



DOMAIN TECHNOLOGIES

Getting Started Guide

Version 1.1

BoxView IDE

Integrated Development Environment

Table of Contents

INTRODUCTION	3
System Requirements	3
INSTALLATION.....	4
License Server.....	4
Registration.....	5
Node Locked License.....	5
Emulator Locked License.....	6
Floating License	6
IDE OVERVIEW.....	8
Windows in BoxView IDE	9
Create a Project.....	9
Create a Source File.....	10
Configure the Debug Session.....	11
Debugging Your Application	13

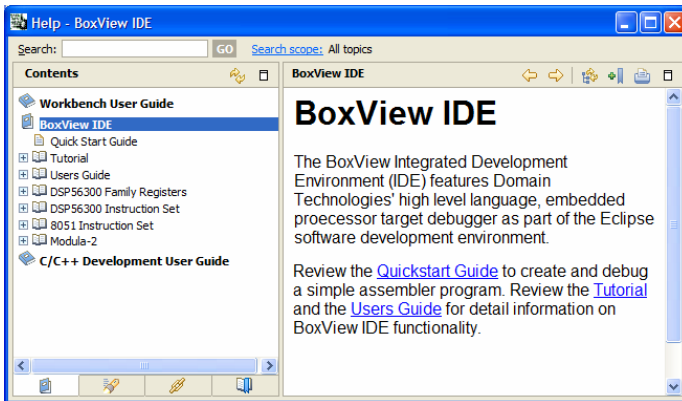
Introduction

The BoxView Integrated Development Environment (IDE) features our high level language, embedded processor target debugger as part of the Eclipse software development environment. The Eclipse platform is an open, industry supported, extensible, software development platform. Programmers create, test and debug applications with fast data access and extensive display capabilities. A typical debugging environment consists of the BoxView IDE running on a host, the host connected to an emulator, and the emulator connected to the target embedded processor; the IDE's simulator can be used when the target hardware is not available.

This Getting Started Guide to BoxView IDE will provide users an introduction to the development environment, the steps to create a simple application, define a debug session, start the simulator, and view project debug windows. Throughout the following pages, each user step is denoted with a button:

- Sample User Step

BoxView IDE contains comprehensive online help including a Getting Started Guide, Tutorial, Users Guide, DSP Instruction Set and Registers, and 8051 Instruction Set. To review, Select Help->Help Contents from the BoxView IDE main window; open the BoxView IDE Booklet in the Help Window:



System Requirements

- MS Windows (2000, XP, Vista)
- Java 1.5 or later
- 8051 Development: SDCC 2.6 or later
- DSP56K Development: Code Generation Tools from Freescale

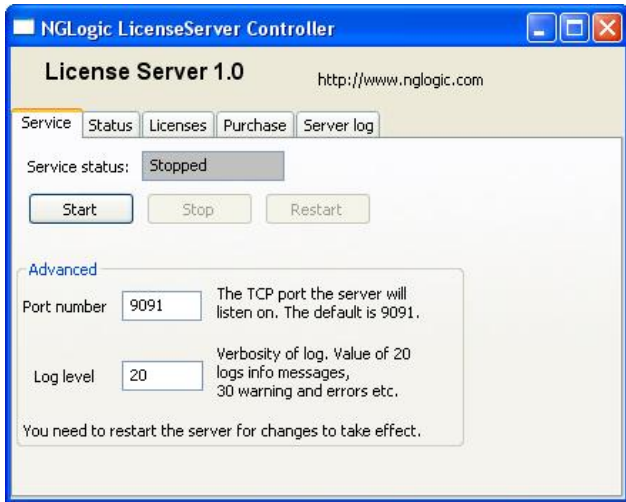
Installation

- Insert the BoxView IDE CD into the drive.
- Choose the appropriate installation for your development environment from the six installation options available with the CD:

Option	Description
BoxView IDE	Install IDE and BoxServer. Requires Java 1.5 or later
License Server	Install License Server for floating licenses
SDCC for 8051	Install SDCC Code Generation tools for 8051
Flash-56K	Install DSP56xxx Flash programming utility
Windows I/O Driver	Install Windows I/O Device Driver for Win NT/2000/XP. Required for PP-JTAG and EVM's Host Interfaces
Java 1.6	Sun J2SE™ Development Kit version 1.6

License Server

If you have purchased a Floating License, you must install the NG License Server as a Windows service on your server machine. When installed, the service will startup each time the server is started. The interface program to the service is called the NG Logic License Server Controller:



Registration

When BoxView IDE is installed, users must obtain a valid license to use the debug features of the product. Each license is specific to the computer on which the software is installed. There are three license types supported by BoxView IDE: node locked, emulator locked, and floating.

- Select pull-down menu *Help ->BoxView IDE Registration* from the main BoxView IDE Window

The Registration dialog will be displayed.

Node Locked License

- Fill in the *User Name* and *Company* entry fields.
- Select *License type* Node-Locked
- Press the *Request Key* button

Your default browser will access the Registration website

- Complete the *User Name*, *Company Name*, *Address*, *City*, *Postal Code*, *Country*, *Phone*, and *E-mail* fields on the website screen.
- Select your *Operating System* radio button.
- Choose the *Send Message* button to invoke the registration request

An email will be returned containing the text for your computer's License key.

- Copy the text from the email and paste it in the *License Key* field of the Registration dialog.
- Choose the *Apply* button to continue.

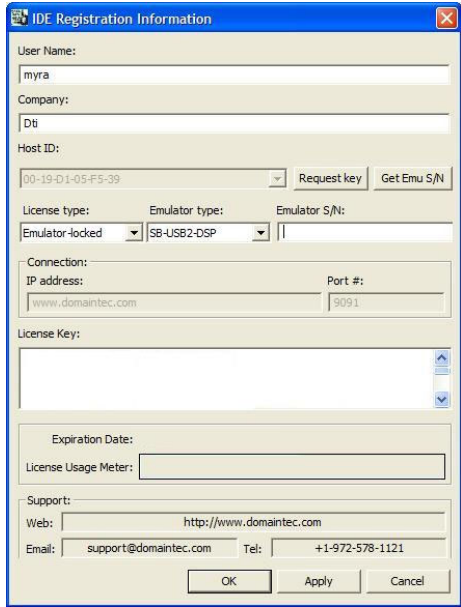
The License *Expiration Date* will now be visible in the Registration dialog.

Emulator Locked License

- Fill in the *User Name* and *Company* entry fields.
- Select *License type* Emulator Locked
- Select *Emulator Type*
- Press the *Get Emu S/N* button
- Press the *Request Key* button

Your default browser will access the Registration website

- Complete the *User Name*, *Company Name*, *Address*, *City*, *Postal Code*, *Country*, *Phone*, and *E-mail* fields on the website screen.
- Select your *Operating System* radio button.
- Choose the *Send Message* button to invoke the registration request



An email will be returned containing the text for your computer's License key.

- Copy the text from the email and paste it in the *License Key* field of the Registration dialog.
- Choose the *Apply* button to continue.

The License *Expiration Date* will now be visible in the Registration dialog.

Floating License

In the NGL License Service Controller, add a license:

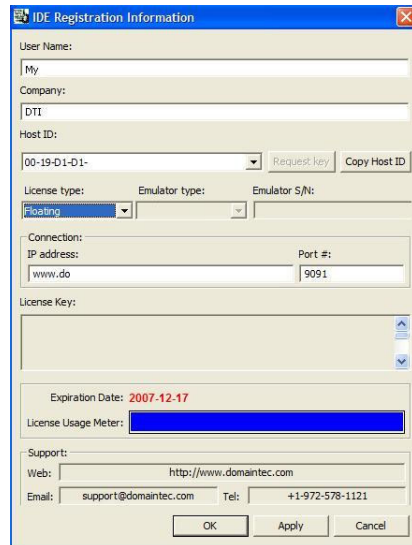
- Go to the *Licenses* tab. Select the *Add License* button at the bottom left.
- Find the appropriate license text file, sent from Domain Technologies, with the provided Disk Browser dialog; the .txt file will be displayed in the provided list box.
- Verify its validity: select the text box and the text *BoxView IDE* will be displayed in the application column next to the text file name

Make sure the service is running; users can start and stop the service through the buttons located on the *Service* tab. The *Status* tab in the Controller is used to monitor license usage; use the *Refresh* button to update the display when a user connects to the License server. The *Status* tab displays available licenses and reserved (in use) licenses.



In the IDE Registration dialog, perform the following steps:

- Fill in the *User Name* and *Company* entry fields.
- Select *License type* Floating
- If your server is the same as your current machine, use *localhost* for your *IP address*; otherwise, enter the *IP address* of the license server machine in the associated field.
- By default the *Port #* is *9091*, however, it must be the same as the port number used by the NG Logic License Server Controller program (found on the *Service* tab page)
- Select *Apply*



If the service is not running or all licenses are in use, an error dialog will be displayed and the text "*License Time Expired*" will be displayed in the *License Usage Meter* field. If the service is running and a license is available, the *License Usage Meter* field will be populated with a blue line.

IDE Overview

BoxView Integrated Development Environment (IDE) features the high level language, embedded processor target debugger as part of the Eclipse software development environment. The Eclipse platform is an open, industry supported, extensible, software development platform.

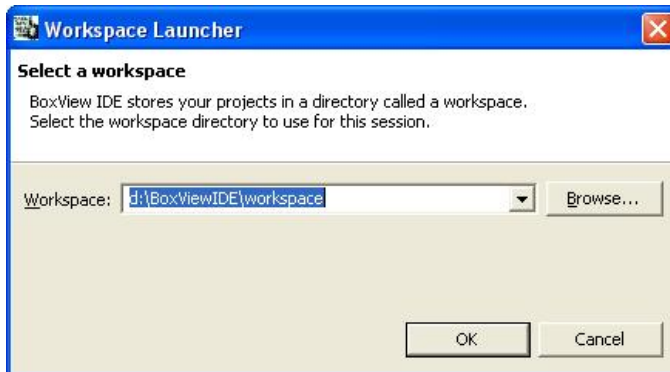
- After installing and registering, load BoxView IDE from the desktop.

Several windows are displayed, including a navigator window which shows folders containing sample projects. There are three basic types of resources that exist in the Development Environment; a resource is a collective term for the projects, folders, and files viewed in the IDE:

Files	Files in the file system
Folders	Directories on a file system. Folders are contained in projects or other folders. Folders can contain files and other folders.
Projects	Contain folders and files. Projects are used for builds, version management, sharing, and resource organization. Like folders, projects map to directories in the file system. (When you create a project, you specify a location for it in the file system.)

The directory location on your computer where the resources will be stored is called a workspace. The sample projects are stored in the default installed location. The Development Environment supports multiple workspaces.

- Create a new workspace by selecting *File->Switch Workspace...* from the main IDE menu.
- Use the *Browse* button to define the directory that will hold your project files and folders:

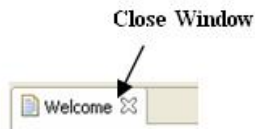


Windows in BoxView IDE

The BoxView IDE contains many windows or views. A set of views is called a perspective. The IDE has 2 predefined perspectives: the *Project Perspective* and the *Debug Perspective*. The *Project Perspective* displays a set of windows valuable when manipulating project files. The *Debug Perspective* displays windows valuable when debugging a project. Users can change perspectives by choosing menu option *Window->Open Perspective* or by using the associated button on the IDE's button bar:



Users can add windows to any perspective by using the IDE's *Window* pull-down menu. Users can close windows in a perspective by selecting the window's close icon:

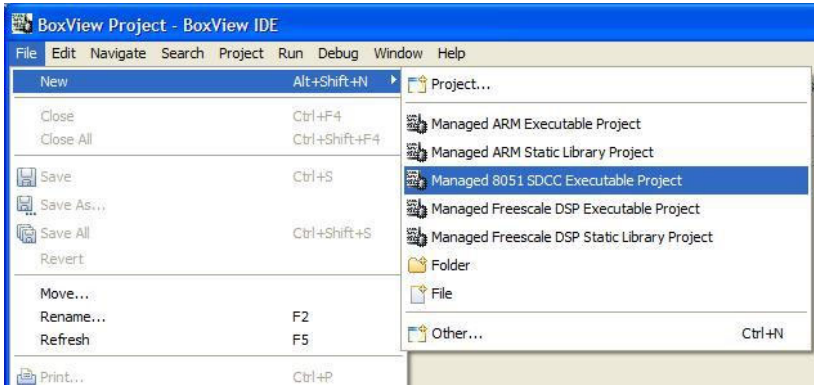


Create a Project

The Navigator window provides a hierarchical view of the resources in your workspace; users can open files for editing by double clicking on the filename in the navigator window.

- Right-click in the Navigator to open a pop-up menu that allows operations such as create new, copy, paste, import, refresh, etc.

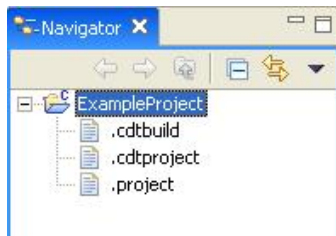
- For 8051 projects, choose *New->Managed 8051 SDCC Executable Project*, for Freescale projects, choose *New->Managed Freescale DSP Executable Project*:



The New Executable project dialog will be displayed.

- Define the new project name in the entryfield; type *ExampleProject*
- Choose the *Finish* button.

The Navigator will now contain your new project. Double click on the *ExampleProject* in the Navigator window. Three files are created automatically by the IDE and are contained in the project:



Create a Source File

Create a source file in your new project that will contain a simple program.

- Select *ExampleProject*, choose *File->New->File* from the main menu.

The New File dialog will be displayed.

- Type *simpleExample.c* in the file name entry field.
- Choose the *Finish* button.

An empty source file named *simpleExample.c* has been added to the *ExampleProject* project. The empty source file is opened in the IDE. Write the content of the new application.

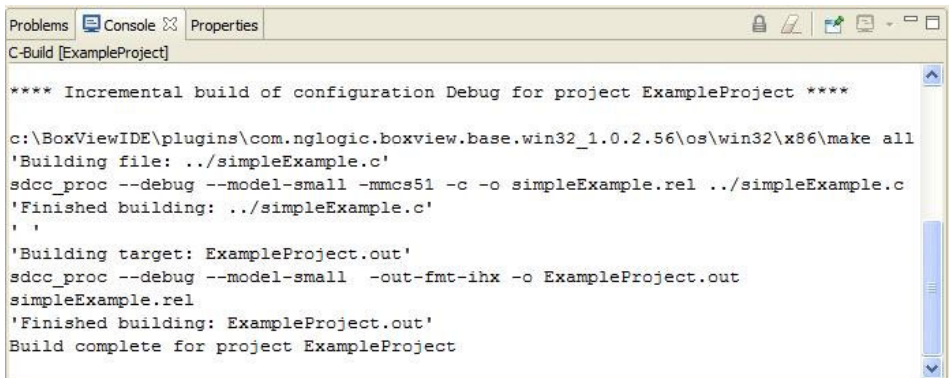
- Fill the *simpleExample.c* editor window with the following instructions:

```
int val;
int val2;

int test (int v)
{
    return v * 2;
}

int main(void)
{
    val++;
    while (val < 100)
    {
        val2 = val / 2;
        val = test(val2);
    }
    return 0;
}
```

- Save the *simpleExample.c* program ; choose *File->Save*
- Anytime a source file is saved, the automatic build operation triggers
- Select the *Console* window for detail build information results:



```
Problems Console Properties
C-Build [ExampleProject]

**** Incremental build of configuration Debug for project ExampleProject ****

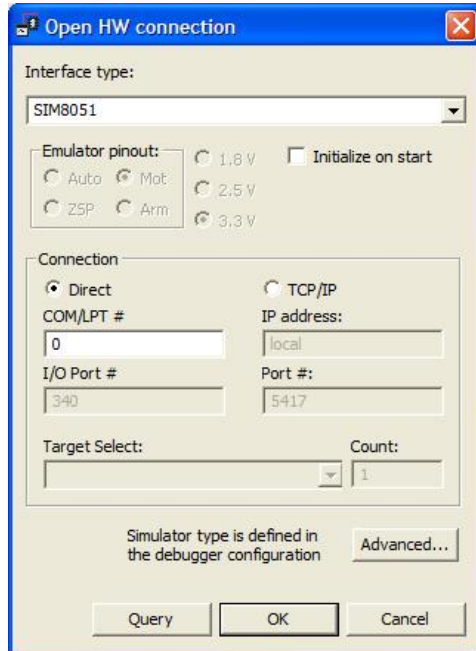
c:\BoxViewIDE\plugins\com.nglogic.boxview.base.win32_1.0.2.56\os\win32\x86\make all
'Building file: ../simpleExample.c'
sdcc_proc --debug --model-small -mmcs51 -c -o simpleExample.rel ../simpleExample.c
'Finished building: ../simpleExample.c'
'
'Building target: ExampleProject.out'
sdcc_proc --debug --model-small -out-fmt-ihx -o ExampleProject.out
simpleExample.rel
'Finished building: ExampleProject.out'
Build complete for project ExampleProject
```

Configure the Debug Session

Establish a connection to the target hardware and configure the session.

- Open the connection to the target, by choosing menu option *Debug->Open Connection*

- Choose an embedded processor simulator for the interface type; 8051 projects choose *SIM8051*, Freescale projects choose *SIM56300*
- Press **OK** to connect:



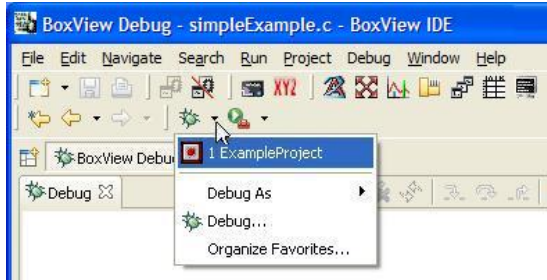
- Change perspectives, to obtain the best views for debugging, by choosing menu option *Window->Open Perspective->BoxView Debug* or use the *BoxView Debug* button on the IDE's button bar
- Select from the main menu *Run-->Debug*
- In the *Debug* session configuration dialog, select *DSP Debugger* in the *Configurations* listbox and choose the *New* button. The *Debug* fields on the right pane will be replaced with fields defining a new session configuration.
- From the *Main* tab of the *Debug* configuration dialog, Name the debug session by typing *ExampleProject* in the *Name* entry field.
- Fill in the *Project* entry field by using the *Browse* button to select *ExampleProject*
- The *Application* entry field will be automatically filled in with the output contained in the chosen project.
- Select the *Debugger* tab of the *Debug* configuration dialog
- In the *Debugger target* drop box, define the connection type for the debug session to be *Emulator*

- Select the *Stop at main() on startup* checkbox.
- Save the session configuration by selecting the *Apply* button

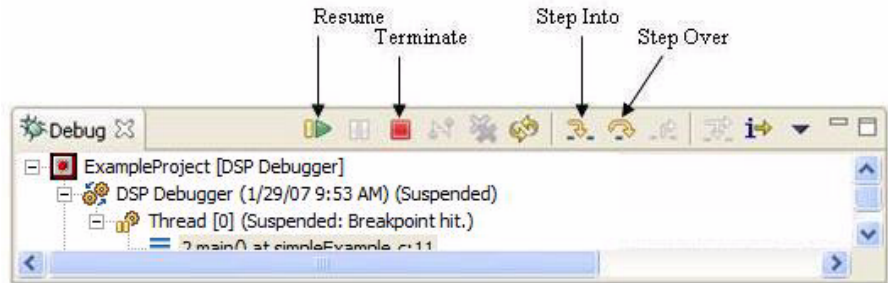
Debugging Your Application

Once the debug session has been configured, the debugger can be started.

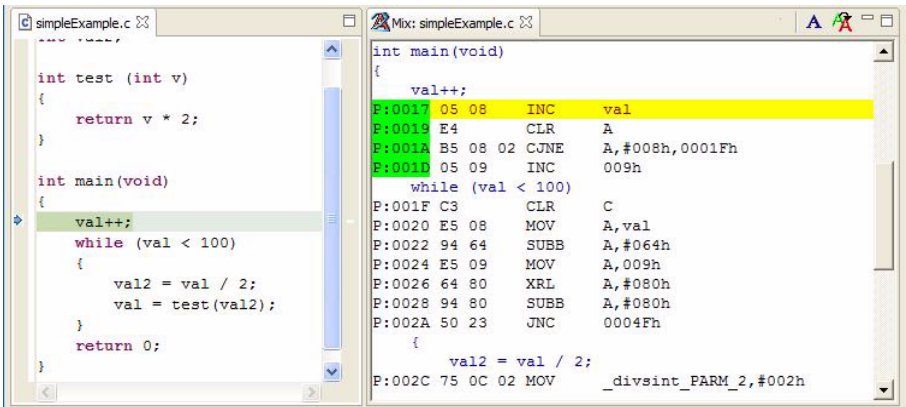
- Start your debug session by selecting the *Debug* button. Future debug sessions for this project can be started through the debug icon:



Many windows are populated in the Debug perspective when the application is started. In the debug window, note the application has stopped (as defined in our session configuration):

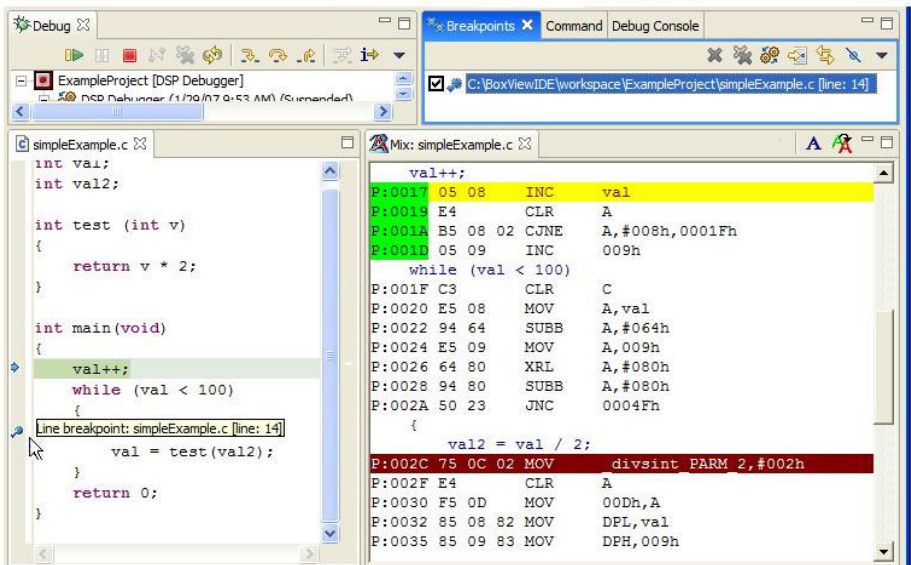


Next, the Source window shows an arrow in the margin for the *Current Program Counter* at the first instruction in the program. The Code Window shows the *Current Program Counter* highlighted in yellow; the assembler lines representing the *Selected Source Line* are highlighted in green.



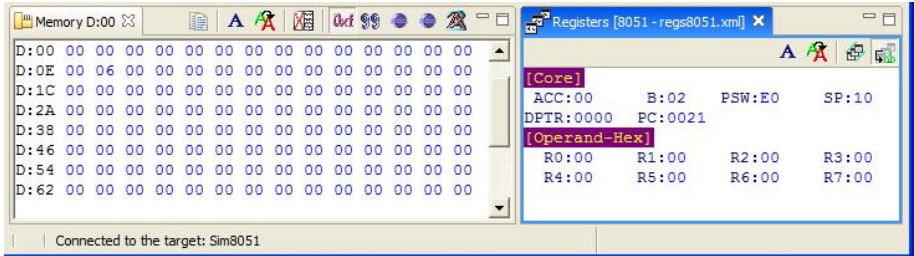
- Set a breakpoint by double clicking in the margin of the source window on the `val2=val/2` instruction

The next graphic shows: the source window with added breakpoint on the selected instruction; the code window with breakpoint in burgundy; and the breakpoint window with the line number of the source file's breakpoint:



- If your debug perspective does not contain a Memory window, add one by selecting *Debug -> Memory* from the main BoxView IDE Window.
- If your debug perspective does not contain a Registers window, add one by selecting *Debug -> Registers* from the main BoxView IDE Window.

- Perform a right mouse click in the *Registers window* for the window menu; select the option *load registers->8051* or *load registers ->56300*
- Right click in the *Registers window*; select *Registers Section-> Core* to display the core registers.
- Right click in the *Registers window*, select *load registers->Operand-Hex*.



- Use the *Step Into* button of the Debug window (or IDE menu option *Run->Step Into*) to walk through the project.
- Use the *Resume* button (or IDE menu option *Run->Resume*) to hit the breakpoint
- In the Debug window, use the *Terminate* button (or IDE menu option *Run->Terminate*) to halt the execution.
- Choose *Debug->Close Connection* to close the simulator target hardware connection

For more information on debug windows, review the *Debug Menu Options* Section of the Users Guide.



© Domain Technologies Inc.
811 East Plano Pkwy • Suite 115
Phone 972.578.1121 • Fax 972.578.1086
<http://www.domaintec.com>
Email: support@domaintec.com