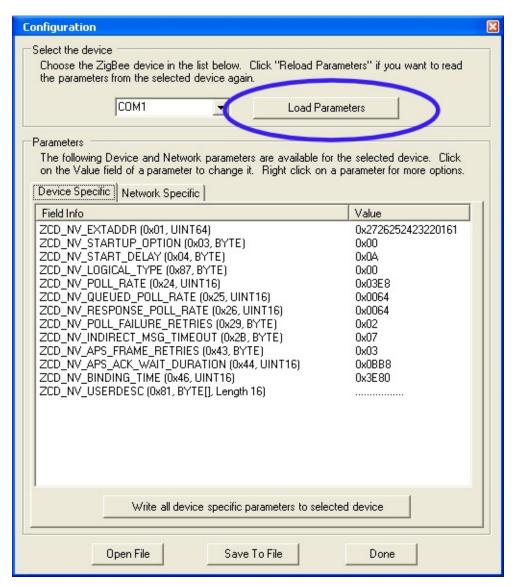
Copyright © 2007 Texas Instruments Inc. All rights reserved.

Load Values from Device

To load values from a device, simply select the device from the device dropdown list. If the device has already been selected then click on Reload Parameters button.

Z-Tool uses series of SYS_GET_NV commands to read the parameters from the selected device. It sends one SYS_GET_NV command per parameter. All SYS_GET_NV commands are sent one at a time. The timeout value is 1 second per SYS_GET_NV message. There are no retries. Parameters with SUCCESS status in the response get displayed with editable fields. All other parameters, no response or bad status, are displayed with "N/A" value.

The default parameters information are defined in TI.MasterKeyListDev.xml and TI.MasterKeyListNwk.xml. Do not modify these files unless you know what you are doing.

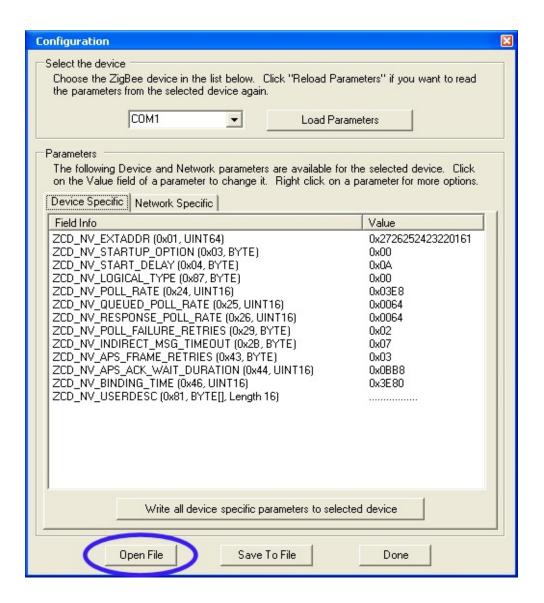


Copyright © 2007 Texas Instruments Inc. All rights reserved.

Load Values from File

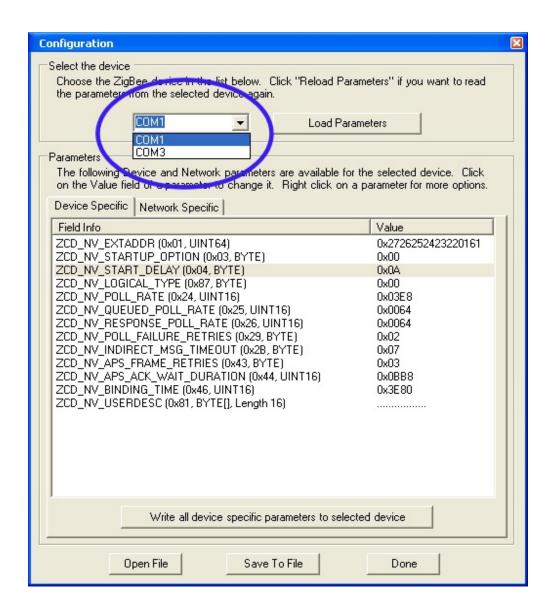
To load values from a file, click on Open File. Select the file then click Open. Open File function automatically detects the type of the parameters, Device Specific or Network Specific. It loads the values to the associated tab accordingly.

Open File function does NOT automatically write the parameter values to the selected device. It simply loads the values to the Configuration Window display. To write the values to a device, select the device and click on the Write button.



Select a Device

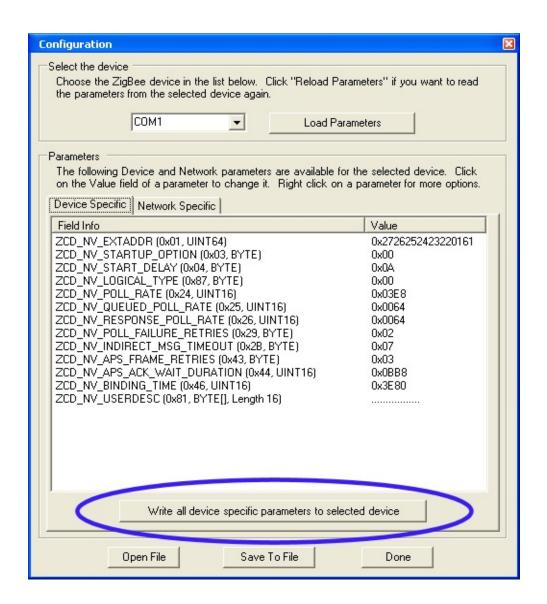
To select a device, simply select it from the device drop down list. Upon selecting a device, Z-Tool automatically reloads the parameters, both Device Specific and Network Specific, from the device.



Save Values to Device

To save values to the selected device, simply click on the "Write all.." button. Each tab, Device Specific and Network Specific, has its own write button. The button for Device Specific tab is labeled "Write all device specific parameters to selected device". The button for Network Specific tab is labeled "Write all network specific parameters to selected device". Each button is only responsible for writing the parameters in its associated tab. To write both Device Specific and Network Specific parameters to the device, click on both buttons.

Z-Tool uses series of SYS_SET_NV commands to write the parameters to the selected device. It sends one SYS_SET_NV command per parameter. All SYS_SET_NV commands are sent one at a time. The timeout value is 1 second per SYS_GET_NV message. There are no retries. Error messages are displayed to notify user of those that failed accordingly.

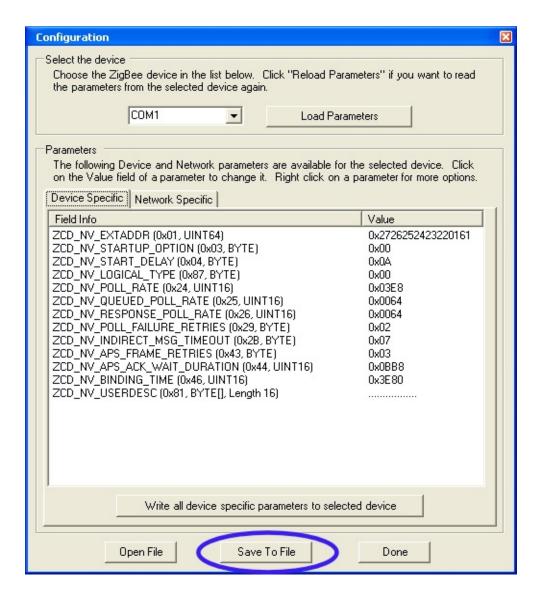


Copyright © 2007 Texas Instruments Inc. All rights reserved.

Save Values to File

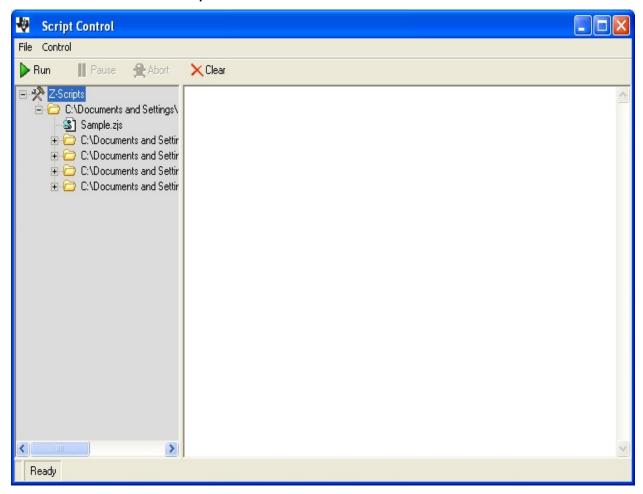
To save the values to file(s), click on Save To File button. Two file names will be prompted, one for Device Specific parameters and one for Network Specific parameters.

Only values for the selected device are saved. To save configuration values for more than one device, use the above instructions to save the values one device at a time.



Script Control Main Screen

Below is what the Script Control main screen looks like.



Script Engine

Script Engine is a predefined global object, which can be accessed anywhere in a script. The name of the Script Engine is **ZEngine**. Script Engine contains the following members

Methods

Complete(hool status)	Quit current carint with complete status. Deced or failed recult
Complete(bool status)	Quit current script with complete status. Passed or failed result was produced.
	·
	Parameters
	status: true for passed, false for failed
	Example
	ZEngine.Complete(true);
Pause()	Pause current script until a key is pressed.
	Example
	ZEngine.Pause();
Pause(int milliseconds)	Pause current script for specified number of milliseconds
	Parameters
	milliseconds: number of milliseconds to pause
	Syample
	Example ZEngine.Pause(1000);
Send(string deviceName,	Send the specified message to the specified device
object message)	
	Parameters deviceName: name of the device. Name is defined in Settings
	-> Alias field
	message: message to be sent
	Example
	var req : SYS_VERSION = new SYS_VERSION();
	ZEngine.Send("Device1", req);
WriteLog(string textMsg)	Display the specified text on the Script Window log panel
	Parmeters
	textMsg: text message to be displayed
	Example
	ZEngine.WriteLog("The quick brown fox jumped over the lazy
	dog");

Events

Subscribe to Script Engine to receive incoming messages.
Interface format. To subscribe, use
ZEngine.add_OnMessageZPI1(this.MessageHandler). To u

		ZEngine.remove_OnMessageZPI1(this.MessageHandler)
OnMessageZPI1(string deviceName ZPI1.MESSAGE_ID messageID, Object msg)	1(string deviceName, ZPI1.MESSAGE_ID	Parameters deviceName: name of device which message is coming f messageID: ID of the message msg: received message
	Example	
		ZEngine.add_OnMessageZPI1(this.Mes
		function MessageHandler(zportName:String id:MESSAGE_ID, msg:Object)
		ld.WE33AGE_ID, llisg.Object)
		}
		Subscribe to Script Engine to receive incoming messages. Interface format. To subscribe, use ZEngine.add_OnMessageZPI2(this.MessageHandler). To u ZEngine.remove_OnMessageZPI2(this.MessageHandler)
OnMessageZPI2(string deviceNa ZPI2.MESSAGE messageID, Object msg)	2(string deviceName, ZPI2.MESSAGE_ID	Parameters deviceName: name of device which message is coming f messageID: ID of the message msg: received message
	Object msg)	Example
		ZEngine.add_OnMessageZPI2(this.Mesfunction MessageHandler(zportName:Stringid:MESSAGE_ID, msg:Object)
		{ }

Writing a Script

Writing a script to run on Z-Tool Script Engine is simple. All you need are some JScript and Object Oriented backgrounds. Make sure that you have read Script Engine topic to have an idea of the functions available in the Script Engine. Look up **ZPI.chm** help file for formats of all messages. Follow these steps to start writing a script:

- 1. Ensure devices are referred to by their correct Alias Names. See Settings page for information on how to set Device Alias Names.
- 2. Import required namespaces
 - import System;
 - import System.Text;
 - import TI.ZPI1; // use OLD Serial Interface format. Replace with 'TI.ZPI2' to use the NEW one
- 3. Declare main class.

```
class ZScriptSampleClass
{
// insert the following code here
}
```

Create a Start function

```
function Start()
{
// insert code here
}
```

Create a message handler function

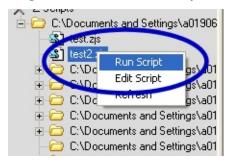
- Subscribe to message event
 - // subscribe to message handler. Put inside Start function.
 - ZEngine.add_OnMessageZPI1(this.MessageHandler);
- End script using Complete function
 - // quit script to Passed status. Put inside MessageHandler function.
 - ZEngine.Complete(true);
- 4. Initialize main class and call start function
 - var zTestScript:ZScriptSampleClass = new ZScriptSampleClass(); // create class object
 - zTestScript.Start(); // run start function

See Script Example for sample script which send and receive messages from two devices.

Running a Script

Follow these steps to run a script:

- 1. Explore the script browser. Ensure your script folders are added to Script Folders under Settings
- 2. Select the desired script
- 3. Right click on the script and select Run Script



Script Example

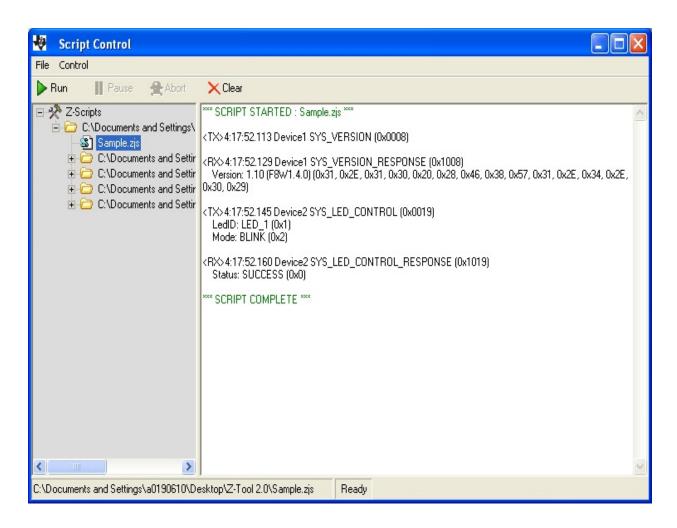
The following is a sample code and its expected output.

```
// Z-Tool Test Script Example
// Written in JScript.NET
import System;
import System. Text;
import TI.ZPI1;
class ZScriptSampleClass
  // define states
  enum TEST STATE
  {
     STATE1.
     STATE2,
  }
  // create private state variable
  private var currentState : TEST STATE;
  function Start()
    // subscribe to message handler
    ZEngine.add OnMessageZPI1(this.MessageHandler);
    // initialize state
    currentState = TEST STATE.STATE1;
    // Send SYS VERSION command to Device1
    var reg : SYS VERSION = new SYS VERSION();
    ZEngine.Send("Device1", req);
```

```
function MessageHandler(zportName:String,
id:MESSAGE_ID, msg:Object)
    switch(currentState)
      case TEST STATE.STATE1:
        // set state to STATE2
        currentState = TEST STATE.STATE2;
        // send SYS LED CONTROL command to Device2
        var req : SYS LED CONTROL = new
SYS_LED_CONTROL();
        reg.LedID =
SYS_LED_CONTROL.LED_NUM.LED_1;
        req.Mode =
SYS LED CONTROL.LED MODE.BLINK;
        ZEngine.Send("Device2", req);
      break:
      case TEST STATE.STATE2:
        // get the response message
        var ledRsp: SYS LED CONTROL RESPONSE =
msg;
        // check response status
        if (ledRsp.Status ==
SYS_LED_CONTROL_RESPONSE.CMD_STATUS.SUCCESS)
          // quit script to Passed status
          ZEngine.Complete(true);
```

```
else
            // write message to log screen
            ZEngine.WriteLog("Failed changing LED for " +
zportName);
            // quit script with Failed status
            ZEngine.Complete(false);
       break;
       default:
var zTestScript:ZScriptSampleClass = new
ZScriptSampleClass();
zTestScript.Start();
```

The following is the expected output



Copyright © 2007 Texas Instruments Inc. All rights reserved.

Troubleshooting

- ⊕ Z-Tool starts up very slow
 □
- **2-Tool appears to hang on startup**
- ⊕ Not all parameters are displayed in Configuration Window
 □
- # My scripts do not show up in script browser |
- Script Window takes a long time to load
 □
- ⊕ How to get more information on Z-Architect SDK
 □

Technical Support

For technical support please contact:

Texas Instruments, Inc.

Low Power Wireless

1455 Frazee Rd, Suite 800

San Diego, CA 92108 USA

+1 (619) 542-1200

lpwsupport@ti.com