Delta-Sigma ADC EvaluaTIon Software User Manual

This document contains instructions, operating information, and details for using the Delta-Sigma ADC EvaluaTIon Software with evaluation modules from Texas Instruments.

Software Requirements

- PC with:
  - Microsoft® Windows® 7, or Windows 10
  - .NET Framework 4.5, or higher
- 35MB of storage on the PC
- PC with available USB port

This document describes the generic functionality of the EVM application software. Functions specific to the device (register settings, controls, scripts, and so forth) are described in the individual device EVM user guide.

Contents

List of Figures

Trademarks

ADCPro is a trademark of Texas Instruments.
MATLAB is a registered trademark of MathWorks, Inc.
Microsoft, Windows, Excel are registered trademarks of Microsoft Corporation.

1   Glossary and Abbreviations

1.1   *Abbreviations*

ADC –   Analog-to-Digital Converter
USB –   Universal Serial Bus
EVM –   Evaluation Module
LSB –   Least Significant Bit
MSB –   Most Significant Bit
PC –   Personal Computer

1.2   *Glossary*

Device Manager refers to the Windows® Device Manager.

2 Installation

This section provides instructions for installation of the software and drivers.

2.1 Core Application Installation

Use the following steps to install the core application:

1. Download the latest software installer from the EVM product folder on www.ti.com.

   NOTE: Registration for the software at www.ti.com is required. Once you have registered and completed a brief request form, the software should be available for download.

2. Locate the installer named *DELTASIGMAEVAL-GUI-installer.exe* and double-click to run. The default install directory is C:\Program Files(x86)\Texas Instruments\DSEvalSW. The product version can be viewed under the *File > About* tab of the installer.

3. Follow the prompts and agree to the license terms.

2.2 Device Package

The core application supports multiple EVMs and requires a device package for each device EVM. The main software identifies the connected hardware at startup and uses the appropriate files for the specific device.

The device package (-DVCPKG) should be installed after the core application and is located in the product folder (or provided).

2.3 Driver Installation

2.3.1 Automatic Installation

The software drivers install as part of the core software installation process. The drivers are pre-installed and require that the hardware be connected to the PC to complete the installation process.

During the installation processes, you may receive a warning message regarding the failure of the driver installation. This error can result if the drivers are already installed or due to other installation failures. This message should not affect the installation of the main application. After the completion of the installer, check to see if the drivers were correctly installed. The correct driver shows as "Generic Bulk Device" (with VID = 1CBE and PID = 003) in the *Device Manager*.

If the drivers installed correctly, no further action is required. If the drivers are not correctly installed, please use the manual process shown in Section 2.3.2.

2.3.2 Manual Driver Installation

The driver installer and files are located in the installation directory under the ***drivers*** folder.

2.3.2.1 Automated Manual Driver Installation

The ***installers*** folder (under the ***drivers*** folder) contains a 32-bit and 64-bit driver installer that attempts to automatically install the files to your PC.

2.3.2.2 Manual Driver Installation

If both methods (Section 2.3.1 and Section 2.3.2.1) fail, use the *usb_dev_bulk.inf* file to manually install the driver once the hardware is connected.

If Windows automatically installs a different driver than the included hardware driver, manually update the driver to the correct one in the *Device Manager*.

When the hardware is connected and the correct driver is installed, the hardware is shown as "Generic Bulk Device" with VID = 1CBE and PID = 003.

Using the *Device Manager*:

1. Locate the correct device and select *Update Driver Software* from the right-click menu
2. Select the *Browse my computer...* option.
3. Navigate to the **drivers** folder and select *usb_dev_bulk.inf.*

3 Software Startup

When the software is started, a brief splash screen displays before opening the main application screen. When the main application screen opens, one of two screens are presented.

3.1 Startup Screen (No Hardware Connected)

If no hardware is connected to the PC at the time the software is started, the software starts and shows Figure 1. Steps on how to proceed from this screen are located on the right side of the screen. Once the hardware is connected to the PC, the software changes to the screen shown in Figure 2
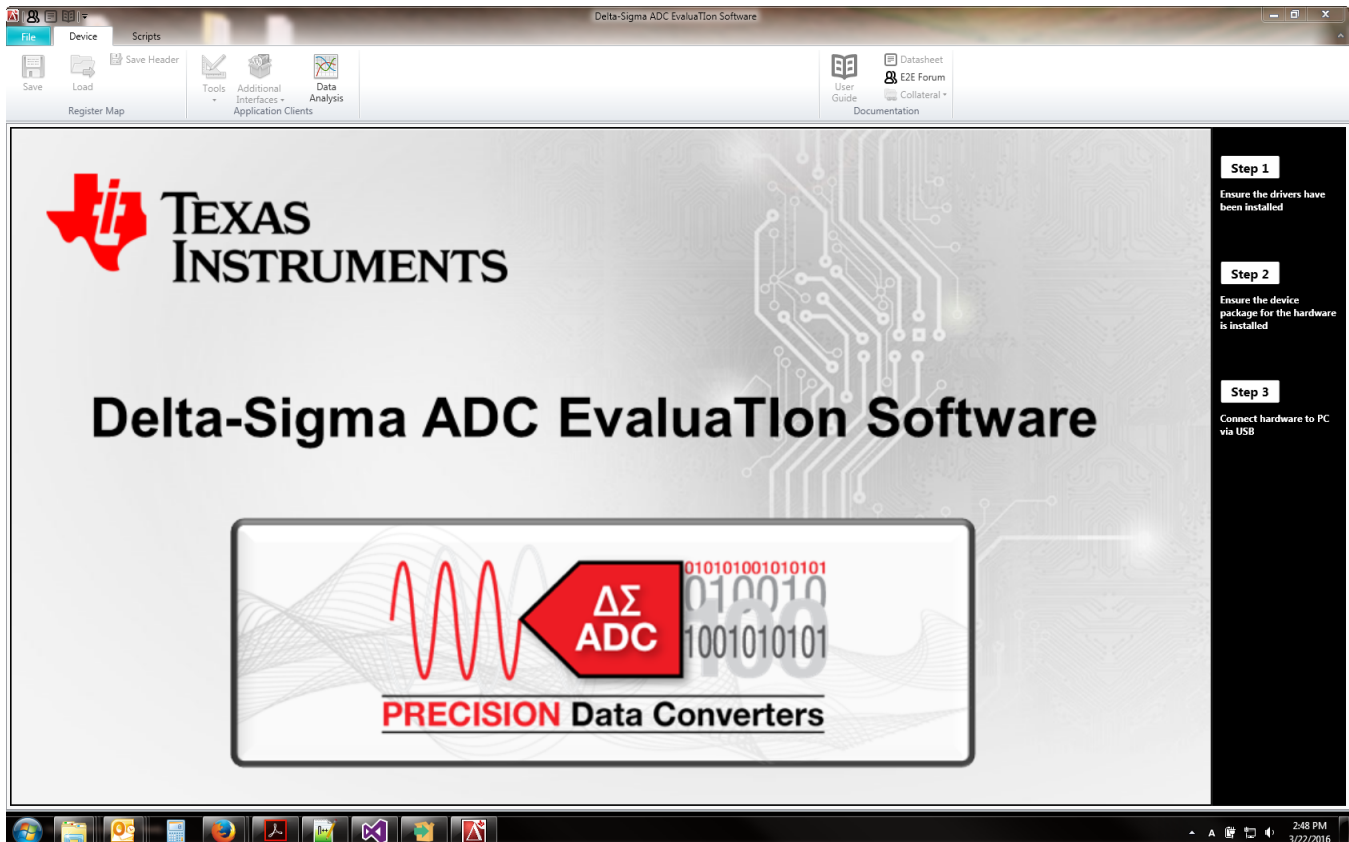

Figure 1. Startup Screen With no Hardware Connected

3.2 Startup Screen (Hardware Connected)

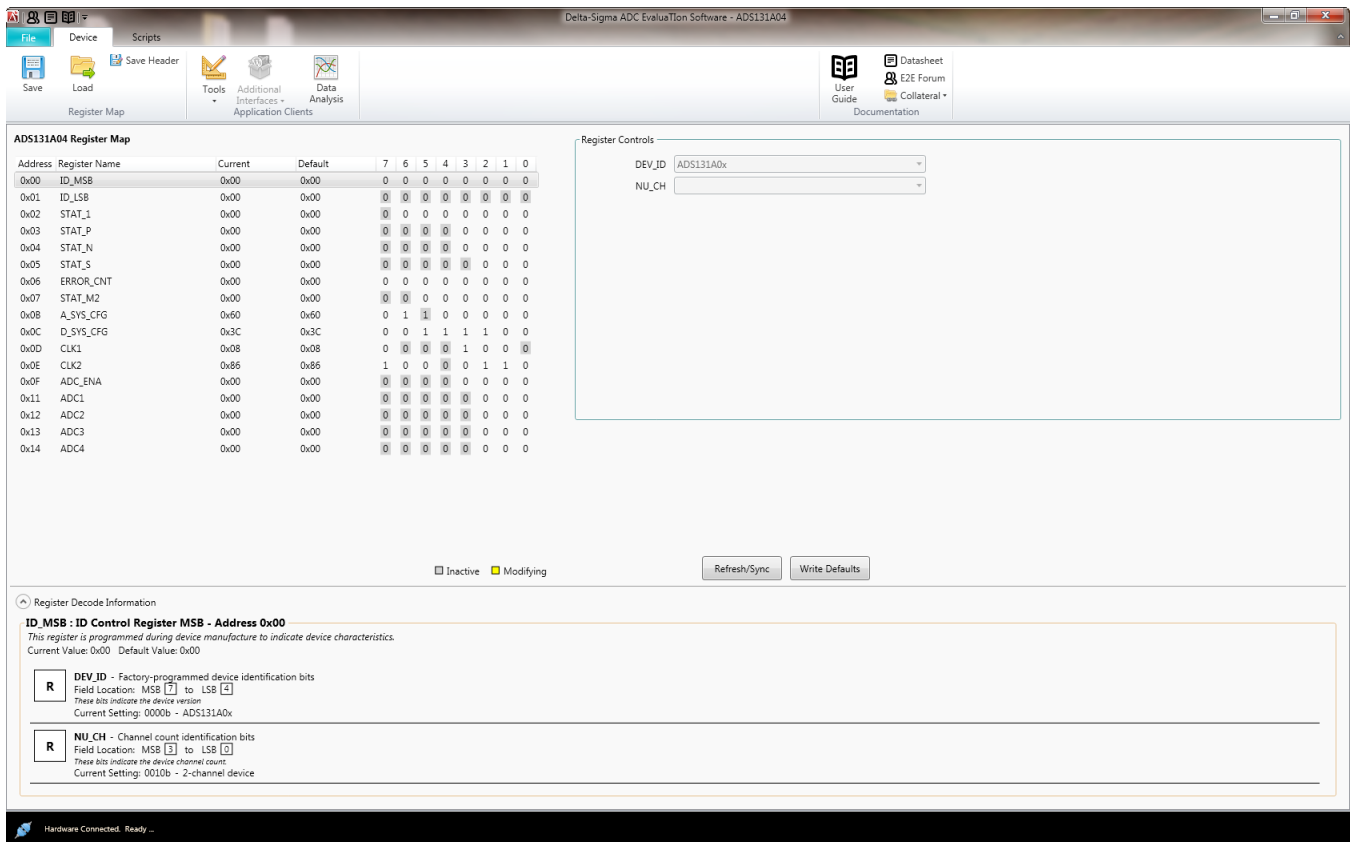Figure 2 is displayed after the hardware is connected, drivers are installed correctly, and the software is started.

Figure 2. Startup Screen With Hardware Connected

4 Application Window

The application window is divided into 3 key areas as shown in Figure 3:
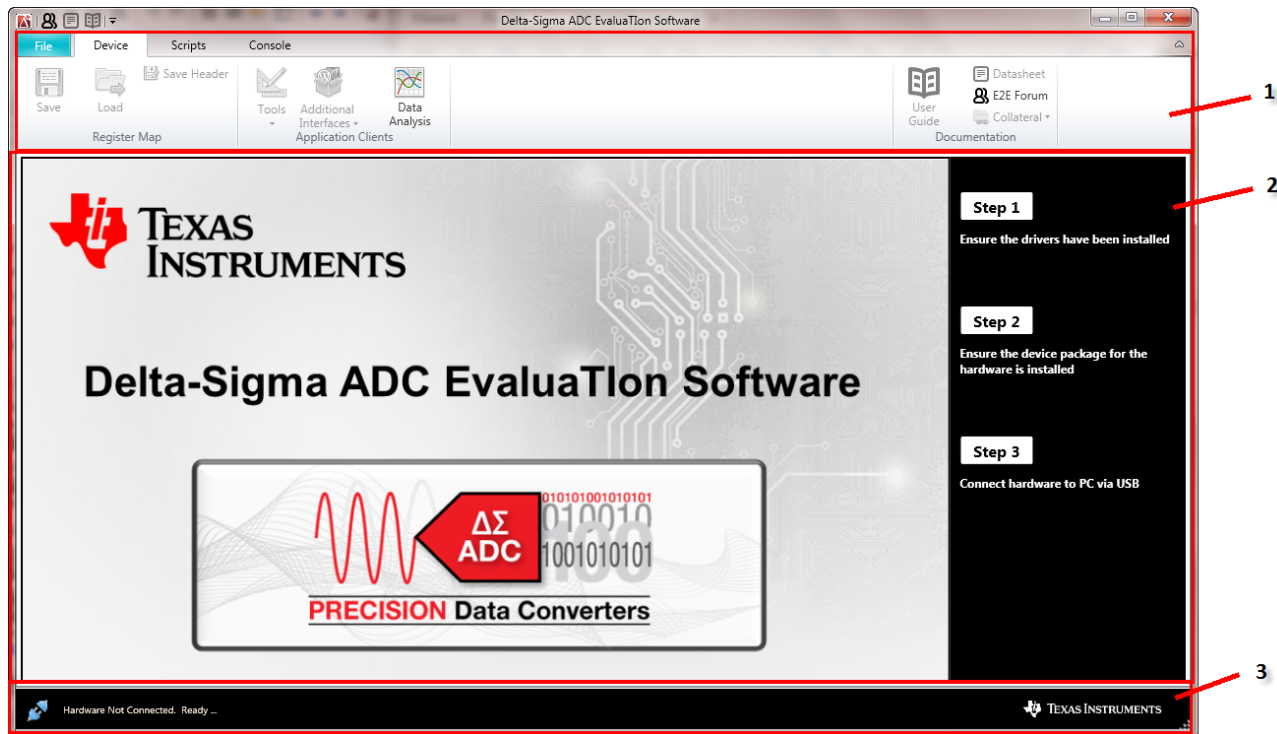
1. Application menu
2. Main display area
3. Status bar

Figure 3. Application Window Areas

4.1 Application Menu

The application menu is shown at the top of the screen and provides different tabs and menu options that depend on the tab selected. The tabs and general descriptions are provided in the following list:

- **File** - This tab provides general application functions for Master Log Browser tool, Options, and About tab (Section 5).
- **Device** - This tab provides access to the register map once an EVM is connected and allows manipulation of the register settings. It displays the individual register settings and describes register functionality.
- **Scripts** - This tab provides the capability to run and write scripts to configure the ADC.
- **Console** - This tab provides manual commanding capabilities and view of the Activity Log for communication traffic. This tab is not displayed, by default, and is enabled in the Options settings.

4.2 Main Display Area

The main display area of the application window adapts to provide content based on the tab selection.

4.3 Status Bar

The status bar is located along the bottom of the application window. This area provides connection information, status messages, and collection progress from the program.

The left side of the bar shows a graphic that indicates the connection status of the hardware and software.

Next to the connection status graphic is the status message portion of the bar. This provides various status messages during operation.

A progress bar displays in the status bar to the left of the TI logo during data acquisition to indicate the progress of the data collection.

5 File Menu

The **File** menu provides additional options when using the software.
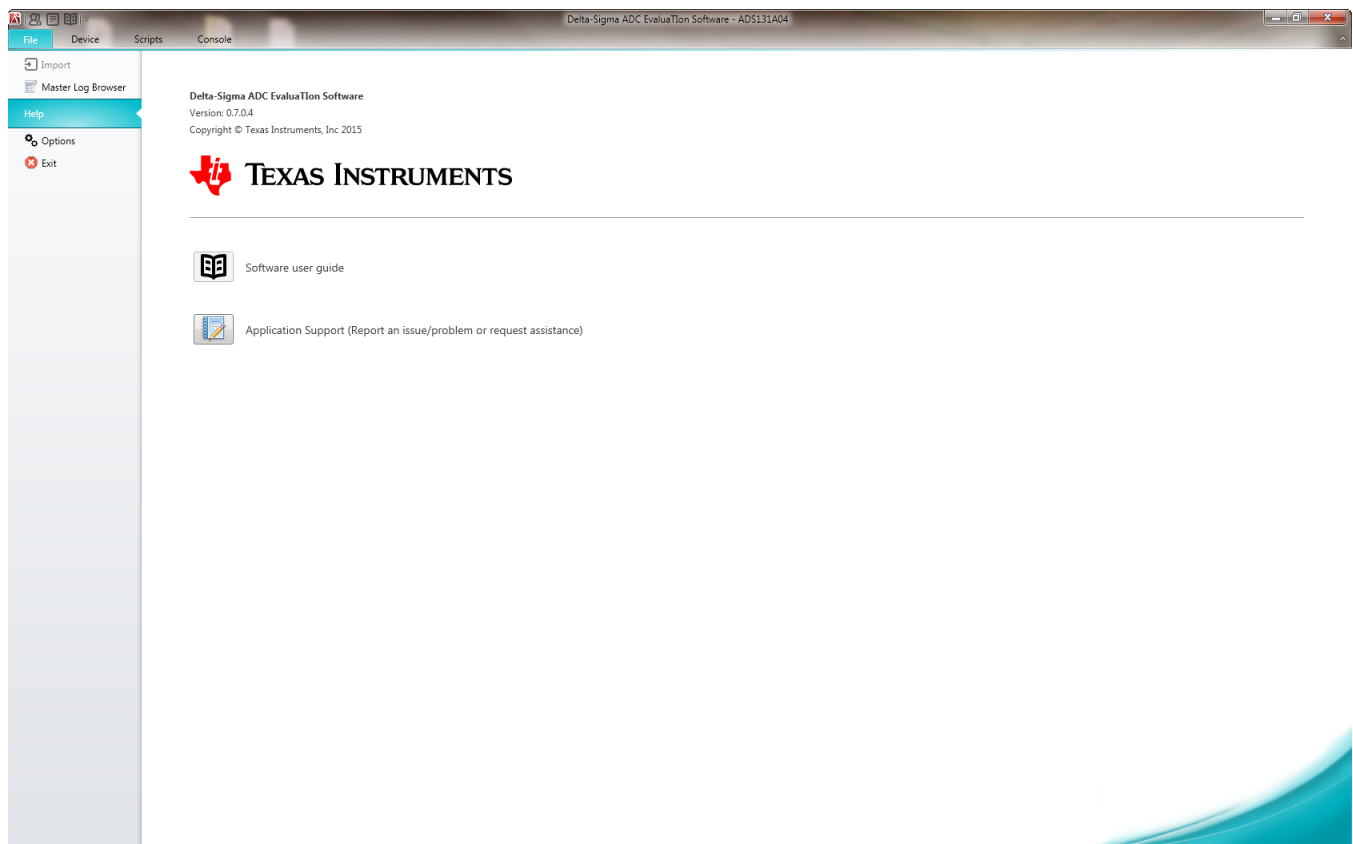

Figure 4. Application File Menu

5.1 *Master Log Browser*

This menu option opens another window that allows the user to see a list of all commands and filter them by list. See Section 10, *Master Log Browser*.

5.2 *About*

The *About* item displays the screen shown in Figure 4. The software version and copyright information are displayed for reference. Additionally, two controls are provided for accessing the software user guide and applications support.

5.2.1 Software User Guide

This button opens a browser window that displays the software user guide for reference purposes.

5.2.2 Application Support

This button opens a form that is used to send an email for support related to the software or device. The form is shown in Figure 5.

Alternatively, post the question in the TI E2E Forum by using the **E2E Forum** button in the **Documentation** menu group located in the menu.
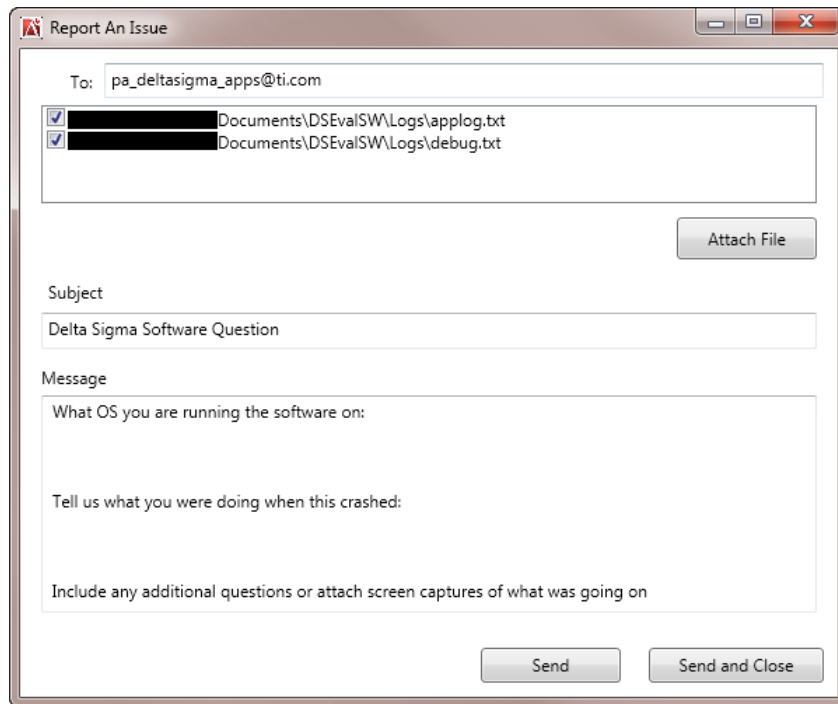
Figure 5. Applications Email Form

5.2.2.1   *Email Address*

The *To:* field should be auto-populated with the destination email address.

5.2.2.2   *Attachments*

The box below the *To:* field contains a list of files that are attached to the email. By default, the log files for the software are attached to the email. These log files help TI understand what was happening when you experienced an issue. For improved efficiency in resolving issues, include screen captures or other files by clicking on the **Attach File** button and select the file to add. Any file listed in the box is attached to the email, if the corresponding checkbox is checked.

5.2.2.3   *Subject*

The *Subject* field provides user entry for the subject field of the email. Change the default value to better describe your inquiry.

5.2.2.4   *Message*

The *Message* field provides user entry for the email subject. By default, there are a few questions populated that are applicable for software questions. Please provide as much detail as possible.

5.2.2.4.1   *Sending the Email*

Clicking the **Send** or **Send and Close** button attempts to send the email with the information provided. If there is an error, a message is displayed with instructions for sending the email manually.

5.3   *Options*

This menu item provides various user options related to the application.

The **Accept** button updates and saves any changes to the user settings for use during future evaluation efforts. The **Cancel** button prompts the user to confirm their selection and closes the Options window.

5.3.1 General Options

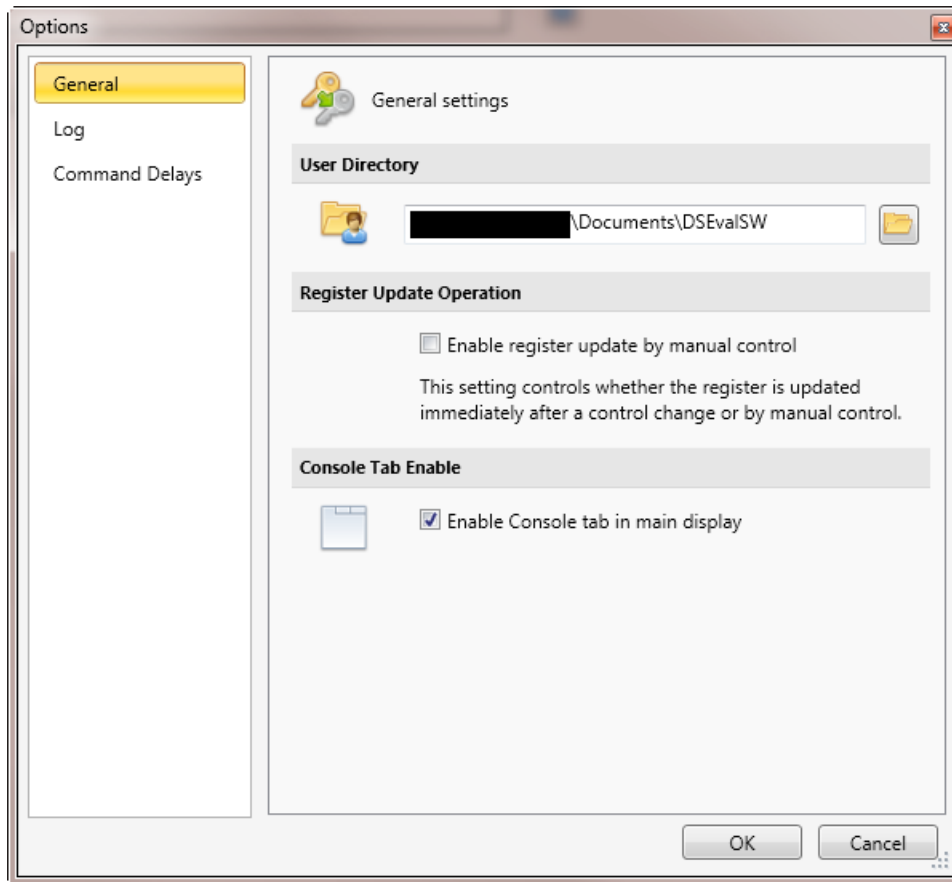Figure 6 shows the **General** preferences available for change.

Figure 6. Options Display

5.3.1.1 *User Directory Controls*

This is the path to the directory that contains all the device information. This location is set by default, but can manually be changed to use different data.

5.3.1.2 *Register Update Operation*

This option enables the user to change controls for the register settings without creating a register write or update after every control change. Once all the settings within the register are configured to the user's specifications, a manual write command must be issued using the **Write Selected Register** button that appears once this option is enabled on the **Device** tab.

> **NOTE:** Without the manual button press, the values in the register map table reflect changes but the device hardware is not synchronized to these values.

5.3.1.3 *Console Tab Enable*

This option enables the **Console** tab for manually entering commands or viewing the larger communications log display.

5.3.2 Log Settings Options

Figure 7 provides preferences relating to the log file for the application.

5.3.2.1 Log Newline Inclusion

Selecting the *Add New Line to All Log Items* checkbox, adds a newline to each log item in the log.

5.3.2.2 Log Settings

The controls for the log settings allow the user to specify the level of details that are displayed in the activity log window. Each communication type allows for **Full Detail**, **Summary**, or **Off**.

- **Full Detail** provides the most verbose log settings for the log item.
- **Summary** provides a brief summarized version of the log message.
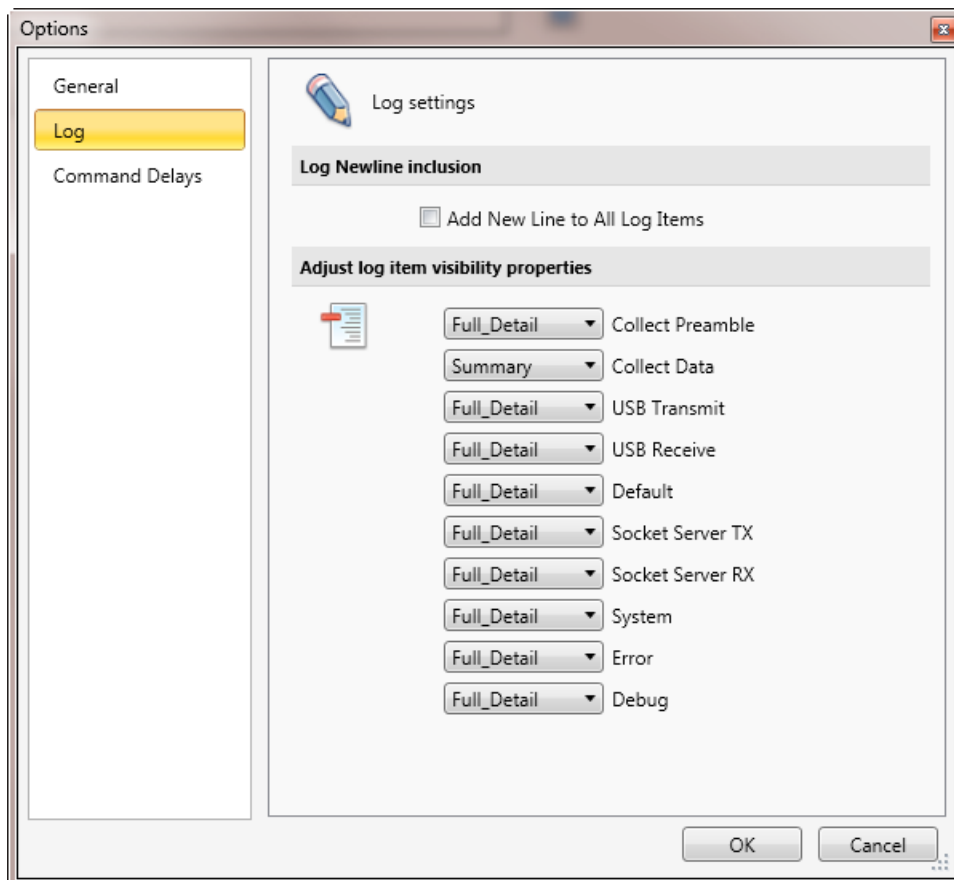- **Off** suppresses the output of the log item message to the display.

Figure 7. Log Preferences Tab

5.3.3 Command Delay Timing Options

Figure 8 provides settings that relate to various delays and timing in sending the commands to the hardware from the software.

Delays are provided to allow a command time to execute before the next command is sent.

NOTE: All timing options are relative to the PC and are **not** exact timings for execution. These delays are from the time the command is sent by the PC to the next command sent from the PC. The timing of receipt and execution is dependent on USB and receive loop delays.
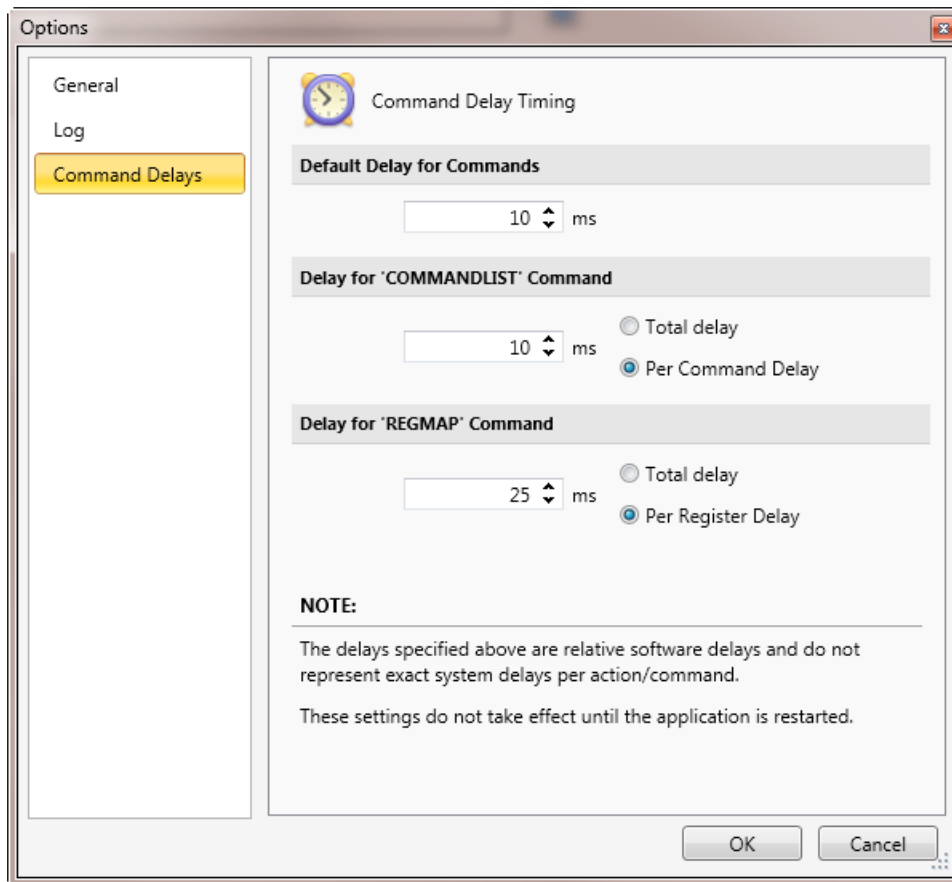
Figure 8. Command Delay Timing Settings

5.3.4 Default Delay for Commands

This number is the delay used between sending commands by default.

5.3.5 COMMANDLIST Command Delay

This value is the delay used before sending a command after the *COMMANDLIST* command. The value can be specified as a single value for the delay or the delay-per-device command (multiply this number by the number of device commands).

5.3.6 REGMAP Command Delay

This value is the delay used before sending a command after the *REGMAP* command. The value can be specified as a single value for the delay or the delay-per-device register (multiply this number by the number of device registers).

6 Device Tab

The Device tab of the application provides a register map interface to interact with the device and view the current settings of the device.

The following descriptions are provided assuming that a device is connected and the register map functionality is displayed.

6.1 Device Tab Menu

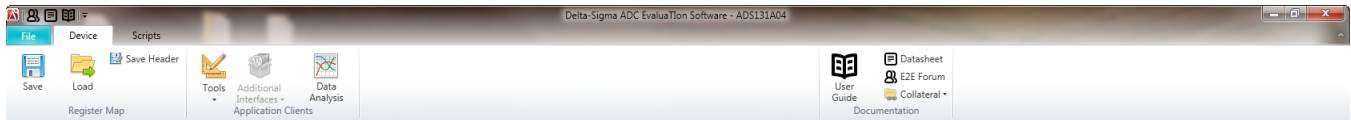The main tab menu provides menu items for interacting with the application.

Figure 9. Device Tab Menu

6.1.1 Register Map Menu Group

This group of controls in the application menu relates to saving and loading the register map settings to and from the application.

6.1.1.1 Save

The *Save* menu item allows the current register settings to be saved to a text file for future use in this application or for reference in other applications.

The file is formatted as a tab-delimited file with 2 columns; the first column contains the register name (as defined in the register map shown in the datasheet) and the second column contains the register value as a two-character (byte) hex value. An example of the file format is shown in Section C.2

6.1.1.2 Load

The *Load* menu item allows a previously saved register file to be loaded into the application.

6.1.1.3 Save Header

The *Save Header* menu item allows the current register settings to be saved to a standard C header file that can be directly linked into a firmware code project for initialization of the device at startup.

The header file provides *#define* statements for various register related values and an array of structs containing the register address and values.

The file details and an example file are shown in Section C.3.

6.1.2 Application Clients Menu Group

This area of the register map menu provides other tools or applications that are integrated with the main application.

6.1.2.1 Tools Menu

The *Tools* menu item lists various support tools and applications that are specific to the device under evaluation. The content of this menu changes for each device based on device package inclusions. These items may include calculators, widgets, spreadsheets, PDF documents, as well as other items. The tool or application is invoked or started by clicking the item in the drop-down menu list.

6.1.2.2 Additional Interfaces

The *Additional Interfaces* menu item lists various interfaces/UI windows that can provide additional view/control of the device. These items can take the form of separate windows that provide customized interfaces or views for specific applications, demos, or alternate programming and control methods. All available interfaces are list in this menu and are started by clicking the item in the menu drop-down list. If the menu item is disabled, no additional interfaces are available for this device. The content of this menu changes for each device based on device package inclusions.

6.1.2.3 Data Analysis

The *Data Analysis* menu item opens the *Analysis Engine* window. The window allows the user to collect, view, and analyze data.

At this time, the **Data Inspector** and **Time Domain Analysis** are available for data analysis. There are plans to include future analysis options. These options will be made available when the functionality is complete. See Section 9 for more details.

6.1.3 Documentation Menu Group

This area provides access to various links and documents that are related to the current device under evaluation.

This menu group exists on all tabs to provide quick access to the documentation. The menu group contains the following items:

- **User Guide**
  This item opens the device EVM product folder in your default browser. From the product folder, you can download the user guide or explore other information for the EVM under evaluation.

- **Data Sheet**
  This item opens the device product folder in your default browser. From the product folder, you can download the datasheet or explore other information for the current device.

- **E2E Forum**
  This button opens your default browser to the TI E2E Forums. This is a great place to post search through other forum questions and answers or post your own questions.

- **Collateral**
  This button provides links to other documents that are available for the current device. These items may include application notes, reference designs, white papers, or other materials.

6.2 Operation

The main portion of this window is dedicated to the register map view and provides the ability to manipulate the device settings. While the register contents can be written and read directly using the write and read commands, this interface provides a graphical and informative view of the individual components of each register.

Figure 10. Device Tab Main Region Areas

6.2.1 Register Map Table

The region labeled '1' in Figure 10 shows the register map of the device. Each register and its data are shown in the table. The columns of the register map table, as shown in Figure 11 are:

1. Register address
2. Register name
3. Current register value (may require refresh)
4. Register default value (reset register value)
5. Bit field to show the bit representation of the current register value

Figure 11. Register Map Table Areas

Selecting a register in the register map table displays the register bit controls (Section 6.2.2) for the register and provides a detailed view or summary of the register (Section 6.2.5).

Register bits that are not used by the device are highlighted darker gray for reference purposes. By hovering the mouse over a control in the *Register Controls* area, the corresponding bits in the *Register Map* table bit field are highlighted yellow for reference purposes.

6.2.2 Register Controls

When a register is selected in the *Register Map* table, the *Register Controls* area (the region labeled '2' in Figure 10) displays a drop-down control for each register component that can be written or changed. The control is labeled with the name of the register group as specified in the device datasheet. The control has a value for each setting that is a valid configuration. Changing the control to the desired value results in a write register command that updates the register contents.

6.2.3 Hardware Controls

If the EVM under evaluation has GPIO-controlled hardware settings, the **Hardware Controls** area is displayed in the region labeled '3' in Figure 10. These controls function similar to the register controls in the *Register Controls* area. Any changes to these controls sends the appropriate command to the hardware. Refer to the individual EVM user guides for details about the specific commands and functions that are available.

NOTE: The hardware controls do not represent actual commands for the device. These are added functionality for the purposes of the evaluation module.

The hardware controls are not updated or refreshed when the **Refresh/Sync** or **Write Defaults** buttons are pressed.

6.2.4   Refresh/Sync and Write Defaults Controls

The **Refresh/Sync** button sends a command to read all the device registers and update the register map table. These controls are located in the region labeled '4' in Figure 10.

The **Write Defaults** button restores the device registers to their default device values (default value shown in the register map). This is accomplished via register writes and does not use the device reset pins or send a device reset command (if supported).

6.2.5   Register Decode and Information

This region labeled '5' in Figure 10, provides detailed information and breakdown of the current register settings. Available information is displayed in Figure 12 and includes:

(1) Register name - short and long name
(1) Register address
(3) Register description
(2) Register current and default values

List of register groups and components:
(4) Group/component read or write indication
(5) Group/component name - short and long name
(6) Group/component MSB and LSB values
(7) Group/component description
(8) Group/component current setting and description
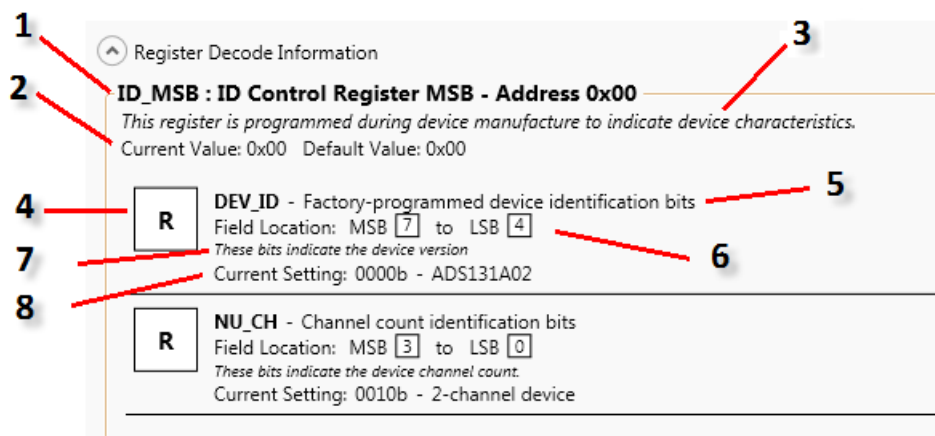
Figure 12. Register Decode Areas

6.2.6   Synchronous Manual Write

Under default operation, any changes to registers via the *Register Controls* area triggers an immediate write register command. By selecting the manual update option in the **File > Options** menu, this button is enabled. When this mode is enabled, changes to register controls do not trigger a register write. Instead, this button must be clicked to submit a manual register write. The write operation is only performed on the register selected in the register table.

NOTE: The register map table will update with the change in register map control. However, a write command in not sent until commanded by the user via this button. Selecting the **Refresh/Sync** button before issuing the manual write will sync the register map and controls to the current value of the device.

---

7 Script Tab

The Script tab provides the capability to create, edit, and execute script files. A script file contains a list of commands to send to the device for configuration. Scripts can be used to initialize a device, test a specific sequence of commands, or other actions.

Scripts can be created and edited through this interface for custom needs. Device packages may contain factory-created scripts created for example applications or common modes of operation.
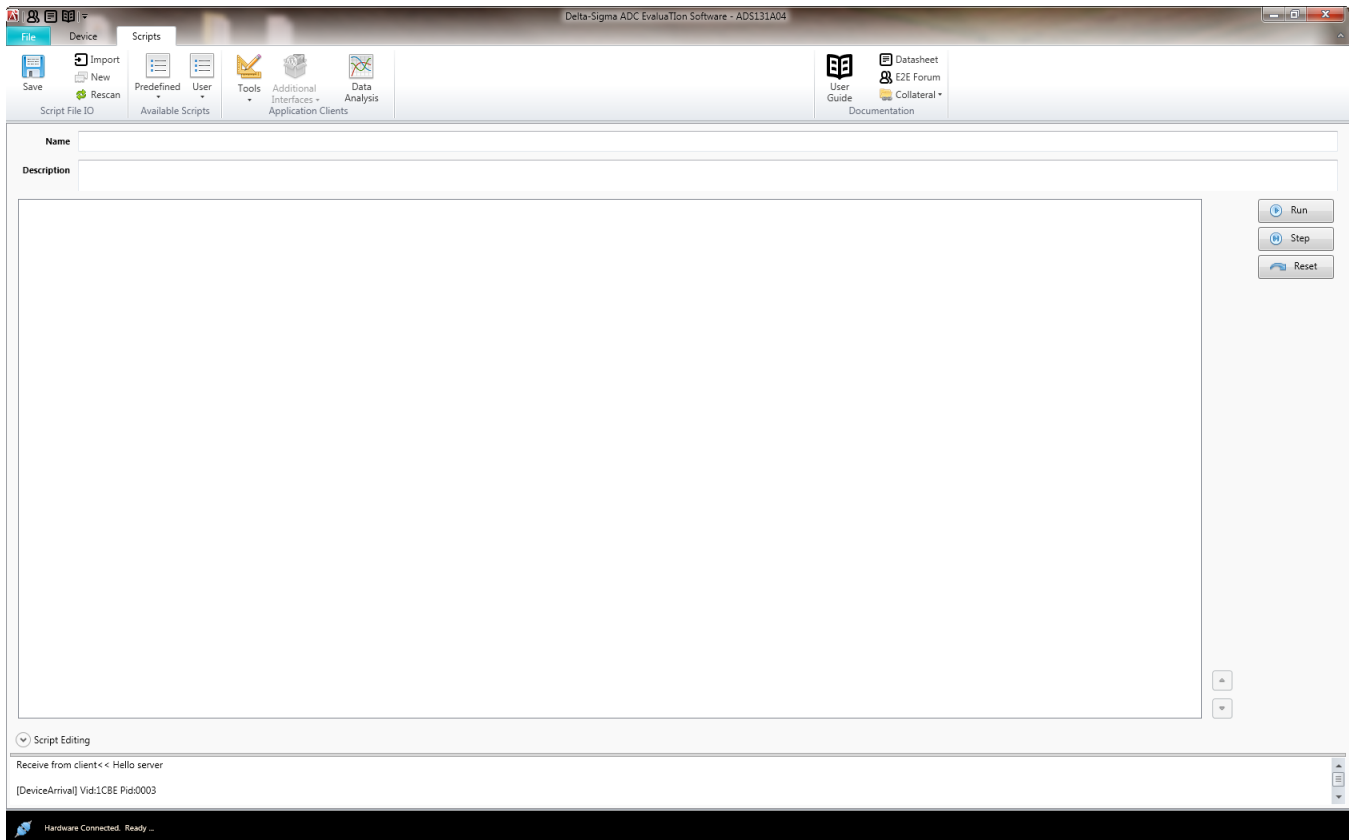
Figure 13. Script Tab Window

7.1 Script Tab Menu

The *Script* tab menu has items specific to the script menu and items in common with the main menu. For more information on scripts, see Appendix B.

7.1.1 Script File IO Menu Group

This Script File IO menu group contains four menu buttons. These buttons and their function are described in this section.

7.1.1.1 Save

The *Save* menu item brings up a save dialog to save the current script contents (name, description, and command list) to the selected file. By default, the script file appears in the *Available Script > User* menu item.

> **NOTE:** TI recommends saving the script to the default location so that it is recognized by the software at startup.
>
> Scripts not saved to the default location must be imported so they are recognized after future application restarts.

7.1.1.2   Import

The *Import* menu item imports a script file from any location on your PC to the user directory to allow the script file to be recognized for future application restarts. Once a script has been imported, it will always be recognized upon future evaluation activities.

7.1.1.3   New

This button clears all of the script fields to allow for a new blank script.

7.1.1.4   Rescan

This button forces the application to rescan the **script** folder and repopulate the available script menu buttons.

7.1.2   Available Scripts Menu Group

This menu group provides access to any scripts, both from the manufacturer or user-written, that may be used for pre-defined sequences to configure the hardware. The menu group contains two drop-down controls: *Pre-Defined* and *User*.

7.1.2.1   Pre-Defined

This menu button provides a list of the scripts that come in the device package. These scripts are provided for reference and may illustrate specific examples or common-use cases.

7.1.2.2   User

This menu button provides a list of the scripts that have been created or edited and saved by the user for reference. Any imported scripts are populated into this menu item by default. If scripts do not appear in the user menu as anticipated, check the directory path (see Section 5.3.1.1) and format of the script files (see Appendix B).

7.1.3   Available Clients Menu Group

These items are the same as described in Section 6.1.2.

7.1.4   Documentation Menu Group

These items are the same as described in Section 6.1.3.

7.2   Operation

7.2.1   Script Information

The controls along the left side of the window all show information regarding the selected script. Beginning at the top-left corner of the window:

- Name – name of this script
- Description – brief description of what the script does or what function to perform
- Commands – list of the commands within the script

7.2.2 Script Controls

Scripts are executed or run via two different methods: *Run* and *Step*.

7.2.2.1 Run Operation

Clicking the **Run** button sends all of the commands within the script to the hardware. The commands are executed by the main application and the results displayed in the activity log.

The commands are sent in sequence from the current command (highlighted command) to the end of the script or until a PAUSE command is reached.

7.2.2.2 Step Operation

Clicking the **Step** button sends the current command in the script to the hardware and then pauses at the next command. To send the next command, click the **Step** button. The current command is sent and then execution is paused at the next command. The **Step** button can be clicked until the last item in the list is reached and sent, at which point the script sends notification that the end of the script has been reached.

7.2.2.3 Reset Button

The **Reset** button resets the current command index to the beginning of the script. This button can be clicked at any time to reset the script to the beginning of the script to allow **Run** or **Step** operations.

7.2.3 Script Editing

New scripts are created or existing scripts are edited using the integrated script editor found in the Script tab.

To use the script editor, expand the editing area by clicking on the arrow next to **Script Editing**. The tab screen will expand to show the editing controls as shown in Figure 14. Once the tab expands, click the *Enable script editing* checkbox as described in Section 7.2.3.1.

NOTE: Any scripts created or edited should be saved using the **Save** menu item to save them to the PC for use at a later time. Any changes not saved are discarded when the program exits.
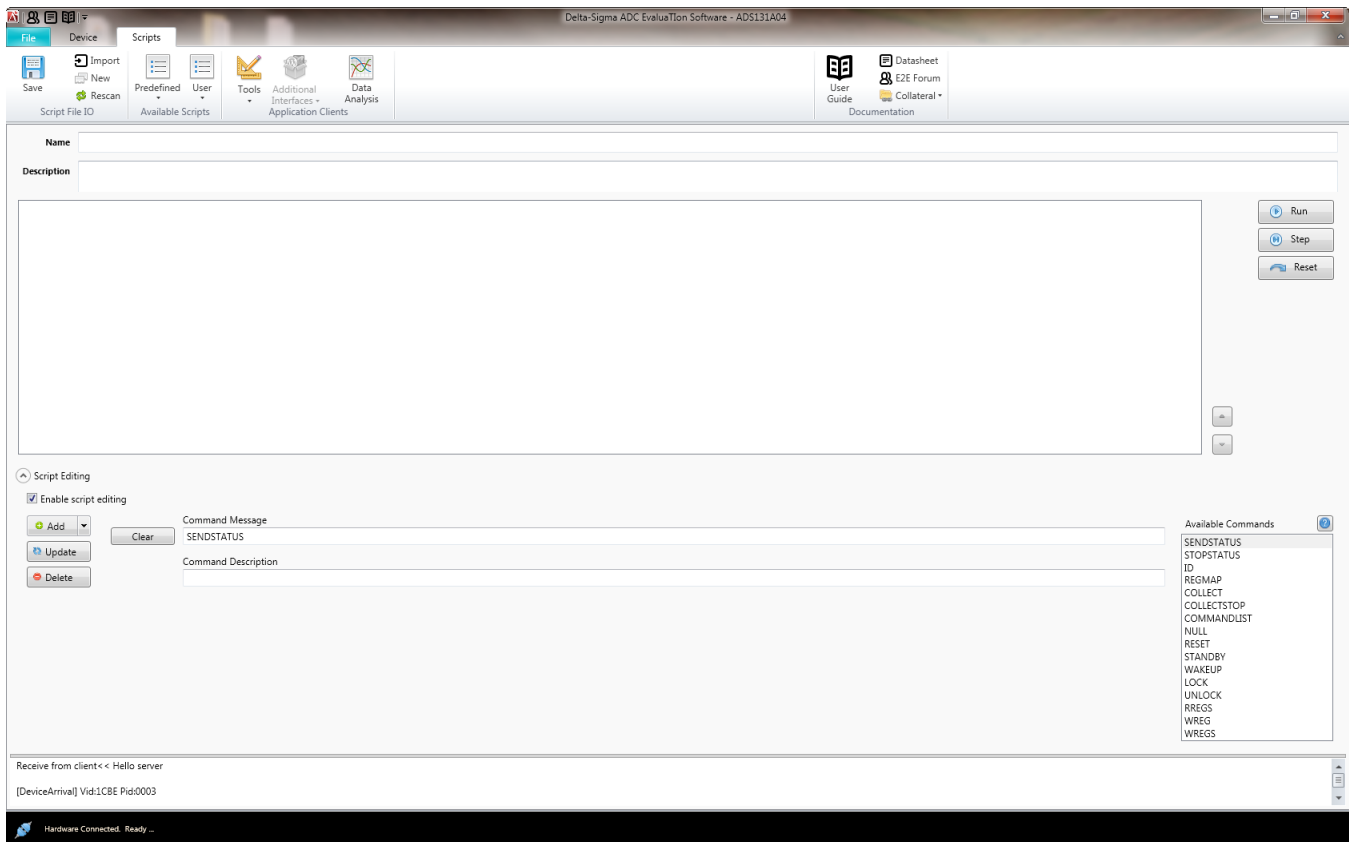

Figure 14. Script Tab Showing Script Editing Capabilities

7.2.3.1 Enabling Script Editing

This checkbox allows user editing of the current script using the editor controls in the expanded view.

7.2.3.2 Arrow Buttons

The arrow buttons located just outside the lower-right corner of the script command list allow the user to move the selected script command within the script order. Select a script command, then click the up or down arrow to change the order of the command operation by one position per click. The script command list window reflects the position of the command as the actions are taken.

7.2.3.3 Available Commands

This control shows a list of the commands that are available for use in the script. Selecting an item in this list populates the **Command Message** control with the selected command and corresponding help.

> **NOTE:** If the command requires additional arguments, they need to be added to the **Command Message** control after the command is selected.

7.2.3.4 Available Command Help

The small button located directly above and to the left of the **Available Commands** list opens a browser window that provides more detail about the commands.

7.2.3.5 Command Message Field

This control displays the command message of the current selected command for editing or the command syntax to be created.

7.2.3.6 Command Description Field

This control displays the command description of the current selected command for editing or the command description to be created.

7.2.3.7 Clear Button

This button clears the *Command Message* and *Command Description* fields.

7.2.3.8 Add New Button

This button adds a command to the bottom of the script. The button adds commands in three different methods.

1. Clicking the **Add New** button on the left side, a command containing the contents of the *Command Message* and *Command Description* fields is added to the last line of the command list.
2. By selecting the drop-down menu for the button and choosing **Delay**, a delay command is added as the last command in the command list. After adding the *DELAY* command, select **DELAY** in the command list and update the delay (in milliseconds).
3. By selecting the drop-down menu for the button and choosing **Pause**, a pause command is added as the last command in the command list. The *PAUSE* command pauses script execution at this point in the script and waits for user input command to continue execution, either by Run or Step (See Section 7.2.2.1 and Section 7.2.2.2).

7.2.3.9 Update Button

This button updates the currently selected command with the contents of the *Command Message* and *Command Description* fields.

7.2.3.10 Delete Button

This button deletes the currently selected command from the script list. If the command was not the last command in the script, all remaining commands are moved up in the command list.

7.2.4 Activity Log

At the bottom of the window, a smaller version of the Activity Log described in Section 8.2.2 is displayed to monitor communications.

8 Console Tab

The Console tab provides a command console for manually entering and sending commands to any hardware connected. This tab displays a log of all transactions, to the hardware, from the hardware, and various other messages that may be relevant during evaluation.

NOTE: The Console tab is hidden by default. If you wish to view this tab and contents, enable the tab in the *Options > General* menu.
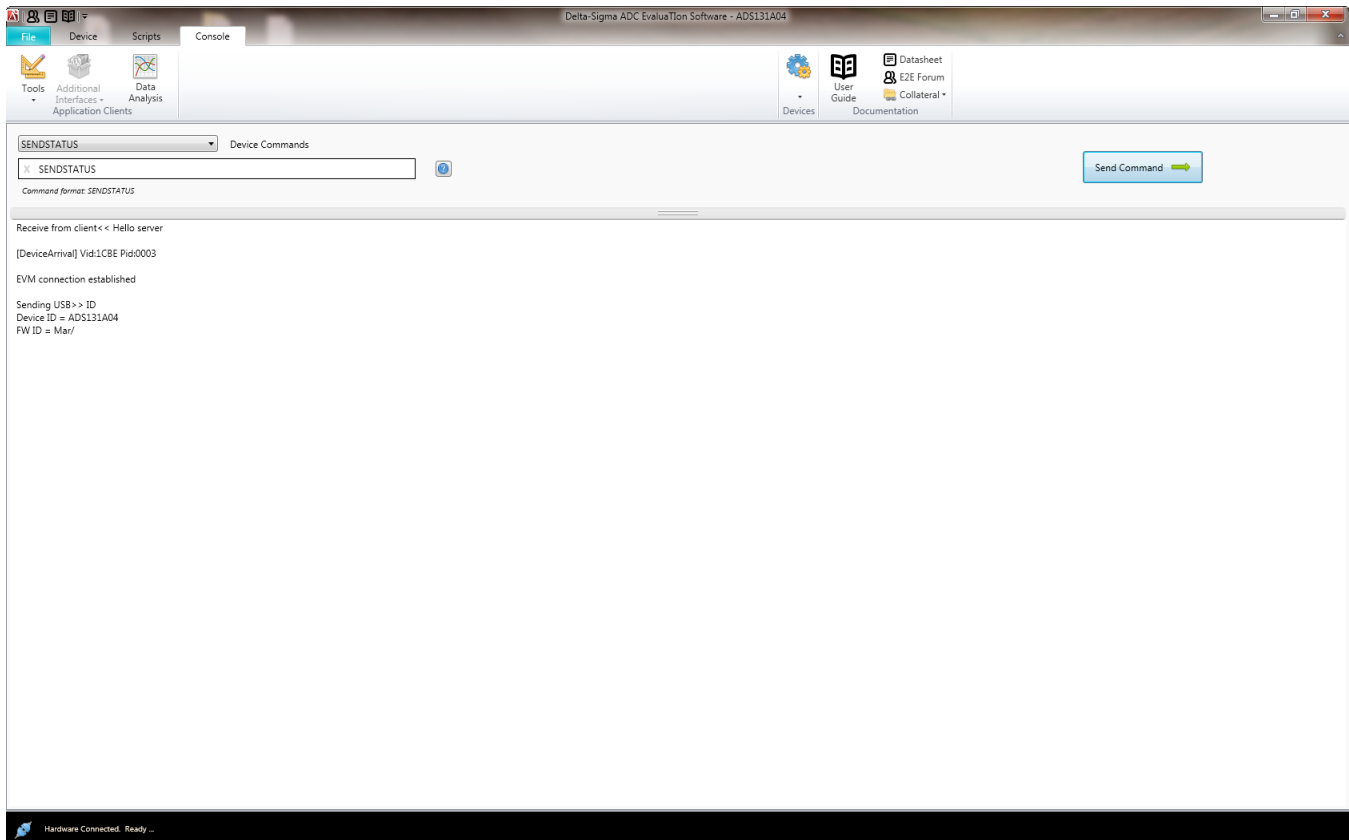

Figure 15. Console Tab Window

8.1 Console Tab Menu

8.1.1 Available Clients Group

These items are the same as described in Section 6.1.2.

8.1.2 Devices

The Devices menu item is used to display a list of all the devices the application recognizes. By selecting a device, the application configures the user interface and load files related to the selected device. This should only be used for demo operation or testing purposes to avoid conflicting configuration information.

The *Devices* folder is rescanned for devices added while the application is running by selecting the **Rescan** button located to the right of the *Devices* menu button.

8.1.3 Documentation Group

These items are the same as described in Section 6.1.3.

8.2 Operation

8.2.1 Command Controls

The command control set consists of:

- Commands support by the device (see the datasheet for details)
- Required application commands supported by the software
- Any additional commands that are recognize by the firmware

8.2.1.1 Device Commands

The available commands for each device are listed in the drop-down control next to the **Command List** label. By selecting a command from the control, it is entered in the **Command to Send** control located directly beneath it.

8.2.1.2 Command to Send Field

Commands can be manually typed into the **Command to Send** control using the keyboard. Command keywords are not case sensitive and arguments should comply with the firmware format requirements.

Some commands require additional arguments, such as register writes or reads. The *Command Syntax* text (directly under the **Command to Send** field) displays the required format information for each command. The command name is followed by any required arguments and description to help ensure the argument is formatted correctly. If no argument symbols are shown, the command does not require any arguments.

The **Send Command** button only enables for recognized, properly formatted commands and allows them to be sent to the hardware. If the command is not properly formatted, a red circle and 'X' displays and the **Send Command** button is disabled. Moving the mouse over the **Command to Send** field displays text indicating the reason for the failed command.

Clicking on the black 'X' to the left of the text field clears the contents of the **Command to Send** field.

The small button to the right of the **Command to Send** field is the command help information. Clicking this button opens a browser window that displays the available commands and command details.

8.2.1.3 Send Command

The entire command in the **Command to Send** field is sent to the hardware when the **Send Command** button is selected. Once the button is selected, the command is sent via USB to the hardware and logged to the activity window. Any replies or responses from the hardware are shown in the log upon receipt of the command.

8.2.2 Activity Log

The activity log is the control that consumes most of the display located in the lower half of the window. This shows any messages, USB activity, and communication that may be of interest. The *Options* menu allows for the customization of message-level display in the activity log. See Section 5.3 for more details.

The activity log has a small context menu that can be accessed by right-clicking on the activity log. Select **Clear Log Screen** or **Clear Log** to perform the desired function to the log. **Clear Log Screen** only clears the contents of the activity log display and does not clear the log history. **Clear Log** clears the entire log history. Once the log is cleared, only log items from this point forward can be viewed using the Master Log Viewer (see Section 10).

9 Data Analysis Tool and Data Acquisition

To collect data from the hardware, the Data Analysis Tool must be started. Select the **Data Analysis** button from any of the application menus to launch the Analysis tool as a separate window.
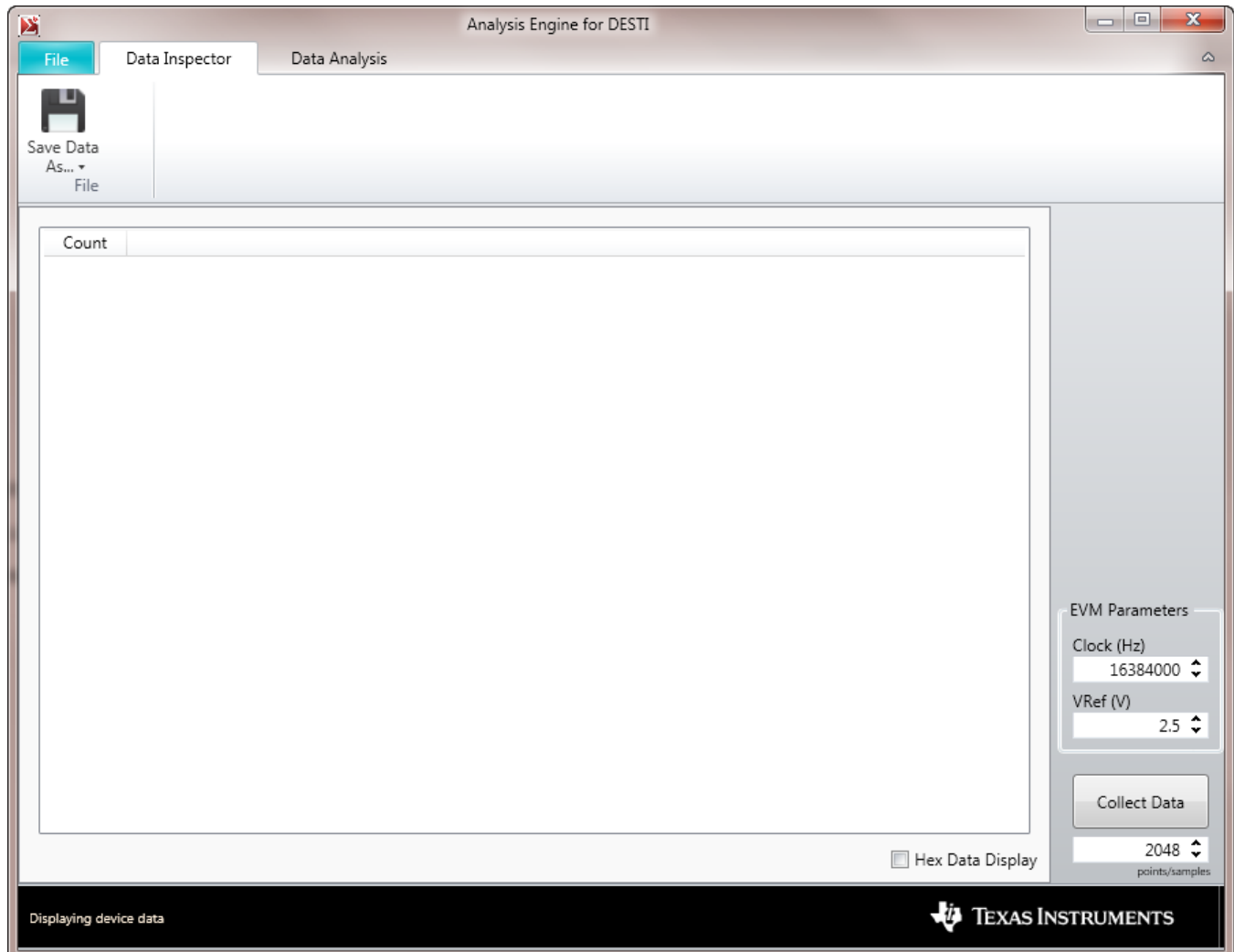

Figure 16. Analysis Tool Window

9.1 Data Acquisition Process

The application allows the collection of ADC data from the EVM for analysis by the tool or external tools (via file export).

9.1.1 Begin the Collection Process

To begin the data collection process:

1. Ensure the device is configured as desired for the data collection. For example, device datarate settings, and channel enables are configured as desired.
2. Ensure the **EVM Parameters** controls accurately reflect the hardware configuration. Any changes made after the data is acquired are not applied to the current data set.
3. Enter the number of points to collect in the numeric control below the **Collect Data** button.

NOTE: When collecting data, ensure that the estimated time required to collect the data is taken into account for your settings. Large datasets or slow datarates may result in lengthy acquisition times.

4. Press the **Collect Data** button to begin the data collection.

When the collection begins, the colored circles in front of the available channels should all turn red. Additionally, a progress bar should appear on the main application window that shows the data acquisition progress. When the acquisition process is complete, any channels that are selected should begin to transfer data to the analysis tool. The channel status indicators turn yellow to indicate data is being transferred and green when the data is transferred and available for analysis or display.

9.1.2   Viewing the Data

After data has been collected, the data is provided in table format or in any of the available data analysis tools. Data can also be saved to a file for analysis using external tools.

9.2   Data Analysis Tool

The Data Analysis tool is used to collect, analyze, and save data collected from the EVM hardware.

The main window is divided into three main regions: the menu, analysis region (center or main portion of window), and the controls region.

NOTE: Multiple Data Analysis tools can be operational at the same time. This allows any collected data to be analyzed by multiple analysis tools. A new Data Analysis tool is launched every time the **Data Analysis** button is clicked.

9.2.1   General Controls Region

The general controls region of the window, located on the right-hand side of the window, provides the channel selection controls for each device, the *EVM Parameters*, and data collection controls. The controls apply to all view and analyses available with the tool.

9.2.1.1   Channel Selection

The *Channel Selection* checkboxes are automatically populated based on the available channels from the device under evaluation. They are labeled according to the functionality and index of the channel within the data set. By checking the checkbox, the corresponding dataset is loaded for display and analysis. Multiple channels can be viewed simultaneously or independently, depending on your needs. By default, the first channel is always displayed when you collect data for the first time. Subsequent data collections maintain the channel selection as long as the channel arrays and channel parameters (datarate, gain, and so forth) are not changed via registers or hardware settings. If register changes or hardware settings are made that influence the mentioned items, the channel selection array is cleared and the first item in the array is shown by default.

9.2.1.2   EVM Parameters

The *EVM Parameters* controls allow for the manipulation of the ADC master clock and voltage reference. Update the controls prior to data collection; changes after data collection are not applied until the next data acquisition.

NOTE: The *EVM Parameters* controls are provided to allow the software analysis to reflect changes to the default hardware settings. The control values should reflect the setting of the hardware and do not programmatically change the hardware configuration or hardware settings.

The **Clock** field allows for the ADC master clock frequency to be changed. The value should be the clock frequency in Hertz (Hz).

The **Vref** field allows for the ADC voltage reference to be changed. The value should be in Volts (V).

9.2.1.3 Collect Controls

The two controls in the bottom right-hand corner of the window provide control over acquisition of data from the ADC.

The **Collect Data** button beings a new data collection, collecting the number of *points/samples* specified in the numeric field directly below the button.

> **NOTE:** At slow data rates, large data samples may require extended time frames to complete.

9.2.2 File Menu

The *File* menu provides the basic tool functions and options.

9.2.2.1 About

The *About* item shows information regarding the tool name, version, and copyright.

9.2.2.2 Options

The *Options* item provides all available application user options. The options are saved and restored when the tool is restarted.

Data Handling and Presentation allows the user to show the data in 32-bit MSB left-aligned data format. For device data that is less than 32 bits, the data is left-shifted to the MSB position and the lower bits are zero-filled.

View Communications Log allows the user to show the **Log** tab on the main screen.

9.2.3 Data Inspector Tab

The *Data Inspector* tab provides a table that displays the collected data for viewing and saving.
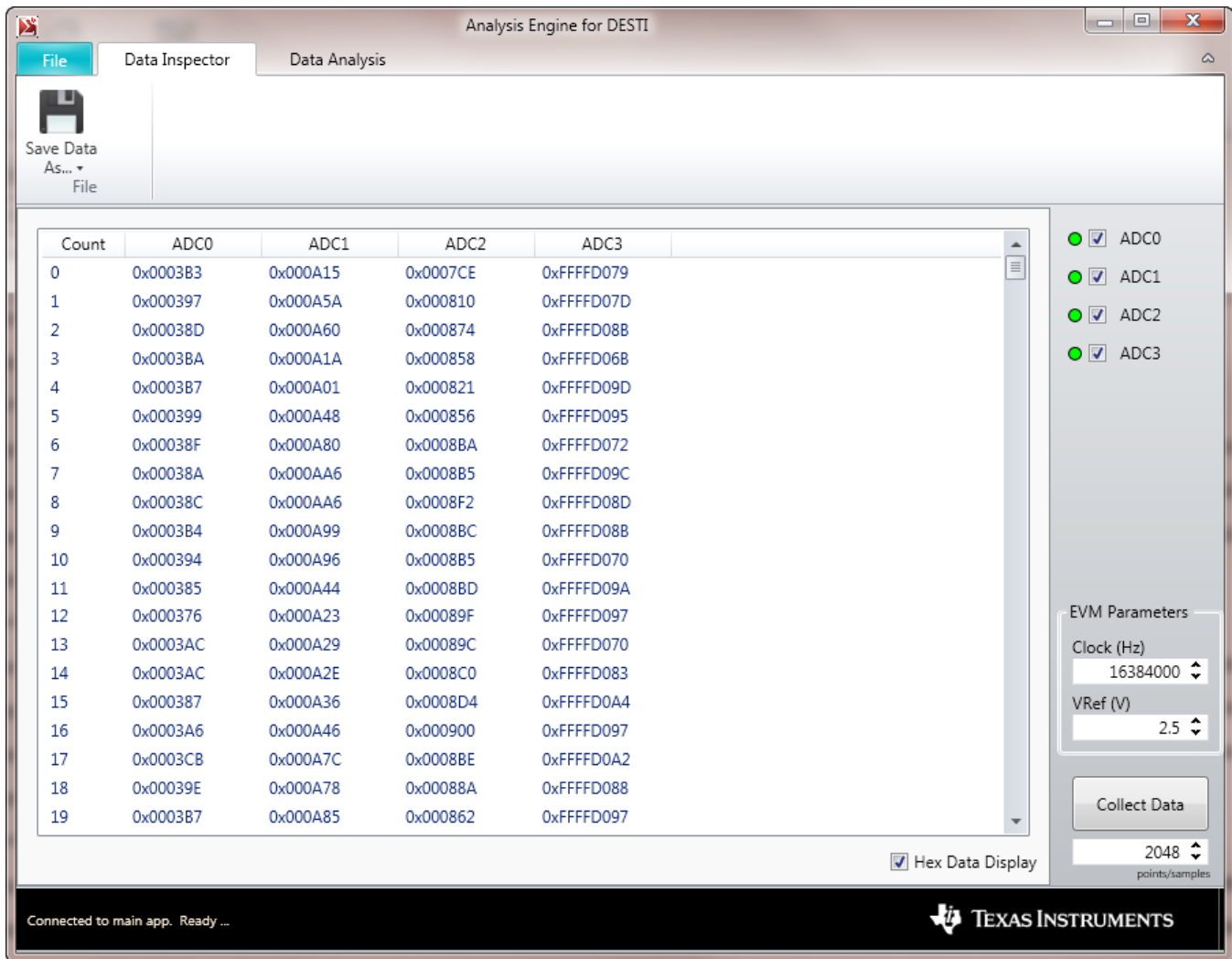
Figure 17. Data Inspector Window

9.2.3.1 Data Display

The table area of the window is where the data is displayed in tabular format. The first column in the table is the index of the data point for reference purposes.

By default, the data is displayed as a decimal value. The **Hex Data Display** checkbox allows the data to be displayed in hexadecimal format.

9.2.3.2 Menu

9.2.3.2.1 Save Data As... Menu Option

The **Save Data As...** menu item allows the user to save the collected data to one of two formats by way of the arrow button at the bottom of the button. The selected data channels (the ones displayed in the table) are saved to the file; non-selected channels are not saved.

- **Standard** format saves the selected channel data to a tab-delimited text file that can be imported in Excel®, MATLAB®, or other tools for later analysis or archiving. The file contains the following information for each channel:
  - Name - name of the channel

- – Datarate - datarate of the device channel in samples per second (SPS)
- – Gain - gain setting for the device channel in V/V
- – FSR - full scale range of the device channel in Volts (V)
- – FSMID - full scale range mid-scale voltage (V)
- – Data (in 32-bit, MSB left-aligned, hexadecimal format)

An example data file is provide in Section D.1.

- **ADCPro™** format saves the selected channel data to a format that is compatible with ADCPro™, our legacy evaluation software. This provides extended analysis features that may not have been implemented yet in this analysis tool. The file format is specified in the *Data Recorder* section of the *ADCPro User Guide* (SBAU128).

9.2.4 Data Analysis Tab

The *Data Analysis* tab provides different analysis options to view the data collected from the ADC EVM under evaluation.

The *Data Analysis* region contains a tabbed menu on the left potion of the window and a graph that consumes most of the window area on the right-hand side of the region. The tabbed-portion of the window allows the selection of the data analysis to perform. Controls and options related to the particular analysis are located on the individual tabs. When an analysis is selected, the graph region changes to reflect the selected analysis.

9.2.4.1 Available Analysis Tools

Current analysis options are:

- Time-Domain Analysis

New analysis tools may be available in the future, check for the latest software version.

9.2.4.2 Time Domain Analysis

The *Time Domain* tab provides the collected data in a graphical format similar to an oscilloscope in the graph window.

Figure 18. Data Analysis - Time Domain Tab Window

9.2.4.2.1 *Time Domain Options*

The *Time Domain* analysis provides several options to manipulate and change the displayed data.

The **Points to Display** control allows the user to specify the number of points to display on the graph. This number can be less than the total points collected to allow the user to look at a smaller portion of the data. If the number is larger than the points specified to collect, all the points collected are displayed. The points are always displayed beginning from the first data point in the array.

The **Display value as:** control allows for the data to be displayed as:

• Volts (Input Referred)

• Volts

• Codes

Note that the **Volts (Input Referred)** option is provided for data channels that may have gain or digital scaling active. If the gain or digital scaling equals 1, then the two *Volt* options will show the same graph.

9.2.4.2.2 *Time Domain Graph*

The graph window displays the data on a line graph with time as the x-axis and magnitude as the y-axis. The different channel or data sets selected via the checkboxes are shown as different colors on the graph.

9.2.5 Log

The *Log* tab provides a view of the communications log that is taking place between the server and client.

> **NOTE:** This tab is hidden, by default, and can be enabled through the options menu.

10   Master Log Browser

The *Master Log Browser* window allows the user to review all of the commands since startup or from when the master log was last cleared by the user. Figure 19 illustrates the *Master Log Browser* window. This window also allows the creation of scripts based on sent commands and the export of the log to a text file for reference or assistance purposes.
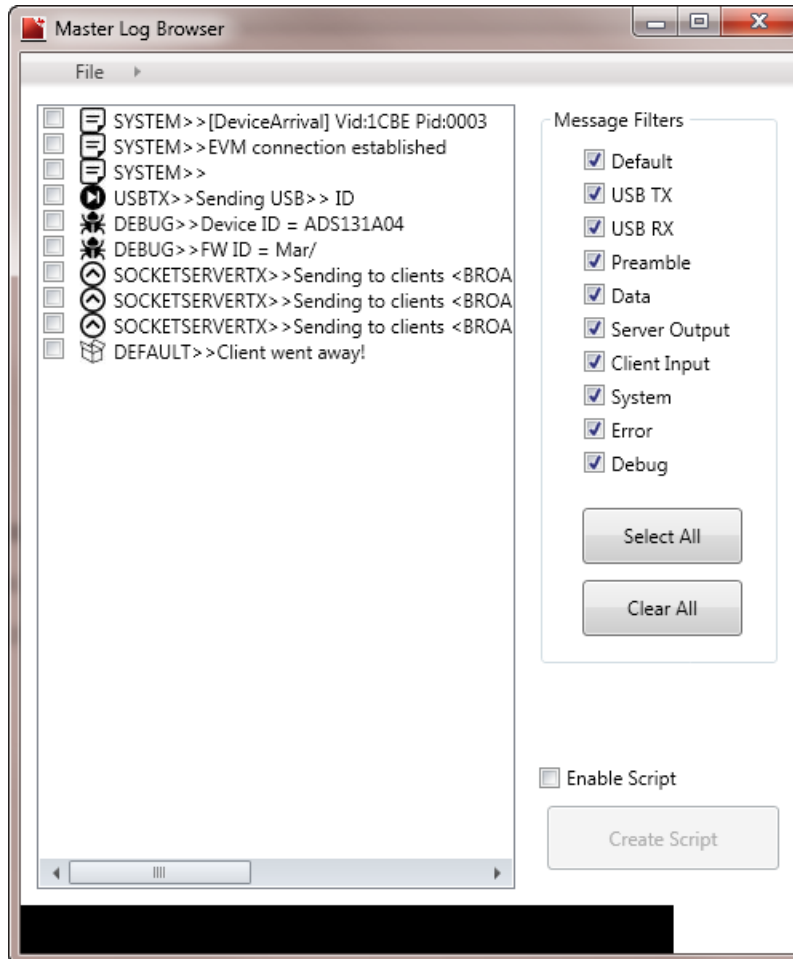
Figure 19. Master Log Browser Window

The *Master Log Browser* window is composed of 4 main areas: browser menu, log item list, log filters, and script creation controls.

10.1   Filtering the Log

The message filters located along the right side of the window allow the user to look at only the specific type of commands that may be of interest. The available filters are enabled by checking the checkbox next to the corresponding filter or disabled by deselecting the checkbox. Two buttons, **Select All** and **Clear All**, provide methods to check or uncheck all of the filter options at once. The log window only displays log entries that match the selected filter options. For example, if you only want to see the commands sent via USB to the hardware, which may be helpful when writing a script, disable all the controls except the **USB TX** control.

10.2   Saving the Log

The log file can be saved to a text file for future reference or to provide for technical support inquiries. The saved file **only** contains the current view (using the current filter options), ensure the desired options are selected.

10.3 Creating Scripts From the Master Log

The *Master Log Browser* allows the user to review commands entered during this session and use them to create a script file for future use.

To create a script file, click the *Enable Script* checkbox. All filters are disabled automatically, except for the *USB TX* option and the **Create Script** button enables. Select the desired commands to be saved in the script file and then click the **Create Script** button. A dialog allows the location and name of the script file to be selected or entered.
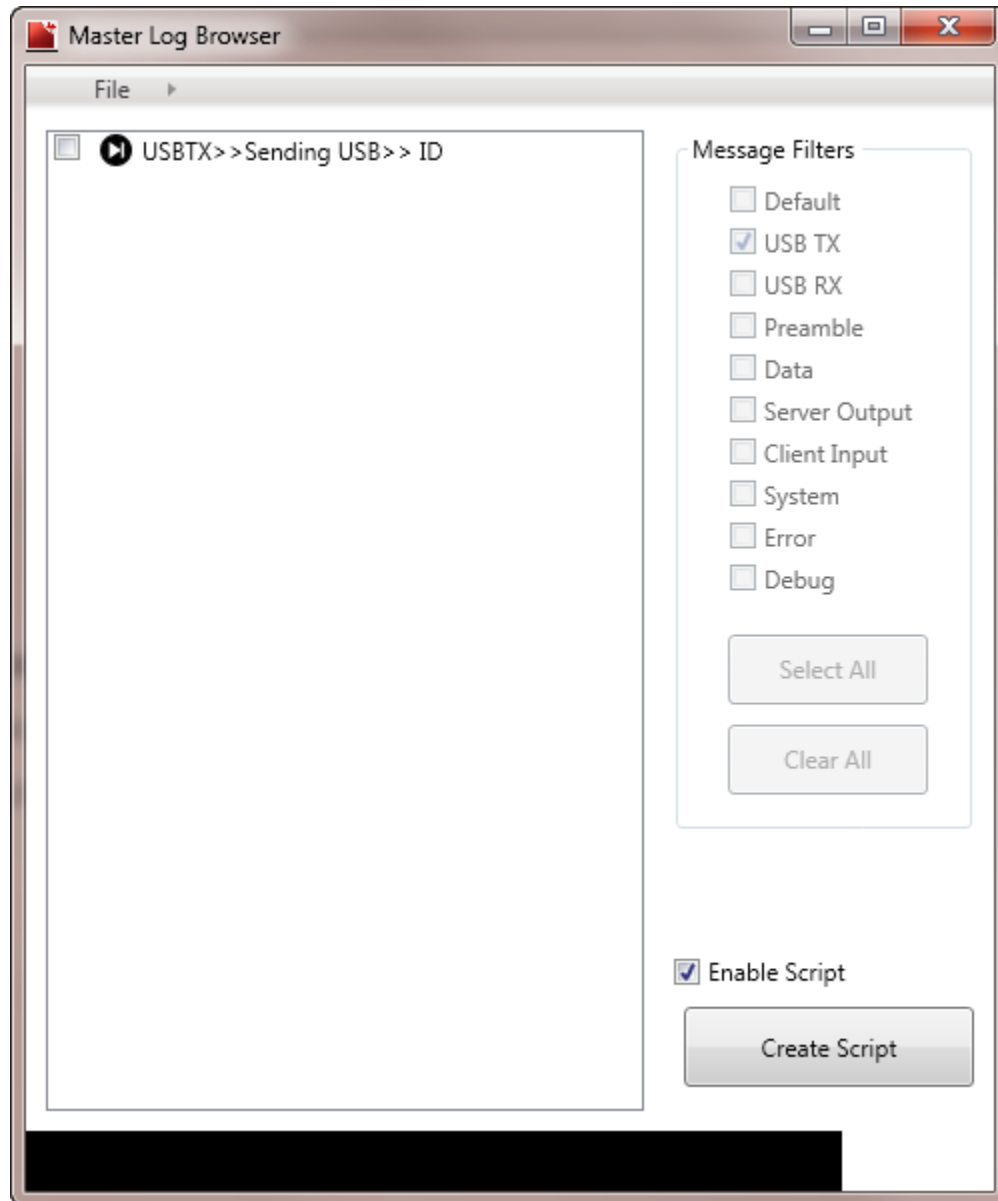
Figure 20. Master Log Browser Creating a Script

The script is saved with a default script name of "log_save_<data/time>" in the user directory. The name of the script and description can be edited using the *Script Editor*. If the script was saved outside the 'user' directory, the script must be imported before the script is recognized. If the script was saved to the default location, the script is recognized.

Frequently Asked Questions (FAQs)

A.1 What if the hardware is not recognized by the software?

If the software does not recognize the hardware when plugged in, verify the following:

1. Are the drivers installed correctly? While the drivers should install automatically, due to varying operating systems and configurations, the driver process may not install correctly. See Section 2.3.2 for manual installation instructions.

2. Verify that the user directory points to the correct location. It should point to the parent directory of the **Devices** folder on the PC. See Section 5.3.1.1 for more details.

A.2 I don't see any way to collect data! How can I do this?

To collect data, open the **Data Analysis** window. See Section 9 for more information on this process.

A.3 Can I manually enter commands in the software?

Yes. The software provides a method to manually enter commands without the aid of the register map or script window. You need to enable the **Console** tab using the *File > Options > Log*. See Section 8 for more information

A.4 Are there any additional analysis available, such as FFT or histogram?

Currently, the only analysis available are:

* *Data Inspector*
* *Time-Domain Analysis*

TI is working to include additional analysis and statistics in future versions, so stay tuned. In the meantime, you can export the data into data files for analysis outside this tool. See Section 9.2.3.2.1 for more information.

A.5 I created a script but it does not appear in the menu item?

If you created a script using the built-in editor:

1. Ensure that you saved the file. If you closed the application without saving the file, it is removed from memory.

2. Ensure that you saved the file to the default directory. By default, this location is *<user_directory>/script/user*. If the file was not saved to this location, use the **Import** function located on the **Script** tab to import the script to the correct location. See Section 7.1.1.2 for more information.

If a script is created using a text editor, import the file into the application using the functionality described in Section 7.1.1.2.

Scripts

B.1 Device Script Generic Rules

Scripts are lists of commands that can be imported and executed in the software in rapid succession to configure a device for a certain configuration.

1. Script commands are executed as written, ensure each command is valid.

2. *DELAY* commands temporarily delay execution after a script command is read. Most commands have a built-in delay but additional delay may be required, based on the script.

3. *PAUSE* commands block execution until manually resumed to allow time to perform work between commands.

4. Use *COLLECT* commands with caution, they cannot be used to perform continuous data collection. A *COLLECT* command should not follow another *COLLECT* command in a script unless:

    1. **Always** use the step command and allow the software to complete one *COLLECT* command before beginning the next.

    2. *COLLECT* commands **must** be separated with a *PAUSE* command to allow *COLLECT* commands to complete before beginning the next.

B.2 Device Script Node Description

Script files are text files using XML-formatted text. The XML nodes follow:

- **script** – root node of the XML file
- **name** – name of script that is shown in the script tab *Name* field
- **description** – description of script that is shown in the script tab *Description* field
- **command list** – list of commands in the script. The commands are executed from top (first) to bottom (last)
- **command** – one command in the command list. This should be a command recognized by the firmware and properly formatted. Each command consists of two nodes:
    - **command_string** – command string to be sent. This is formatted in the proper command syntax.
    - **description** – description of the command or commands settings

B.3 Device Script Example

This is an example script file.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<script>
  <name>Basic startup</name>
  <description>This script gets the device up and running, reads back all the registers, and
enables all the ADCs (for data collection)</description>
  <command_list>
     <command>
          <command_string>UNLOCK</command_string>
          <description>Unlock the device</description>
     </command>
     <command>
          <command_string>WAKEUP</command_string>
          <description>Wakeup the device</description>
     </command>
     <command>
          <command_string>WREG 0F 0F</command_string>
          <description>Enables all ADCs</description>
     </command>
     <command>
          <command_string>REGMAP</command_string>
          <description>Reads back all of the user control registers</description>
     </command>
  </command_list>
</script>
```

Register Map Files

C.1 Register Map Files

The register map settings from the software can be saved into two different formats.

C.2 Register Map Settings File

The register map settings are saved in a tab-delimited format. The first column contains the short register name that match the register map for the device. The second column contains the hexadecimal value (2 characters) representing the register value.

```
ID_MSB      00
ID_LSB      00
STAT_1      00
STAT_P      00
STAT_N      00
STAT_S      00
ERROR_CNT   00
STAT_M2     00
A_SYS_CFG   60
D_SYS_CFG   3C
CLK1        08
CLK2        86
ADC_ENA     00
ADC1        00
ADC2        00
ADC3        00
ADC4        00
```

C.3 Register Map Header File

The register map settings file is provided for use when developing your own code. The file can be included in your project to use the settings from the evaluation software.

The first section contains *#defines* that provide constants for:

- total register count
- register address (referenced by device register name appended with "_ADDRESS)
- register value and setting (reference by device register name appended with "_VALUE") (writable registers only)

The second section contains a struct definition and an array of the structs that contain the register address and value for each of the writeable registers:

```
/*       ADS131A0x Register Settings Export       */
/***************************************************************************/
/* This file contains the register map settings stub */

// General defines
#define ADS131A0x_REGISTER_COUNT 17

/* Register #define values (register address and value) */
```

```
/*****************************************************************************/
/* This section contains the defines for register address and register settings */

/* Register address defines  - All registers */
#define ID_MSB_ADDRESS          0x00
#define ID_LSB_ADDRESS          0x01
#define STAT_1_ADDRESS          0x02
#define STAT_P_ADDRESS          0x03
#define STAT_N_ADDRESS          0x04
#define STAT_S_ADDRESS          0x05
#define ERROR_CNT_ADDRESS         0x06
#define STAT_M2_ADDRESS          0x07
#define A_SYS_CFG_ADDRESS          0x0B
#define D_SYS_CFG_ADDRESS          0x0C
#define CLK1_ADDRESS        0x0D
#define CLK2_ADDRESS        0x0E
#define ADC_ENA_ADDRESS          0x0F
#define ADC1_ADDRESS        0x11
#define ADC2_ADDRESS        0x12
#define ADC3_ADDRESS        0x13
#define ADC4_ADDRESS        0x14


/* Register value defines  - Only writeable registers*/
#define A_SYS_CFG_VALUE          0x60
#define D_SYS_CFG_VALUE          0x3C
#define CLK1_VALUE         0x08
#define CLK2_VALUE         0x86
#define ADC_ENA_VALUE          0x00
#define ADC1_VALUE         0x00
#define ADC2_VALUE         0x00
#define ADC3_VALUE         0x00
#define ADC4_VALUE         0x00


/* Register array constant and structure */
/*****************************************************************************/
/* This section contains the structure definition and array of structures containing writable
register settings*/

struct registerData
{
    int address,
    int value
}

const registerData ADS131A0x_Registers[] =
{
    { 0x000B, 0x0060 },          // A_SYS_CFG
    { 0x000C, 0x003C },          // D_SYS_CFG
    { 0x000D, 0x0008 },          // CLK1
    { 0x000E, 0x0086 },          // CLK2
    { 0x000F, 0x0000 },          // ADC_ENA
    { 0x0011, 0x0000 },          // ADC1
    { 0x0012, 0x0000 },          // ADC2
    { 0x0013, 0x0000 },          // ADC3
    { 0x0014, 0x0000 },          // ADC4
};
```

Data File – Standard File Format

D.1 Standard File Example

The following text provides an example of the standard data save file.

The first several lines provide general file information for reference purposes only.

Being with the channel names, the data file is organized by tab-delimited columns. The channel data starts with the critical channel properties and parameters and then lists the channel data.

```
EVM Device Name
ADS131A04
Data and Time
172421Mar16
Notes
This is a test file using default settings and scripts.

Name     ADC0      ADC1      ADC2      ADC3
Datarate    8000.000    8000.000    8000.000    8000.000
Gain    1    1    1    1
FSR    5.000    5.000    5.000    5.000
Offset    0.000    0.000    0.000    0.000
Data
    00000697    00000F7E    00000E3A    00004D2F
    0000069D    00000FF3    00000E78    00004D2C
    0000069D    00000FA7    00000E78    00004D3B
    000006B5    00000F6F    00000E23    00004D2F
    000006BC    00000F03    00000D6E    00004D52
    000006B4    00000EC9    00000CF9    00004D57
    0000069D    00000F08    00000D12    00004D36
    0000069E    00000F03    00000D5A    00004D5C
    000006B6    00000F27    00000D74    00004D2A
    000006BD    00000F40    00000D4A    00004D30
    000006B6    00000ED3    00000DAA    00004D3A
    000006B2    00000ED7    00000DF2    00004D51
    000006D7    00000EC7    00000E22    00004D3E
    000006C3    00000E86    00000E61    00004D54
    000006AA    00000E2A    00000E7D    00004D67
    000006B6    00000E72    00000E76    00004D39
    000006C5    00000ED2    00000E45    00004D2F
    000006BB    00000EE8    00000DFA    00004D2F
    000006A1    00000F10    00000DC1    00004D42
    000006AC    00000F49    00000DBE    00004D34
    000006CD    00000F53    00000D8E    00004D22
    000006C8    00000F54    00000DB3    00004D1F
    000006C6    00000F8E    00000DBF    00004D0E
    000006B4    00000F79    00000E27    00004CF8
       .
       .
       .
```