

# FlashPro Express User Guide

## Software v11.9

NOTE: PDF files are intended to be viewed on the printed page; links and cross-references in this PDF file may point to external files and generate an error when clicked. **View the online help included with software to enable all linked content.**



**Microsemi Corporate  
Headquarters**

One Enterprise, Aliso Viejo,  
CA 92656 USA  
Within the USA: +1 (800) 713-4113  
Outside the USA: +1 (949) 380-6100  
Fax: +1 (949) 215-4996  
Email:  
[sales.support@microsemi.com](mailto:sales.support@microsemi.com)  
[www.microsemi.com](http://www.microsemi.com)

©2018 Microsemi Corporation. All rights reserved. Microsemi and the Microsemi logo are registered trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

**About Microsemi**

Microsemi Corporation (NASDAQ: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions; security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, California, and has approximately 4,800 employees globally. Learn more at [www.microsemi.com](http://www.microsemi.com).

5-02-00560-8/08.18

---

# Table of Contents

---

<b>Table of Contents.....</b>	<b>2</b>
<b>Introduction to FlashPro Express .....</b>	<b>4</b>
About FlashPro Express .....	4
Secure Job Programming .....	4
Migrating FlashPro Projects to FlashPro Express .....	4
Supported Families - FlashPro Express .....	5
Installing FlashPro Express Software and Hardware .....	6
<b>Getting Started.....</b>	<b>7</b>
Starting FlashPro Express .....	7
FlashPro Express Interface.....	7
Creating a Job Project from a FlashPro Express Job.....	8
Load a Job Project .....	9
Saving a Job Project .....	10
Programming Tutorials.....	10
<b>Programmer Settings and Operations .....</b>	<b>14</b>
Introduction .....	14
Programmer Settings .....	14
Ping Programmers .....	16
Performing a Self-Test .....	16
Scanning and Checking a Chain.....	16
Enabling and Disabling Programmers .....	16
Renaming a Programmer.....	16
Removing a Programmer.....	16
Selecting and Running an Action.....	16
<b>Chain Programming.....</b>	<b>18</b>
Chain Order.....	18
Multiple Device Chain Programming.....	18
<b>TCL Commands - FlashPro Express .....</b>	<b>20</b>
About TCL Commands - FlashPro Express Tcl Command Reference .....	20
Running Tcl Scripts from within FlashPro Express.....	21
Running Tcl Scripts from the Command Line .....	21
Exporting Tcl Scripts from within FlashPro Express .....	22
Tcl Commands Supported by FlashPro Express.....	23
<b>Troubleshooting .....</b>	<b>39</b>
Exit Codes (SmartFusion2 and IGLOO2) .....	39

Exit Codes for Software v8.6 and Above (SmartFusion, IGLOO, ProASIC3 and Fusion).....	45
Exit Codes for pre-v8.6 Software (SmartFusion, IGLOO, ProASIC3 and Fusion) .....	61
Exit Codes (ProASICPLUS).....	67
<b>SmartDebug .....</b>	<b>74</b>
<b>Electrical Parameters .....</b>	<b>75</b>
DC Characteristics for FlashPro5/4/3/3X.....	75
DC Characteristics for FlashPro Lite.....	76
DC Characteristics for FlashPro .....	77
<b>Electrical Specifications.....</b>	<b>79</b>
FlashPro5.....	79
<b>Electrical Specifications.....</b>	<b>81</b>
FlashPro4.....	81
FlashPro3.....	82
FlashPro Lite.....	83
FlashPro.....	84
JTAG Switching Characteristics.....	85
<b>FlashPro Express Reference .....</b>	<b>87</b>
FlashPro Express Start Page.....	87
FlashPro Express Project Menu .....	87
FlashPro Express Edit Menu .....	88
FlashPro Express View Menu.....	88
FlashPro Express Tools Menu.....	88
FlashPro Express Help Menu .....	89
FlashPro Express Log Window and Status Bar.....	89
<b>Contacting Microsemi .....</b>	<b>90</b>
Microsemi SoC Products Group Headquarters .....	90
Contact Information.....	90
Regulatory and Compliance Information.....	90

---

# Introduction to FlashPro Express

---

## About FlashPro Express

FlashPro Express is Microsemi's programming tool designed from the ground up to address secured programming assurance in production programming house environments. FlashPro Express supports SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion, and ProASICPLUS devices in the Windows OS environments and supports SmartFusion2, IGLOO2, and RTG4 in the Linux OS environments.

You can install FlashPro Express two ways:

- **Integrated with Libero** - FlashPro Express is installed automatically when Libero is installed. FlashPro Express is used by Libero to perform the programming tasks for SmartFusion2, IGLOO2, and RTG4, as part of the design flow.
- **Stand-Alone** - FlashPro Express is also available as a standalone installation. This mode is primarily used for production programming or lab programming on machines in which a full version of Libero is not required. When installed using the stand alone installation, FlashPro Express is joined by the predecessor product FlashPro.

See [the Microsemi Website for more information on FlashPro and FlashPro Express](#).

## Secure Job Programming

Job programming is the concept of using a single file to program a Microsemi device or chain of Microsemi devices using encrypted bitstreams.

The single job file contains all of the information necessary to setup FlashPro Express as well as the encrypted bitstream images for the devices in the job. Once a job file is created it can be handed off securely to production programming houses or contract engineering facilities to load the Microsemi images during manufacturing. Job projects can be exported from Libero and imported into stand-alone FlashPro Express providing a clean delineation between design flow and production programming.

## Migrating FlashPro Projects to FlashPro Express

Existing FlashPro projects (\*.pro) files are now called Job Project files in FlashPro Express. These Job Projects can be opened with FlashPro Express to take advantage of Linux programming support and the simplified tool targeted for operators in a production floor environment.

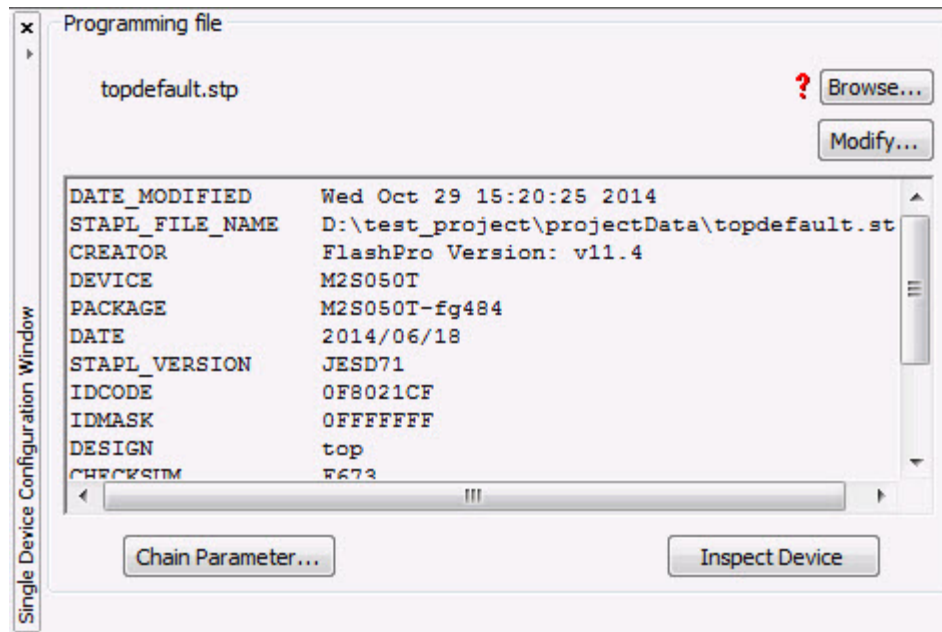
FlashPro projects that were created in single mode will not be supported with this tool. Microsemi recommends that you convert these projects to chain mode projects. To convert the project to a chain project, do the following steps:


1. Open the FlashPro project (\*.pro) in FlashPro.
2. Locate the loaded STAPL file by one of two methods:

The log will print "STAPL file '<stapl\_path>' has been loaded successfully." <stapl\_path> is the location of the STAPL file loaded.



Within the Single Device Configuration Window there is a field STAPL\_FILE\_NAME, which displays the location of the STAPL file loaded.



3. Switch the project to chain mode by one of the two methods:
  - a. Press the chain button from the toolbar.
- 
  - b. From the Tools menu, select **Mode->Chain Programming**.
4. Load the STAPL file in chain mode by adding a Microsemi device in the chain.
  - a. From the File menu, select **Configuration->Add Microsemi Devices from Files**.
  - b. Browse to the location of the STAPL file and click **Open**.
5. To save the project, from the File menu, select **Save Project**.
6. You may now open the project using FlashPro Express.

When moving FlashPro project (\*.pro) files to another machine, Microsemi recommends that you archive the entire project folder, copy it to the new machine, extract it locally, then load the job project within FlashPro Express. FlashPro Express will only open a job project if a programmer is connected to the machine, at least one Microsemi device has programmed enabled, and all enabled Microsemi devices have a bitstream file loaded.

## Supported Families - FlashPro Express

The table below lists the device families and their derivatives that FlashPro Express can program directly through Libero or by exporting a FlashPro Express job:

Table 1 · Product Families and Derivatives Directly Supported by FlashPro Express

Device Family	Family Derivatives	Description
<a href="#">SmartFusion2</a>	N/A	Address fundamental requirements for advanced security, high reliability and low power in critical industrial, military, aviation, communications and medical applications.
<a href="#">IGLOO2</a>	N/A	Low-power mixed-signal programmable solution
<a href="#">RTG4</a>	N/A	Radiation-tolerant programmable solution

In addition, FlashPro Express supports the programming of all the following Microsemi devices in a JTAG chain by one of the following methods:

- [A FlashPro Project that was migrated to a FlashPro Express Project](#) .

OR

- A FlashPro Express Project for one of the primary devices (above) that includes other devices in the JTAG chain from the list of devices below.

Table 2 · Product Families and Derivatives Supported in the JTAG chain

Device Family	Family Derivatives	Description
<a href="#">SmartFusion</a>	N/A	SmartFusion intelligent mixed-signal FPGAs are the only devices that integrate an FPGA, ARM Cortex-M3, and programmable analog, offering full customization and IP protection.
<a href="#">Fusion</a>	N/A	Mixed-signal FPGA integrating ProASIC3 FPGA fabric, programmable analog block, support for ARM® Cortex™-M1 soft processors, and flash memory into a monolithic device.
<a href="#">IGLOO</a>	IGLOO	The ultra-low-power, programmable solution
	IGLOOe	Higher density IGLOO FPGAs with six PLLs and additional I/O standards
	IGLOO nano	The industry's lowest power, smallest size solution
	IGLOO PLUS	The low-power FPGA with enhanced I/O capabilities
<a href="#">ProASIC3</a>	ProASIC3	The low-power, low-cost, FPGA solution
	ProASIC3E	Higher density ProASIC3 FPGAs with six PLLs and additional I/O standards
	ProASIC3 nano	Lowest cost solution with enhanced I/O capabilities
	ProASIC3L	The FPGA that balances low power, performance, and low cost
	Automotive ProASIC3	ProASIC3 FPGAs qualified for automotive applications
	Military ProASIC3/EL	Military temperature A3PE600L, A3P1000, and A3PE3000L
	RT ProASIC3	Radiation-tolerant RT3PE600L and RT3PE3000L
<a href="#">ProASIC PLUS</a>	N/A	Supported by FlashPro Express ONLY- not supported by Libero SoC.

## Installing FlashPro Express Software and Hardware

See [the Microsemi Website](#) for more information about FlashPro Express.

---

# Getting Started

---

## Starting FlashPro Express

Start FlashPro Express by running the program at [installation folder] > bin > FPExpress.

## FlashPro Express Interface

The main FlashPro Express UI consists of a list of programmers and a chain table, as shown in the figure below. This view displays the programmers connected to the machine, and the devices within the JTAG chain specified in the job project file (PRO) file, as shown in the figure below.

- Hover over the programmer Info icon to display more information about a programmer.
- Click the Name field to change a programmer name.
- Click the checkbox to enable or disable a programmer.
- Right-click a programmer to Ping, Self-Test, Scan, Check Chain or Remove it from the list.
- Additional information about a device and programming file, if loaded, can be viewed by hovering over the info icon of that device.
- Devices specified as disabled in the job project (\*.pro) file are shown disabled and their HighZ value is displayed in the column header.
- Device/Programmer States:
  - **IDLE**: The devices/programmers are idle and not executing any programming action.
  - **DISABLED**: Devices that are not enabled for programming
  - **PASSED**: The last programming operation passed
  - **FAILED**: The last programming operation failed



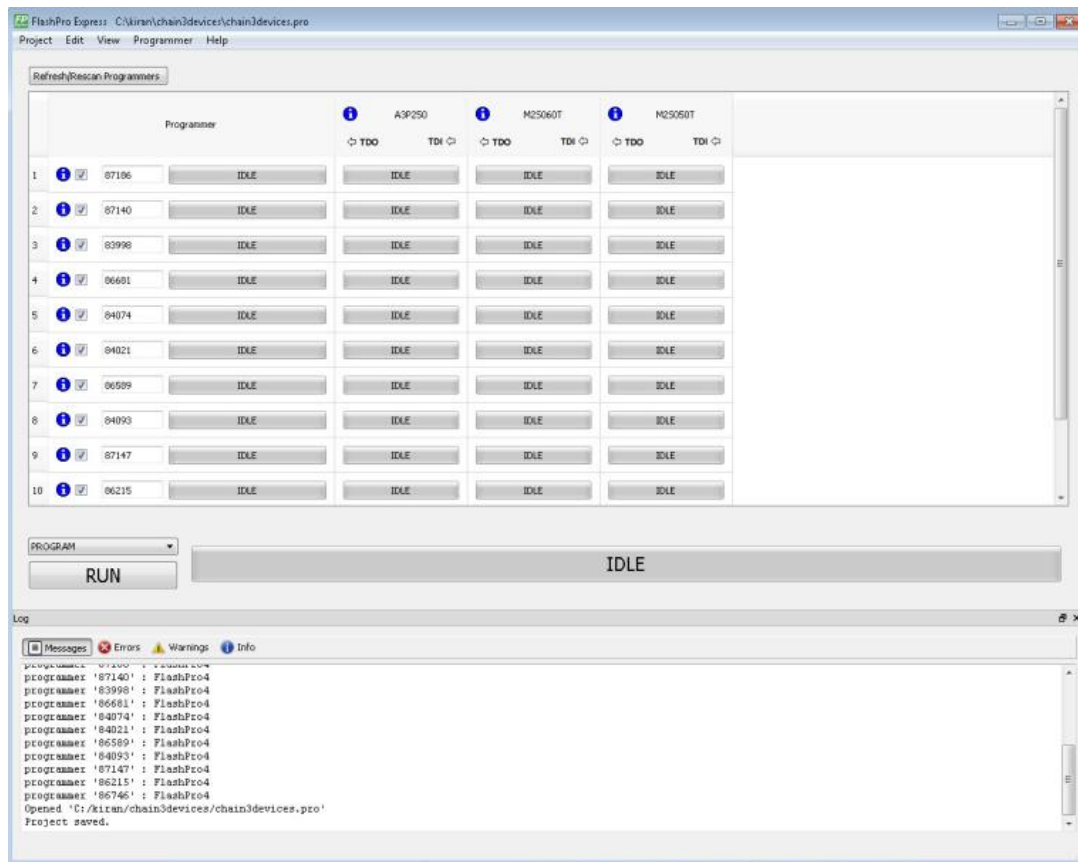


Figure 1 · FlashPro Express Programmers and Chain Table

## Creating a Job Project from a FlashPro Express Job

Once you are ready to hand off your design for production you can create a job project. To do so:

1. In Libero run **Export FlashPro Express Job** to create a container that will be used to transfer programming configuration information, including programming files, to the production programming tool FlashPro Express.

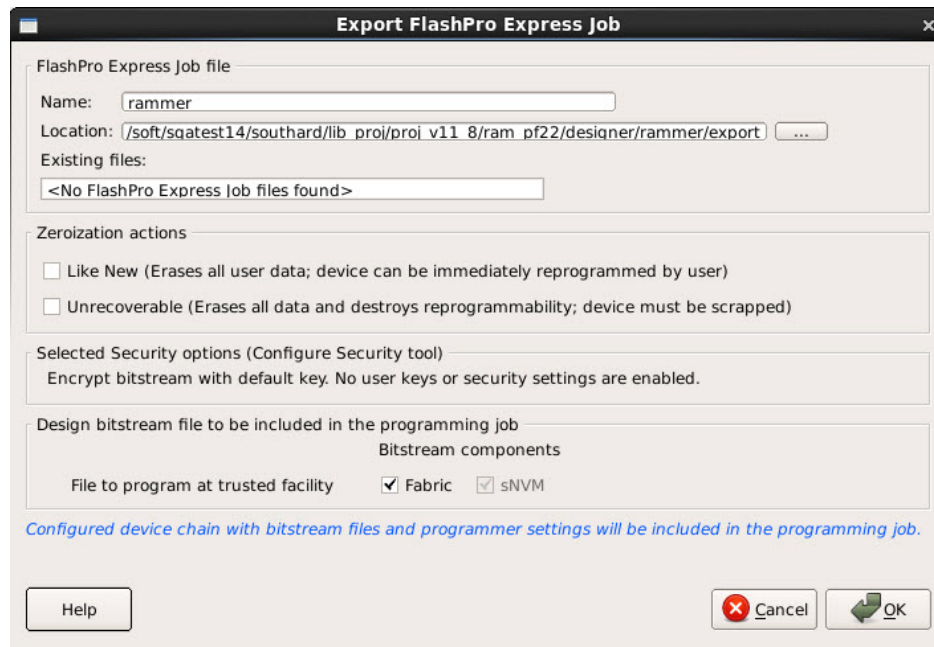


Figure 2 · Export FlashPro Express Job

- In FlashPro Express, from the **Project** menu choose **New Job Project from FlashPro Express Job**. You will be prompted to specify the Programming Job File location that you just exported from Libero and the location of where to store the FlashPro Express Job Project. The Job Project name automatically uses the programming job name and cannot be changed. Click **OK** and a new Job Project will be created and opened for production programming.

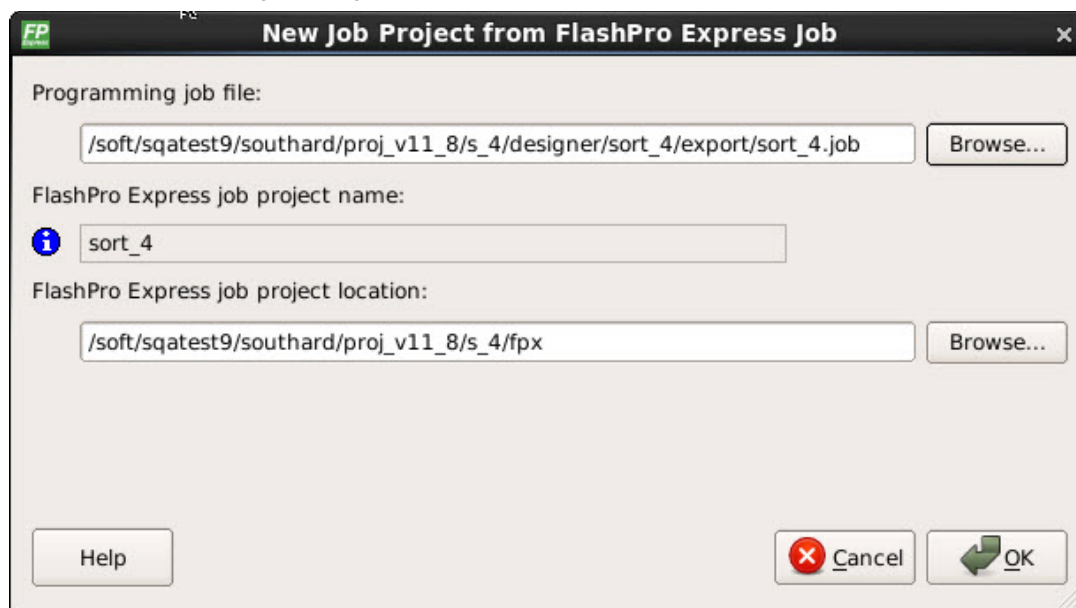


Figure 3 · New Job Project from FlashPro Express Job Dialog Box

## Load a Job Project

To get started in FlashPro Express you must load a job project (\*.pro file). To do so, from the **Project** menu choose **Open Job Project**. A job project will open if:

- At least 1 programmer is connected
- At least 1 Microsemi device is enabled for programming

- Any enabled Microsemi device for programming must have a bitstream file loaded

**Note:** FlashPro projects (\*.pro) created in single chain mode are not supported with this tool. You must create a chain mode project with the existing programming files within FlashPro prior to using FlashPro Express.

**To open a project:**

- From the **Project** menu, choose **Open Job Project**. The **Open Project** dialog box appears.
- Find your project file or type in your project file name in the **File name field**.
- Click **Open**.

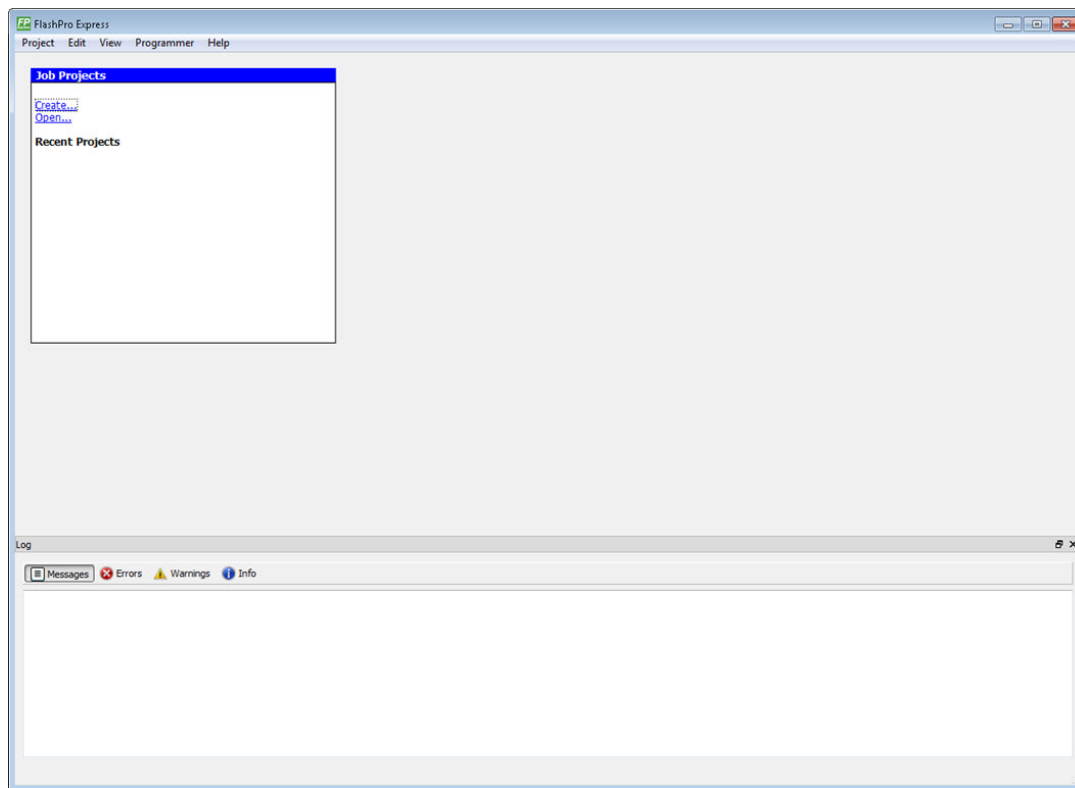


Figure 4 · FlashPro Express Launch Screen

## Saving a Job Project

Click the **Save** button on the toolbar, or from the **Project** menu choose **Save Job Project** to save your project.

## Programming Tutorials

### Parallel Programming with FlashPro5/4/3/3X

Parallel programming enables you to program multiple Microsemi devices in parallel with multiple programmers. In parallel programming, all targeted devices are programmed with the same programming file (STAPL). The targeted device or chain configuration that is connected to each programmer must be identical.

The FlashPro Express software together with the FlashPro5/4/3/3X programmers supports parallel programming via a USB port. You can connect up to sixteen FlashPro5/4/3/3X's to a PC via a USB v1.1 or a USB v2.0 port. FlashPro5/4/3/3X requires a self-powered hub.

Connecting FlashPro5/4/3/3X (a USB v2.0 enabled programmer) to USB v1.1 port increases device programming time due to a slow data transfer rate on the USB v1.1 port in comparison to a USB v2.0 port.

**Note:** FlashPro (USB/LPT1) or FlashPro Lite programmers do not support parallel programming.

The following figure illustrates how you can connect a FlashPro5/4/3/3X programmer for parallel programming.

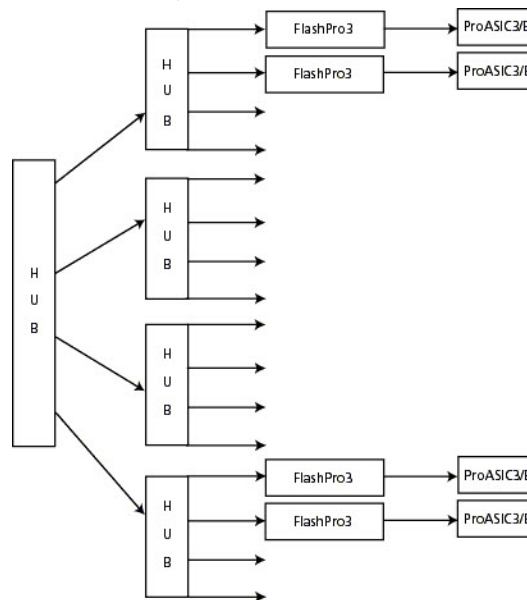


Figure 5 · Connecting a FlashPro5/4/3/3X Programmer

An independent thread processes the STAPL file during parallel programming. In an Microsemi test, parallel programming is approximately five times faster than programming 16 devices sequentially.

**Note:** Microsemi has tested Belkin PCI-USB cards and hubs. We have found that parallel programming works best with the vendor's latest driver installed and with the matching hubs.

## Chain Programming Tutorial

This tutorial demonstrates how to use FlashPro Express to program a multi-device, multi-programmer chain. This tutorial uses the production programming flow that exports a programming job from Libero SoC, which includes chain configuration, programmer settings, and bitstream files for programming, and creates a job project from a programming job.

The figure below shows the chain used in this tutorial. M2S050T is device 1 and A3P250 is device 3. Device 1 is the first device to be programmed in the chain and device 2 is the last; device 3 is disabled and will not be programmed.

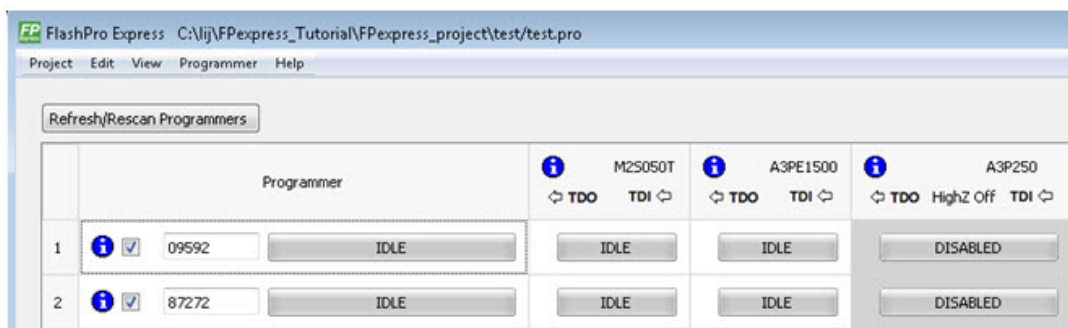


Figure 6 · Chain Programming Devices

### To program a chain:

1. From the **Project** menu, choose **New Job Project from FlashPro Express Job**.
2. Click **Browse** to load a Programming Job File, and specify your **FlashPro Express job project location**. Click **OK** to continue, as shown in the figure below.

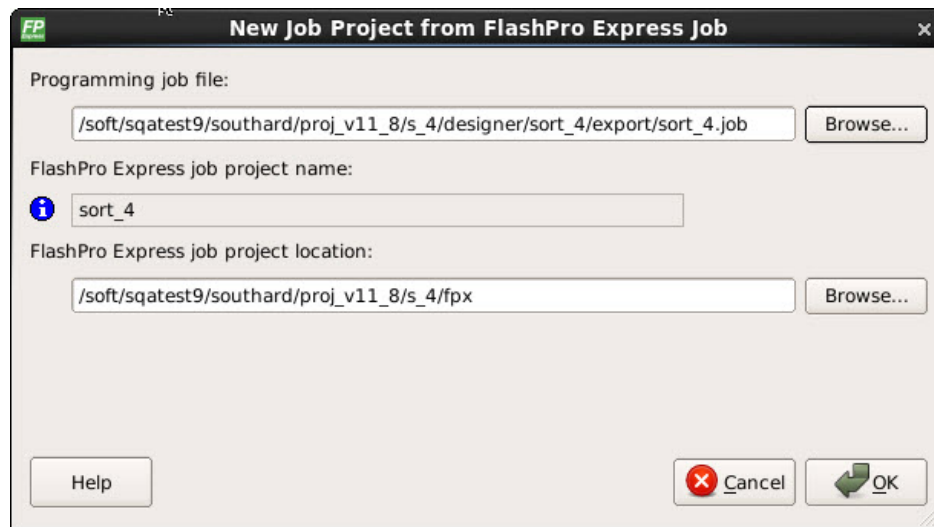


Figure 7 · New Job Project from FlashPro Express Job

FlashPro Express displays your Job Project and programmers, as shown in the figure below. The Device/Programmer states are:

- **IDLE**: The devices/programmers are idle and not executing any programming action
- **DISABLED**: Devices that are not enabled for programming
- **PASSED**: The last programming operation passed
- **FAILED**: The last programming operation failed

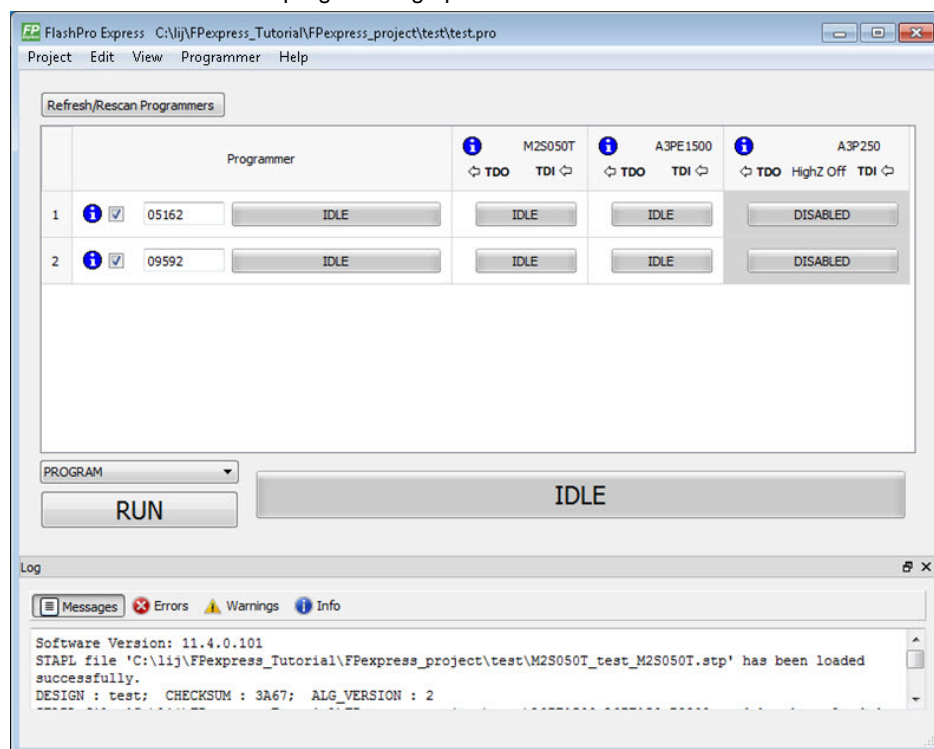


Figure 8 · FlashPro Express with Loaded Job Project

See the Export Programming Job topic for information on how to generate a Programming Job file.

3. Click the **Refresh/Rescan** button if your programmer is not listed. Hover your mouse over the **Info** icon to view device info. If a device is Disabled for programming the HighZ status appears in the GUI, as shown in the figure above.
4. Set the Programming Action in the dropdown menu to **PROGRAM**, as shown in the figure below.

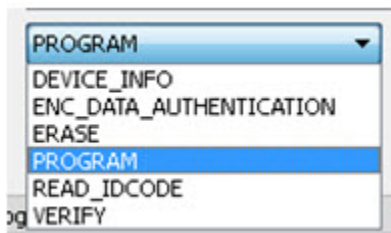


Figure 9 · Programming Action Set to PROGRAM

5. Click **RUN**. Detailed individual programmer and device status information appears in the Programmer List. Your programmer status (PASSED or FAILED) appears in the Programmer Status Bar, as shown in the figure below.
  - Hover over the Programmer Status Bar to display information on the programmers.
  - Hover over the FAILED status to list all programmers that failed programming.
  - Hover over the PASSED status to list all the programmers that programmed successfully.

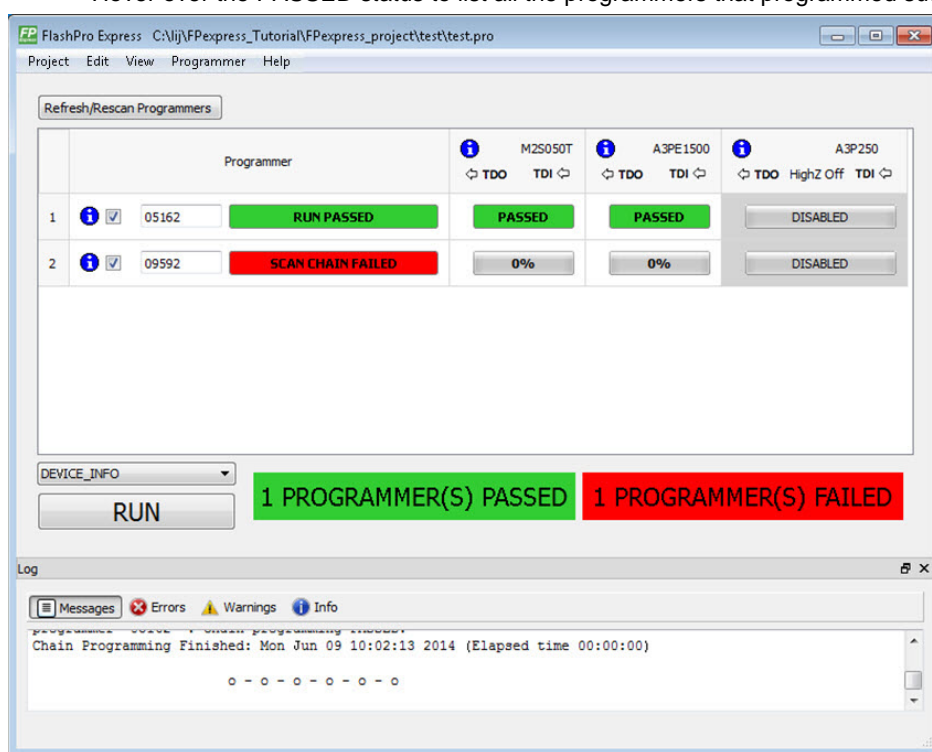


Figure 10 · Chain Programming Complete

View the **Log** for Messages, Errors, Warnings and Info generated during programming.

# Programmer Settings and Operations

## Introduction

The FlashPro Express software enables you to connect multiple programmers to your computer. With each programmer you select, you can connect the programmer, perform a self-test, customize, add, and remove and analyze the JTAG chain, as shown in the figure below.

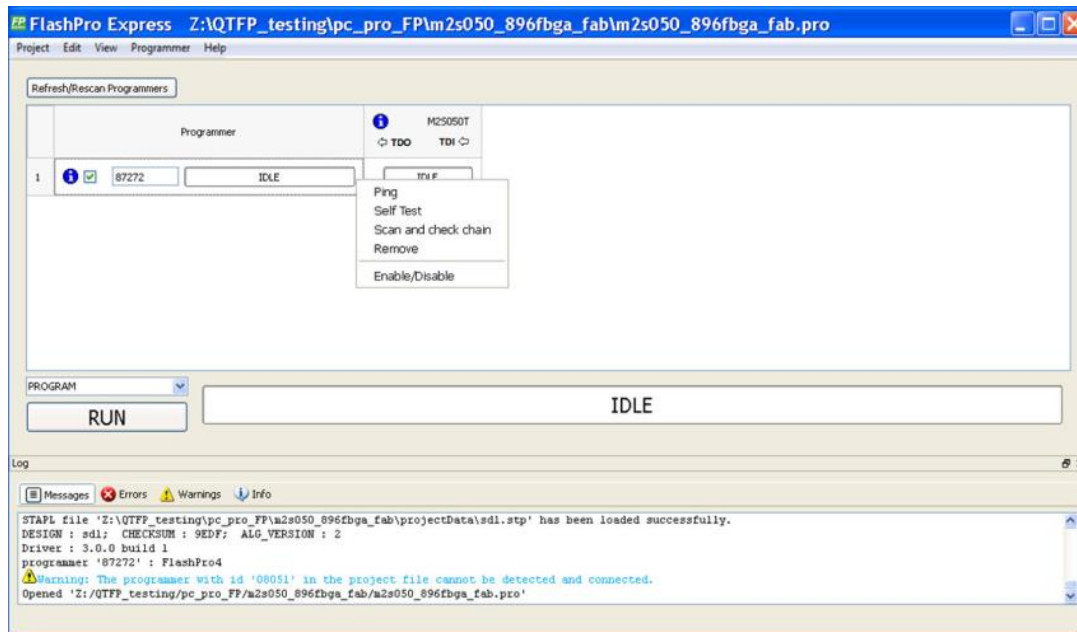


Figure 11 · FlashPro Express Right-Click Menu

## Programmer Settings

The Programmer Settings dialog box includes setting options for FlashPro5/4/3/3X, FlashPro Lite and FlashPro.

**Note:** You can set the TCK setting in the PDB/STAPL file by selecting the TCK frequency in the Programmer Settings dialog box.

Limitation of the TCK frequency for the selected programmer:

- FlashPro supports 1-4 MHz
- FlashPro Lite is limited to 1, 2, or 4 MHz only.
- FlashPro5: 1, 2, 3, 4, 5, 6, 10, 15, 30 MHz
- FlashPro4: 1, 2, 3, 4, 5, 6 MHz
- FlashPro3/3X: 1, 2, 3, 4, 6 MHz

Limitation of the TCK frequency for the target device:

- **IGLOO, ProASIC3, and Fusion – 10MHz to 20MHz**
- ProASICPLUS and ProASIC – 10 MHz.
- SmartFusion2, IGLOO2, and RTG4 - 30MHz

During execution, the frequency set by the FREQUENCY statement in the PDB/STAPL file will override the TCK frequency setting selected by you in the Programmer Settings dialog box unless the **Force TCK Frequency** checkbox is selected.



## FlashPro Programmer Settings

Choose your programmer settings for FlashPro (see figure above). If you choose to add the Force TCK Frequency, select the appropriate MHz frequency. After you have made your selection(s), click **OK**.

### Default Settings

- The Vpp, Vpn, Vdd(I), and Vddp options are checked (Vddp is set to 2.5V) to instruct the FlashPro Express programmer(s) to supply Vpp, Vpn, Vdd(I) and Vddp.
- The Drive TRST option is unchecked to instruct the FlashPro Express programmer(s) NOT to drive the TRST pin.
- The Force TCK Frequency option is unchecked to instruct FlashPro Express to use the TCK frequency specified by the Frequency statement in the STAPL file(s).

## FlashPro Lite Programmer Settings

If you choose to add the Force TCK Frequency, select the appropriate MHz frequency. After you have made your selection(s), click **OK**.

### Default Settings

- The Vpp and Vpn options are checked to instruct the FlashPro Express Lite programmer(s) to supply Vpp and Vpn.
- The Drive TRST option is unchecked to instruct the FlashPro Express Lite programmer(s) NOT to drive the TRST pin.
- The Force TCK Frequency option is unchecked to instruct the FlashPro Express Lite to use the TCK frequency specified by the Frequency statement in the STAPL file(s).

## FlashPro5/4/3/3X Programmer Settings

For FlashPro5/4/3/3X, you have the option of choosing the Set Vpump setting or the Force TCK Frequency. If you choose the Force TCK Frequency, select the appropriate MHz frequency. For FlashPro4/3X settings, you have the option of switching the TCK mode between Free running clock and Discrete clocking. After you have made your selections(s), click **OK**.

### Default Settings

- The Vpump option is checked to instruct the FlashPro5/4/3/3X programmer(s) to supply Vpump to the device.  
NOTE: VPUMP voltage will not be checked for the SmartFusion2/IGLOO2 and newer families of devices. VPUMP does not need to be connected to the programmer for these devices.
- The Force TCK Frequency option is unchecked to instruct the FlashPro5/4/3/3X to use the TCK frequency specified by the Frequency statement in the PDB/STAPL file(s).
- FlashPro5/4/3/3X default TCK mode setting is Free running clock

## TCK Setting (ForceTCK Frequency)

If **Force TCK Frequency** is checked (in the **Programmer Setting**), the selected TCK value is set for the programmer and the Frequency statement in the PDB/STAPL file is ignored.

### Default TCK frequency

When the IPD/STAPL file or Chain does not exist, the default TCK frequency is set to 4MHz. When more than one Microsemi flash device is targeted in the chain, the FlashPro Express software passes through all of the files and searches for the "freq" keyword and the "MAX\_FREQ" **Note** field. The FlashPro Express software uses the lesser value of all the TCK frequency settings and the "MAX\_FREQ" **Note** field values.



## Ping Programmers

Right-click a programmer and choose Ping.

**Note:** You can click the Refresh/Rescan for Programmers button to quickly ping new programmers.

## Performing a Self-Test

Right-click the programmer you want to self-test and choose Self Test.

**Note:** You must connect the programmer to the self-test board that comes with your programmer before performing a self-test.

**Note:** Self-test is not supported with FlashPro5/4 or FlashPro Lite programmers. These programmers are rigorously tested at the factory during production.

## Scanning and Checking a Chain

The scan chain operation scans and analyzes the JTAG chain connected to programmer(s) you have selected and checks that chain scanned matches the chain configured in FlashPro Express.

**To scan a chain:**

Right-click the programmer you want to scan and choose **Scan and check chain**.

## Enabling and Disabling Programmers

After [loading a job project](#), you can enable/disable or remove a programmer and can also ping, self-test, run scan and check chain on any of the connected programmers. These actions are available in the shortcut menu (right-click) for each of the programmers listed in the programmer column.

**Click the checkbox** next to a programmer in the Programmer column to enable or disable it. The programmer is enabled when there is a tick mark in the checkbox and disabled when the checkbox is empty.

## Renaming a Programmer

Enter the new programmer name in the **Programmer** window to rename the programmer. By default, the programmer name is the same as the programmer ID.

## Removing a Programmer

Right-click the programmer and choose **Remove**.

## Selecting and Running an Action

FlashPro Express supports the following programming actions:

- DEVICE\_INFO
- ENC\_DATA\_AUTHENTICATION  
This action is only visible if every device in the chain contains encrypted bitstream files. Selecting this action causes each bitstream file to be checked for authentication.
- ERASE
- PROGRAM
- READ\_IDCODE
- VERIFY

**To select a programming action:**

Select an action from the Programming Action dropdown menu in FlashPro Express, as shown in the figure below.

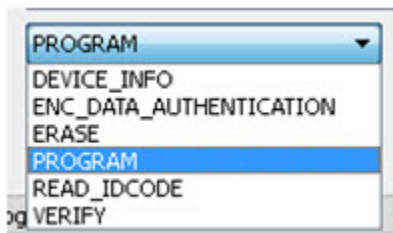


Figure 12 · FlashPro Express Programming Actions

***To run the selected programming action:***

Click on the **RUN** button below the Programming Action dropdown menu.

# Chain Programming

## Chain Order

Chain Programming enables you to program several devices at one time. The order of devices in the chain imported from Job Project must match the physical chain to be programmed.

The TDO for the first device connects to the programmer, and the last device's TDI connects to the programmer. The devices in the chain go in order from a device's TDI into the next device's TDO, as shown in the figure below.

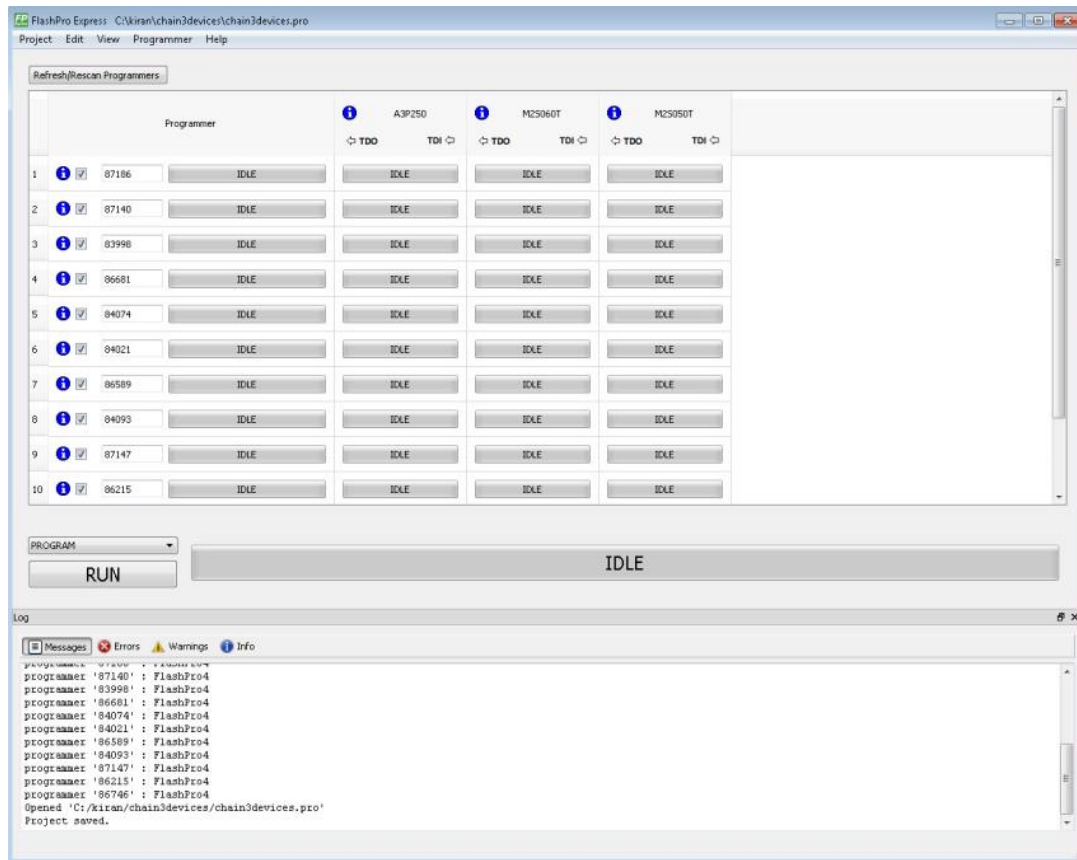


Figure 13 · Chain Order

## Multiple Device Chain Programming

The FlashPro Express software enables direct chain programming without generating a chain STAPL file. Each device will be programmed in sequential order starting from device 1 to device N. See example below. For more information about chain order, see the [Chain Order](#) help topic.

TDI > Device N > Device N-1 > ... > Device 2 > Device 1 > TDO

## Device Programming Compatibility

SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion families can be programmed in the same chain.

## Programmer Support

FlashPro5/4/3/3X supports the SmartFusion2, IGLOO2, RTG4, SmartFusion, IGLOO, ProASIC3, Fusion family devices. The Vpump on FlashPro5/4/3/3X is designed to support the programming of only one device. Please make sure that Vpump, Vcc and Vjtag are provided on board for chain programming. Connect the Vpump to the header as the Flashpro Express software will attempt to check for all external supplies, including Vpump, to ensure successful programming. There is no limitation to the chain length; however, ensure that the JTAG signal integrity and the timing are preserved.

# TCL Commands - FlashPro Express

## About TCL Commands - FlashPro Express Tcl Command Reference

A Tcl (Tool Command Language) file contains scripts for simple or complex tasks. You can run scripts from the Windows command line or store and run a series of Tcl commands in a \*.tcl batch file. The Tcl commands supported by FlashPro Express are listed in the table below.

**Note:** Tcl commands are case sensitive. However, their arguments are not.

Command	Action
<a href="#">close_project</a>	Closes the FlashPro project
<a href="#">configure_flashpro_prg</a>	Changes FlashPro programmer settings
<a href="#">configure_flashpro3_prg</a>	Changes FlashPro3 programmer settings
<a href="#">configure_flashpro4_prg</a>	Changes FlashPro 4 programmer settings
<a href="#">configure_flashpro5_prg</a>	Changes FlashPro 5 programmer settings
<a href="#">configure_flashproLite_prg</a>	Changes FlashPro Lite programmer settings
<a href="#">create_job</a>	Create a job in FlashPro Express
<a href="#">dump_tcl_support</a>	Unloads the list of supported FlashPro Tcl commands
<a href="#">enable_serialization</a>	Enables or disables serialization programming.
<a href="#">open_project</a>	Opens a FlashPro project
<a href="#">ping_prg</a>	Pings one or more programmers
<a href="#">refresh_prg_list</a>	Refreshes the programmer list
<a href="#">remove_prg</a>	Removes the programmer from the programmer list
<a href="#">run_selected_actions</a>	Runs the selected action on the specified programmer and returns the exit code from the action
<a href="#">save_log</a>	Saves the log file
<a href="#">save_project</a>	Saves the FlashPro project
<a href="#">scan_chain_prg</a>	Runs scan chain on a programmer
<a href="#">select_serial_range</a>	Selects the range of indexes to program.
<a href="#">self_test_prg</a>	Runs Self-Test on a programmer
<a href="#">set_prg_name</a>	Changes the user name of a programmer

Command	Action
<a href="#">set_programming_action</a>	Selects the action for a device
<a href="#">set_serialization_log_file</a>	Sets the path and name of the serialization log file.

## Running Tcl Scripts from within FlashPro Express

Instead of running scripts from the command line, you can use FlashPro Express's Execute Script dialog box to run a script.

**To execute a Tcl script file within FlashPro Express:**

1. From the **File** menu, choose **Execute Script** to display the **Run Script** dialog box.

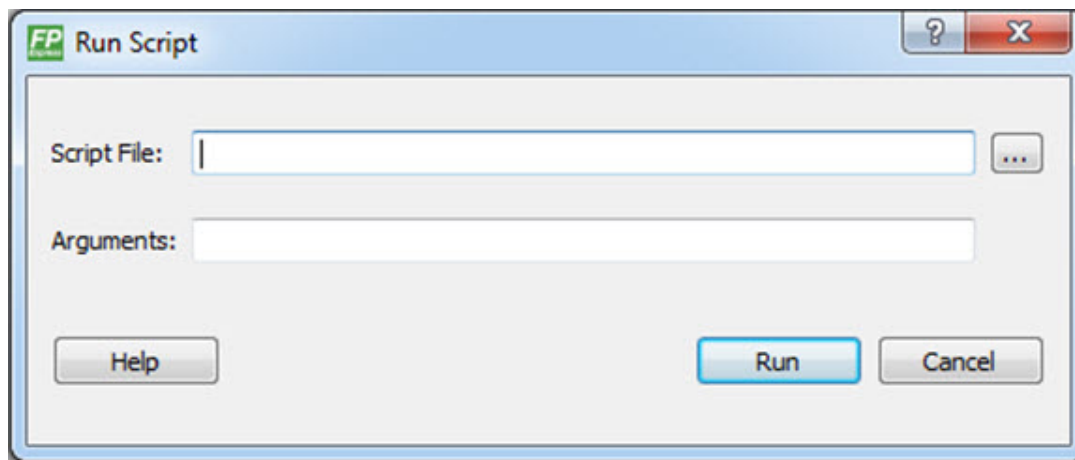


Figure 14 - Run Script Dialog Box

2. Click the **Browse** button to display the **Open** dialog box, in which you can navigate to the folder containing the script file to open. When you click **Open**, FlashPro Express enters the full path and script filename into the Run Script dialog box for you.
3. In the Arguments box, enter the arguments to pass to your Tcl script. Separate each argument by a space character. For information about accessing arguments passed to a Tcl script, see [Running Tcl Scripts From the Command Line](#).
4. Click **Run**.

## Running Tcl Scripts from the Command Line

You can run Tcl scripts from your Windows or Linux command line.

**To execute a Tcl script file in the FlashPro Express software from a shell command line:**

1. At the prompt, type the path to the Microsemi software followed by the word "SCRIPT" and a colon, and then the name of the script file as follows:

```
<location of Microsemi software>/bin/FPExpress.exe SCRIPT:<filename>
```

The example below executes in batch mode the script *foo.tcl*:

```
<location of Microsemi software>/bin/FPExpress.exe script:foo.tcl
```

The example below executes in batch mode the script *foo.tcl* and exports the log in the file *foo.txt*:

```
<location of Microsemi software>/bin/FPExpress.exe script:foo.tcl logfile:foo.txt
```

The example below executes in batch mode the script *foo.tcl*, creates a console where the log is displayed briefly, and exports the log in the file *foo.txt*:

```
<location of Microsemi software>/bin/FPEXpress.exe script:foo.tcl console_mode:brief
logfile:foo.txt
```

If you leave `console_mode` unspecified or set it to 'hide' FlashPro Express executes without a console window. If you want to leave the console window open you can run the script with the `console_mode` parameter set to 'show', as in the following example:

```
<location of Microsemi software>/bin/FPEXpress.exe script:foo.tcl console_mode:show
logfile:foo.txt
```

2. If you want to pass arguments to the Tcl script from the command line, then use the "SCRIPT\_ARGS" variable as follows:

```
<location of Microsemi software>/bin/FPEXpress.exe SCRIPT:<filename> SCRIPT_ARGS:"param1
param2 param3"
```

Arguments passed to a Tcl script can be accessed through the Tcl variables `argc` and `argv`. The example below demonstrates how a Tcl script accesses these arguments:

```
puts "Script name: $argv0"
puts "Number of arguments: $argc"
set i 0
foreach arg $argv {
    puts "Arg $i : $arg"
    incr i
}
```

**Note:** Script names can contain spaces if the script name is protected with double quotes:

```
FPEXpress script:"FPEXpress tcl/foo 1.tcl"
```

## Exporting Tcl Scripts from within FlashPro Express

*To export a set of Tcl commands from the FlashPro Express history:*

1. From the **File** menu, choose **Export Script File**.
2. Enter the filename and click **Save**. The Export Script Options dialog appears (as shown in the figure below).

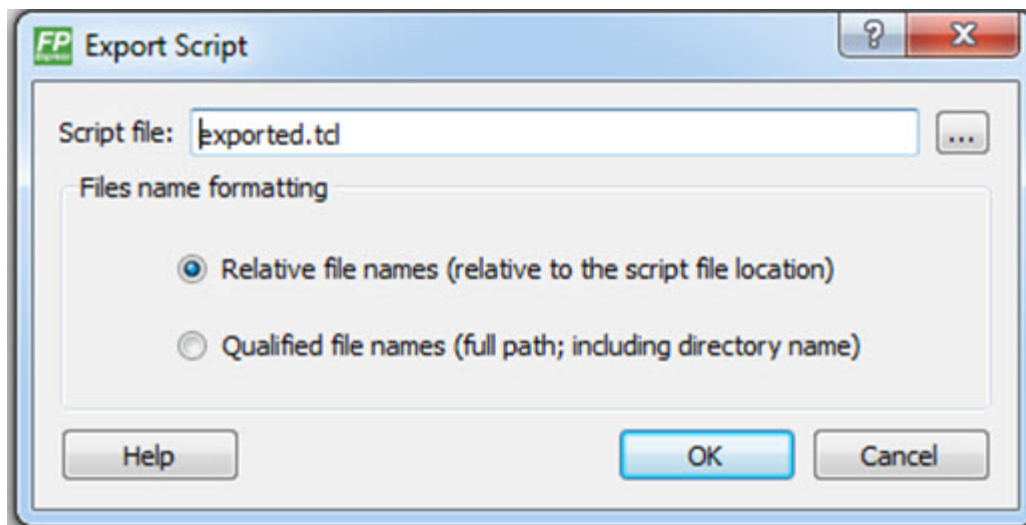


Figure 15 · Script Export Options Dialog Box

Check the **Include commands from current project only** to export commands of the current project only. You can specify the filename formatting by selecting **Relative filenames** (relative to the current directory) or **Qualified filenames** (absolute path, including the directory name).

4. Click **OK**.

## Tcl Commands Supported by FlashPro Express

### close\_project

Closes the FlashPro or FlashPro Express project.

```
close_project
```

#### Arguments

None

#### Supported Families

All

#### Exceptions

None

#### Example

```
close_project
```

### complete\_prog\_job

Tcl command; completes the current open job and generates a Job Status container including cryptographically signed Job Ticket end certifiers and Certificates of Conformance (if enabled) of the programmed devices. It archives ticket data from the HSM database. The resultant Job Status container can be imported into Job Manager and validated using U-HSM. If the job status file is not specified, the information is printed in the log window, and no Job Status container is created for subsequent verification.

The HSM Job can only be completed if the number of devices in each HSM ticket has been exhausted. If devices remain, the job can only be terminated by using the “-terminate” option.

```
complete_prog_job [-job_status_file path]\  
[-terminate]
```

**NOTE:** This command will fail if there are devices left in any HSM ticket and the terminate option is not used.

#### Arguments

`[-job_status_file path]`

Full path to the output Job Status container which contains End-Job Certifier and CofCs. If not specified, information will be printed in the log window.

`[-terminate]`

This option will terminate the HSM job even if there are devices left in any HSM ticket. This parameter is optional if the number of devices in all tickets have been exhausted.

#### Supported Families

SmartFusion2, IGLOO2

#### See Also:

SPPS User Guide



User HSM Installation Guide  
 Manufacturer HSM Installation Guide  
 Job Manager User Guide

## configure\_flashpro\_prg

Changes FlashPro programmer settings.

```
configure_flashpro_prg [-vpp {ON|OFF}] [-vpn {ON|OFF}] [-vddl {ON|OFF}] [-force_vddp {ON|OFF}] [-vddp {2.5|3.3}] [-drive_trst {ON|OFF}] [-force_freq {ON|OFF}] [-freq {freq}]
```

## Arguments

-vpp {ON|OFF}

Enables FlashPro programmer to drive VPP. Set to ON to drive VPP.

-vpn {ON|OFF}

Enables FlashPro programmer to drive VPN; set to ON to drive VPN.

-vddl {ON|OFF}

Enables FlashPro programmer to drive VDDL; set to ON to drive VDDL.

-force\_vddp {ON|OFF}

Enables FlashPro programmer to drive VDDP; set to ON to drive VDDP.

-vddp {2.5|3.3}

Sets VDDP to 2.5 or 3.3 volts.

-drive\_trst {ON|OFF}

Enables FlashPro programmer to drive TRST; set to ON to drive TRST.

-force\_freq {ON|OFF}

Forces the FlashPro software to use the TCK frequency specified by the software rather than the TCK frequency specified in the programmer file.

-freq {freq}

Specifies the TCK frequency in MHz.

## Supported Families

ProASIC<sup>PLUS</sup>, ProASIC

## Exceptions

None

## Example

The following example enables the FlashPro programmer to drive the VPP, VPN, VDDL, VDDP, sets the drive voltage to 3.3v, disables the driver for TRST, and does not force the programmer to use the TCK frequency specified in the software.

```
configure_flashpro_prg -vpp {ON} -vpn {ON} -vddl {ON} -force_vddp {ON} -vddp {3.3} -drive_trst {OFF} -force_freq {OFF}
```

## configure\_flashpro3\_prg

Changes FlashPro3 programmer settings.

```
configure_flashpro3_prg [-vpump {ON|OFF}] [-clk_mode {discrete_clk|free_running_clk}] [-force_freq {ON|OFF}] [-freq {freq}]
```

## Arguments

**-vpump** {ON|OFF}

Enables FlashPro programmer to drive VPUMP. Set to ON to drive VPUMP.

**-clk\_mode** {discrete\_clk|free\_running\_clk}

Specifies free running or discrete TCK.

**-force\_freq** {ON|OFF}

Forces the FlashPro software to use the TCK frequency specified by the software rather than the TCK frequency specified in the programmer file.

**-freq** {freq}

Specifies the TCK frequency in MHz.

## Supported Families

SmartFusion, IGLOO, ProASIC3 and Fusion

## Exceptions

None

## Example

The following example sets the VPUMP option to ON, TCK to free running, and uses the TCK frequency specified in the programmer file (force\_freq is set to OFF):

```
configure_flashpro3_prg -vpump {ON} -clk_mode {free_running_clk} -force_freq {OFF} -freq {4}
```

The following example sets VPUMP to ON, TCK to discrete, forces the FlashPro software to use the TCK frequency specified in the software (-force\_freq is set to ON) at a frequency of 2 MHz.

```
configure_flashpro3_prg -vpump {ON} -clk_mode {discrete_clk} -force_freq {ON} -freq {2}
```

## configure\_flashpro4\_prg

Changes FlashPro4 programmer settings.

```
configure_flashpro4_prg [-vpump {ON|OFF}] [-clk_mode {discrete_clk|free_running_clk}] [-force_freq {ON|OFF}] [-freq {freq}]
```

## Arguments

**-vpump** {ON|OFF}

Enables FlashPro4 programmer to drive VPUMP. Set to ON to drive VPUMP.

**-clk\_mode** {discrete\_clk|free\_running\_clk}

Specifies free running or discrete TCK.

**-force\_freq** {ON|OFF}

Forces the FlashPro software to use the TCK frequency specified by the software rather than the TCK frequency specified in the programmer file.

**-freq** {freq}

Specifies the TCK frequency in MHz.

## Supported Families

SmartFusion, IGLOO, ProASIC3 and Fusion

## Exceptions

None

## Example

The following example sets the VPUMP option to ON and uses a free running TCK at a frequency of 4 MHz (force\_freq is set to OFF).

```
configure_flashpro4_prg -vpump {ON} -clk_mode {free_running_clk} -force_freq {OFF} -freq {4}
```

The following example sets the VPUMP option to ON, uses a discrete TCK and sets force\_freq to ON at 2 MHz.

```
configure_flashpro4_prg -vpump {ON} -clk_mode {discrete_clk} -force_freq {ON} -freq {2}
```

## configure\_flashpro5\_prg

Tcl command; changes FlashPro5 programmer settings.

```
configure_flashpro5_prg [-vpump {ON/OFF}] [-clk_mode {free_running_clk}]  
[-force_freq {ON/OFF}] [-freq {freq}]
```

## Arguments

-vpump {ON|OFF}

Enables FlashPro5 programmer to drive VPUMP. Set to ON to drive VPUMP. Default is ON.

-clk\_mode {free\_running\_clk}

Specifies free running TCK. Default is free\_running\_clk.

-force\_freq {ON|OFF}

Forces the FlashPro software to use the TCK frequency specified by the software rather than the TCK frequency specified in the programmer file. Default is OFF.

-freq {freq}

Specifies the TCK frequency in MHz. Default is 4.

## Supported Families

SmartFusion2, IGLOO2

## Exceptions

None

## Example

The following example sets the VPUMP option to ON and uses a free running TCK at a frequency of 4 MHz (force\_freq is set to OFF).

```
configure_flashpro5_prg -vpump {ON} -clk_mode {free_running_clk} -force_freq {OFF} -freq {4}
```

The following example sets the VPUMP option to ON, uses a free running TCK and sets force\_freq to ON at 2 MHz.

```
configure_flashpro5_prg -vpump {ON} -clk_mode {free_running_clk} -force_freq {ON} -freq {2}
```

## configure\_flashproLite\_prg

Changes FlashPro Lite programmer settings.

```
configure_flashproLite_prg [-vpp {ON|OFF}] [-vpn {ON|OFF}] [-drive_trst {ON|OFF}] [-force_freq {ON|OFF}] [-freq {freq}]
```

### Arguments

-vpp {ON|OFF}

Enables FlashPro programmer to drive VPP. Set to ON to drive VPP.

-vpn {ON|OFF}

Enables FlashPro programmer to drive VPN; set to ON to drive VPN.

-drive\_trst {ON|OFF}

Enables FlashPro programmer to drive TRST; set to ON to drive TRST.

-force\_freq {ON|OFF}

Forces the FlashPro software to use the TCK frequency specified by the software rather than the TCK frequency specified in the programmer file.

-freq {freq}

Specifies the TCK frequency in MHz.

### Supported Families

ProASIC<sup>PLUS</sup>

### Exceptions

None

### Example

The following example sets the programmer to drive the VPP, drive VPN, drive the TRST and uses the frequency set by the programmer file (sets force\_freq to OFF):

```
configure_flashprolite_prg -vpp {ON} -vpn {ON} -drive_trst {ON} -force_freq {OFF}
```

## create\_job\_project

Tcl command; creates a Flashpro Express job using the programming job exported from Libero.

```
create_job_project -job_project_location location -job_file path -overwrite 0/1
```

### Arguments

-job\_project\_location location

Specifies the location for your FlashPro Express job project.

-job\_file path

Path to the Libero job file that is used as input to create the Flashpro Express job project.

-overwrite 0/1

Set value to 1 to overwrite your existing job project. .

## Supported Families

SmartFusion2, IGLOO2, RTG4

## Exceptions

None

## Example

The following example creates a job project named test.job in the \fpexpress directory. It does not overwrite the existing job project.

```
create_job_project \
-job_project_location {D:\fpexpress} \
-job_file {D:\test\designer\test\export\test.job} -overwrite 0\
```

## dump\_tcl\_support

Unloads the list of supported FlashPro or FlashPro Express Tcl commands.

```
dump_tcl_support -file {file}
```

## Arguments

-file {file}

## Supported Families

All

## Exceptions

None

## Example

The following example dumps your Tcl commands into the file 'tcldump.tcl'

```
dump_tcl_support -file {tcldump.tcl}
```

## enable\_serialization

This Tcl command enables or disables serialization programming.

```
enable_serialization -name {device_name} -enable {true/false}
```

### Arguments

-name

Specifies the device name.

-enable

Enables (true) or disables (false) serialization programming.

## Exceptions

Must be a Microsemi Device

## Supported Families

Fusion, ProASIC3, IGLOO, SmartFusion, SmartFusion2, IGLOO2

## Example

```
enable_serialization -name M2S/M2GL050{T|S|TS} -enable true
```

## get\_job\_status

Tcl command; exports status of current open job. The job status contains number of devices left for each HSM ticket. If job status file is not specified, the information is printed in the log window.

```
get_job_status [-job_status_file path] \  
               [-archive]
```

## Arguments

[-job\_status\_file *path*]

Path to the output FlashPro Express job status container. The job status file can be sent to the Job Manager application and Certificates of Conformance (if available) validated using the U-HSM.

[-archive]

Moves the HSM ticket log files from the HSM ticket database to the HSM ticket archive. The archive folder was specified during HSM installation and setup.

**NOTE:** If no job\_status\_file is specified, the archive option prints the Certificates of Conformance in the log window without exporting them.

## Supported Families

SmartFusion2, IGLOO2

### See Also

SPPS User Guide

Manufacturer HSM Installation Guide

User HSM Installation Guide

Job Manager User Guide

## open\_project

Opens a FlashPro or FlashPro Express project.

```
open_project -project {project}
```

## Arguments

-project {*project*}

Specifies the location and name of the project you wish to open.

## Supported Families

All

## Exceptions

None

## Example

Opens the 'FPPrj1.pro' project from the FPPProject1 directory  
`open_project -project {./FPPProject1/FPPrj1.pro}`

## ping\_prg

Pings one or more programmers.

```
ping_prg (-name {name})*
```

## Arguments

-name {*name*}

Specifies the programmer to be pinged. Repeat this argument for multiple programmers.

## Supported Families

All

## Exceptions

None

## Example

The following example pings the programmers 'FP300085' and 'FP300086'.

```
ping_prg -name {FP300085} -name {FP300086}
```

## process\_job\_request

Tcl command; processes a job request received from Job Manager. It is part of the Job Ticket generation process.

**NOTE1:** This command does not require a FlashPro Express project to be created or opened.

**NOTE2:** HSM parameters must be configured using set\_hsm\_params before processing job request.

```
process_job_request -request_file path \  
  -reply_file path \  
  [-overwrite_reply {TRUE | FALSE}]
```

## Arguments

-request\_file *path*

Full file name of job request file.

```
-reply_file path
Full file name of job reply file.
-overwrite_reply {TRUE | FALSE}
TRUE allows overwriting of any pre-existing reply_file.
```

## Supported Families

SmartFusion2 and IGLOO2

## Example

```
process_job_request \
  -request_file {D:/flashpro_files/jobmgr_project12/cm_request.req} \
  -reply_file {D:/flashpro_files/jobmgr_project12/cm_reply.rep} \
  -overwrite_reply {TRUE}
```

### See Also

SPPS User Guide

Job Manager User Guide

[set\\_hsm\\_params](#)

## refresh\_prg\_list

Refreshes the programmer list. This is most often used to have FlashPro or FlashPro Express detect a programmer that you have just connected.

```
refresh_prg_list
```

## Arguments

None

## Supported Families

All

## Exceptions

None

## Example

```
refresh_prg_list
```

## remove\_hsm\_tickets

Tcl command; removes HSM tickets from the HSM using one of the following methods:

- By specifying the job reply file in which case all tickets that are in the reply file will be deleted.
- By specifying each of the ticket IDs value in hexadecimal string.

```
remove_hsm_tickets [-reply_file path] \
  [-ticket_ids ticketID+] \
  [-all]
```



## Arguments

`-reply_file` *path*

Full file name of job request file.

`-ticket_ids` *ticketID+*

Hex value of each ticket ID to be removed.

## Supported Families

SmartFusion2, IGLOO2

## Example

```
remove_hsm_tickets \
  -reply_file {D:\flashpro_files\jobmgr_project12cm_reply.rep}
remove_hsm_tickets \
  -ticket_ids {000000000000000000000000899f252d9fb55442aa7e
0000000000000000000000b6f385c6a9eeca69705c
0000000000000000000000ed5702d0b767ba686b82}
```

## NOTES

- This command should be used very carefully since it removes HSM tickets, rendering any FlashPro Express jobs based on those tickets to be unusable.
- This command does not require a FlashPro Express project to be created or opened.

### See Also

[SPPS User Guide](#)

[Job Manager User Guide](#)

## remove\_prg

Removes the programmer from the programmer list.

```
remove_prg (-name {name})*
```

## Arguments

`-name` {*name*}\*

Specifies the programmer to be removed. You can repeat this argument for multiple programmers.

## Supported Families

All

## Exceptions

None

## Example

The following example removes the programmer '03178' from the programmer list:

```
remove_prg (name {03178})*
```

## run\_selected\_actions

Runs the selected action on the specified programmer and returns the exit code from the action. If no programmer name is specified, the action is run on all connected programmers. Only one exit code is returned, so return code cannot be used when action is run on more than one programmer. A programming file must be loaded.

```
run_selected_actions [(-name {name})*]
```

### Arguments

-name {*name*}

Optional argument that specifies the programmer name. You can repeat this argument for multiple programmers.

### Supported Families

All

### Exceptions

None

### Example

The following example runs the selected actionS on the programmers 'FP30085' and 'FP30086'.

```
run_selected_actions -name {FP300085} -name {FP300086}
```

Example using return code:

```
if {[catch {run_selected_actions} return_val]} {puts "Error running Action"} else {puts "exit code $return_val"}
```

Example returning exit code to the command line (returns exit 99 on script failure, otherwise returns exit code from selected action):

```
if {[catch {run_selected_actions} return_val]}{exit 99} else {exit $return_val}
```

## save\_log

Saves the log file.

```
save_log -file {file}
```

### Arguments

-file {*file*}

Specifies the log filename.

### Supported Families

All

### Exceptions

None

## Example

The following example saves the log file with the name 'my\_logfile1.log':

```
save_log -file {my_logfile1.log}
```

## save\_project

Saves the FlashPro or FlashPro Express project.

```
save_project
```

## Arguments

None

## Supported Families

All

## Exceptions

None

## Example

```
save_project
```

## scan\_chain\_prg

In single mode, this command runs scan chain on a programmer.

In chain mode, this command runs scan and check chain on a programmer if devices have been added in the grid.

```
scan_chain_prg [(-name {name})+]
```

## Arguments

-name {*name*}

Specifies the programmer name.

## Supported Families

All

## Exceptions

None

## Example

The following example runs scan chain on a single programmer (single mode) named '21428':

```
scan_chain_prg -name {21428}
```

## select\_serial\_range

This Tcl command selects the range of indexes to program.

```
select_serial_range -name device_name -from_data start_index_to_program -to_data  
end_index_to_program
```

### Arguments

-name  
Specifies the device name.

-from\_data  
Specifies the start index to program.

-to\_data  
Specifies the end index.

### Supported Families

Fusion, ProASIC3, IGLOO, SmartFusion, SmartFusion2, IGLOO2

### Exceptions

Must be a Microsemi Device

### Example

```
select_serial_range -name M2S/M2GL050{T|S|TS} -from_data 3 -to_data 5
```

## self\_test\_prg

Runs Self-Test on a programmer.

```
self_test_prg (-name {name}) *
```

### Arguments

-name {*name*}  
Specifies the programmer name. You can repeat this argument for multiple programmers.

### Supported Families

All

### Exceptions

None

### Example

The following examples runs the self test on the programmer '30175':

```
self_test_prg (-name {30175}) *
```

## set\_hsm\_params

Tcl command; saves the HSM parameters for the FlashPro Express application. These parameters remain in effect until overridden by another invocation of this command.

```
set_hsm_params -hsm_server_name hsm_server \
               -hsm_type_u {TRUE|FALSE} \
               -m_hsm_uuid m_uuid \
               -ftp_username ftp_username \
               -ftp_password ftp_password
```

**NOTE1:** The HSM parameters are persistent between multiple FlashPro Express sessions on the same computer.

**NOTE2:** HSM parameters only need to be set for HSM flow jobs.

## Arguments

-hsm\_server *hsm\_server*  
 Name or IP address of HSM server computer

-hsm\_type\_u {TRUE|FALSE}  
 TRUE FlashPro Express will use the Manufacturer features of the User HSM.  
 FALSE FlashPro Express will use a Manufacturer HSM.

-m\_hsm\_uuid *m\_uuid*  
 UUID of HSM to be used for FlashPro Express tasks.

-ftp\_username *ftp\_username*  
 User name to access the HSM files via FTP server.

-ftp\_password *ftp\_password*  
 Password to access the HSM files via FTP server.

## Supported Families

SmartFusion2 and IGLOO2

## Example

```
set_hsm_params -hsm_server_name {10.241.140.224} \
               -hsm_type_u {0} \
               -m_hsm_uuid {00000000000000000000000000000000000000000002} \
               -ftp_username {hsm} \
               -ftp_password {hsm}
```

## set\_prg\_name

Changes the user name of a programmer.

```
set_prg_name -name {name} -new_name {new_name}
```

## Arguments

-name {*name*}  
 Identifies the old programmer name.

-new\_name {*new\_name*}  
 Specifies the new programmer name.

## Supported Families

All devices supported by FlashPro.

## Exceptions

None

## Example

The following example changes the name of the programmer 'FP300086' to 'FP3Prg2':

```
set_prgr_name -name {FP300086} -new_name {FP3Prg2}
```

## set\_programming\_action

Selects the action for a device. The device name parameter must be specified only in chain programming mode. A programming file must be loaded. The device must be a Microsemi device.

```
set_programming_action [-name {name}] -action {action}
```

## Arguments

-name {*name*}  
 Specifies the device name.  
 -action {*action*}  
 Specifies the action.

## Supported Families

SmartFusion, IGLOO, ProASIC3, Fusion, SmartFusion2, IGLOO2, RTG4

## Exceptions

Must be a Microsemi device

## Example

The following example sets the programming action in single programming mode:

```
set_programming_action -action {PROGRAM}
```

And in chain programming mode:

```
set_programming_action -name {MyDevice1} -action {ERASE}
```

## set\_programming\_file

Sets the programming file for a device. Either the *file* or the *no\_file* flag must be specified. A programming file must be loaded. The device must be a Microsemi device .

```
set_programming_file [-name {name}] [-file {file}] [-no_file { }]
```

## Arguments

-name {*name*}

Specifies the device name. This argument must be specified only in chain programming mode.

-file {*file*}

Specifies the programming file.

-no\_file

Specifies to unload the current programming file.

## Supported Families

SmartFusion, IGLOO, ProASIC3, Fusion, SmartFusion2, IGLOO2, RTG4

## Exceptions

Must be a Microsemi device.

## Examples

In single programming mode:

```
set_programming_file -file {e:/design/pdb/TopA3P250.pdb}
```

In chain programming mode:

```
set_programming_file -name {MyDevice2} -file {e:/design/pdb/TopA3P250.pdb}
```

```
set_programming_file -name {MyDevice1} -no_file
```

## set\_serialization\_log\_file

This Tcl command sets the path and name of the serialization log file.

```
set_serialization_log_file -file {log_file_path}
```

## Arguments

-file

Specifies the serialization log file path and name

## Supported Families

Fusion, ProASIC3, IGLOO, SmartFusion, SmartFusion2, IGLOO2

## Exceptions

Must be a Microsemi Device

## Example

```
set_serialization_log_file -file {C:/local_z_folder/work/my_serial_log}
```

# Troubleshooting

## Exit Codes (SmartFusion2 and IGLOO2)

Error Code	Exit Code	Exit Message	Possible Cause	Possible Solution
	0	Passed (no error)	-	-
0x8002	5	Failure to configure device programming at 1.2/1.0 VCC voltage	Unstable voltage level  Signal integrity issues on JTAG pins	Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your device datasheet for more information on transient specifications.  Monitor JTAG supply pins during programming; measure JTAG signals for noise or reflection.
0x8032	5	Device is busy	Unstable VDDIx voltage level	Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your device datasheet for more information on transient specifications.
0x8003	5	Failed to enter programming mode	Unstable voltage level  Signal integrity issues on JTAG pins  DEVRST_N is tied to LOW	Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your device datasheet for more information on transient specifications.  Monitor JTAG supply pins during programming; measure JTAG signals for noise or reflection.  Tie DEVRST_N to HIGH prior to programming the device.
0x8004	6	Failed to verify IDCODE	Incorrect programming file  Incorrect device in chain  Signal integrity issues on JTAG pins	Choose the correct programming file and select the correct device in the chain.  Measure JTAG pins and noise for reflection. If TRST is left floating then add pull-up to pin.  Reduce the length of Ground connection.
0x8005 0x8006 8x804A	10	Failed to program eNVM	Unstable voltage level.	Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi



Error Code	Exit Code	Exit Message	Possible Cause	Possible Solution
			Signal integrity issues on JTAG pins.	<p>specifications. See your device datasheet for more information on transient specifications.</p> <p>Monitor JTAG supply pins during programming; measure JTAG signals for noise or reflection.</p>
0x8027 0x8028	10	Authentication Error  Bitstream and device mismatch	Libero device selection does not match the target device.	Generate a programming file with the correct device selection for the target device.
0x8007 0x804C	11	Failed to verify FPGA Array  Failed to verify Fabric Configuration  Failed to verify Security	<p>Device is programmed with a different design or the component is blank.</p> <p>Unstable voltage level.</p> <p>Signal integrity issues on JTAG pins.</p>	<p>Verify the device is programmed with the correct data/design.</p> <p>Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your device datasheet for more information on transient specifications.</p> <p>Monitor JTAG supply pins during programming; measure JTAG signals for noise or reflection.</p>
0x8008 0x8009 0x8049	11	Failed to verify eNVM	<p>Device is programmed with a different design.</p> <p>Unstable voltage level.</p> <p>Signal integrity issues on JTAG pins.</p>	<p>Verify the device is programmed with the correct data/design.</p> <p>Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your device datasheet for more information on transient specifications.</p> <p>Monitor JTAG supply pins during programming; measure JTAG signals for noise or reflection.</p>
0x8013	-18	Digest request from SPI/JTAG is protected by User Pass Key 1	Digest request from SPI/JTAG is protected by user pass key 1. Lock bit has been configured in the Debug Policy within SPM (Security Policy Manager)	Provide a programming file with a pass key that matches pass key programmed into the device.
0x8014	-19	Failed to verify digest	<p>&gt;Unstable voltage level</p> <p>Signal integrity issues on JTAG pins</p>	Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your device datasheet for more information on transient specifications.

Error Code	Exit Code	Exit Message	Possible Cause	Possible Solution
				Monitor JTAG supply pins during programming; measure JTAG signals for noise or reflection.
0x8015	-20	FPGA Fabric digest verification: FAIL	Programming bitstream components do not match components programmed  FPGA Fabric is either erased or the data has been corrupted or tampered with	Use the same programming file that was used to program the device.
0x8016	-20	eNVM_0 digest verification: FAIL	Programming bitstream components do not match components programmed  eNVM_0 data has been corrupted or tampered with	Use the same programming file that was used to program the device.
0x8017	-20	eNVM_1 digest verification: FAIL	Programming bitstream components do not match components programmed  eNVM_1 data has been corrupted or tampered with	Use the same programming file that was used to program the device.
0x8018	-20	User security policies segment digest verification: FAIL	Programming bitstream components do not match components programmed  User security policy segment data has been corrupted or tampered with	Use the same programming file that was used to program the device.
0x8019	-20	User key set 1 segment digest verification: FAIL	Programming bitstream components do not match components programmed  User key set 1 segment data has been corrupted or tampered with	Use the same programming file that was used to program the device.
0x801A	-20	User key set 2 segment digest verification: FAIL	Programming bitstream components do not match components programmed  User key set 2 segment data has been corrupted or tampered with	Use the same programming file that was used to program the device.
0x801B	-20	Factory row and factory key segment digest verification: FAIL	Programming bitstream components do not match components programmed  Factory row and factory key	Use the same programming file that was used to program the device.

Error Code	Exit Code	Exit Message	Possible Cause	Possible Solution
			segment data has been corrupted or tampered with	
0x801C	-20	Fabric configuration segment digest verification: FAIL	<p>Programming bitstream components do not match components programmed.</p> <p>Fabric configuration segment data has been corrupted or tampered with</p>	Use the same programming file that was used to program the device.
0x801D 0x801E 0x804B	-21	Device security prevented operation	<p>The device is protected with user pass key 1 and the bitstream file does not contain user pass key 1.</p> <p>User pass key 1 in the bitstream file does not match the device.</p>	<p>Run DEVICE_INFO to view security features that are protected.</p> <p>Provide a bitstream file with a user pass key 1 that matches the user pass key 1 programmed into the device.</p>
0x801F 0x8020 0x8040	-22	<p>Authentication Error</p> <p>Bitstream or data is corrupted or noisy</p>	<p>eNVM has been locked by a master in your design</p> <p>Running VERIFY action on a blank device.</p> <p>Bitstream file has been corrupted</p> <p>Bitstream was incorrectly generated</p>	<p>Release the lock on the eNVM after your master has completed its access operations. Write 0x00 to "REQACCESS" register in eNVM Control Registers (address 0x600801FC) to release the access.</p> <p>Program the device prior to running VERIFY action</p> <p>Regenerate bitstream file.</p>
0x8021 0x8022	-23	<p>Authentication Error</p> <p>Invalid/Corrupted encryption key</p>	<p>File contains an encrypted key that does not match the device</p> <p>Attempting to erase a device with no security using master security file</p> <p>File contains user encryption key, but device has not been programmed with the user encryption key</p> <p>Device has user encryption key 1/2 enforced and you are attempting to reprogram security settings</p>	<p>Provide a programming file with an encryption key that matches that on the device.</p> <p>Run DEVICE_INFO action to verify that the device has no security. If the device does not have security, you cannot erase it.</p> <p>First program security with master programming file, then program with user encryption 1/2 field update programming files.</p> <p>You must first ERASE security with the master security file, then you can reprogram new security settings.</p>
0x8041	-23	<p>Authentication Error</p> <p>Invalid/Corrupted encryption key</p>	<p>File contains an encrypted key that does not match the device</p> <p>File contains user encryption</p>	<p>Provide a programming file with an encryption key that matches that on the device.</p> <p>Run DEVICE_INFO action to verify that</p>

Error Code	Exit Code	Exit Message	Possible Cause	Possible Solution
			<p>key, but device has not been programmed with the user encryption key</p> <p>Attempting to erase a device with no security using master security file</p> <p>Device has user encryption key 1/2 enforced and you are attempting to reprogram security settings</p>	<p>the device has no security. If the device does not have security, you cannot erase it.</p> <p>First program security with master programming file, then program with user encryption 1/2 field update programming files.</p> <p>You must first ERASE security with the master security file, then you can reprogram new security settings.</p>
0x8023 0x8024 0x8042	-24	Authentication Error  Back level not satisfied	Design version is not higher than the back-level programmed device	Generate a programming file with a design version higher than the back level version.
0x8001	-24	Failure to read DSN	<p>Device is in System Controller Suspend Mode</p> <p>Check board connections</p>	TRSTB should be driven High or disable "System Controller Suspend Mode".
0x8025 0x8026 0x8043	-25	Authentication Error  DSN binding mismatch	DSN specified in programming file does not match the device being programmed	Use the correct programming file with a DSN that matches the DSN of the target device being programmed.
0x8044	-26	Authentication Error  Insufficient device capabilities	Device does not support the capabilities specified in programming file	Generate a programming file with the correct capabilities for the target device.
0x8027 0x8028	-26	Authentication Error  Bitstream and device mismatch	Libero device selection does not match the target device	Generate a programming file with the correct device selection for the target device.
0x8029 0x802A 0x8045	-27	Authentication Error  Incorrect DEVICEID	<p>Incorrect programming file</p> <p>Incorrect device in chain</p> <p>Signal integrity issues on JTAG pins</p>	<p>Choose the correct programming file and select the correct device in chain.</p> <p>Measure JTAG pins and noise or reflection. If TRST is left floating, then add pull-up to pin.</p> <p>Reduce the length of ground connection.</p>
0x802B 0x802C	-28	Authentication Error  Programming file is out of date, please regenerate	Programming file version is out of date	Generate programming file with latest version of Libero SoC.
0x8046	-28	>Authentication Error  Unsupported bitstream protocol version	Old programming file	Generate programming file with latest version of Libero SoC.

Error Code	Exit Code	Exit Message	Possible Cause	Possible Solution
0x802F	-30	JTAG interface is protected by UPK1	Invalid or no UPK1 is provided	User needs to provide correct UPK1 to unlock device.
0x8030 0x8031 0x8048	-31	Authentication Error  Invalid or inaccessible Device Certificate	M2S090 Rev. A or M2S150 Rev. A:  Either certificate is corrupted or the user hasn't provided the application code in the eNVM or provided invalid application code  FAB_RESET_N is tied to ground	User can program a valid application code. This can be done with SoftConsole.  FAB_RESET_N should be tied to HIGH.
0x8032 0x8033 0x8034 0x8035 0x8036 0x8037 0x8038 0x8039	-32	Instruction timed out	Unstable voltage level  Signal integrity issues on JTAG pins	Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your device datasheet for more information on transient specifications.  Monitor JTAG supply pins during programming; measure JTAG signals for noise or reflection.
0x8010	-35	Failed to unlock User Pass Key 1	Pass key in file does not match device.  Plaintext pass key match is disabled. This occurs if HSM was used to program the device.	Provide a programming file with a pass key that matches pass key programmed into the device.  Match pass key using HSM.
0x8011	-35	Failed to unlock User Pass Key 2	Pass key in file does not match device.  Plaintext pass key match is disabled. This occurs if HSM was used to program the device.	Provide a programming file with a pass key that matches pass key programmed into the device.  Match pass key using HSM.
0x8012	-35	Failed to unlock debug pass key	Pass key in file does not match device.  Plaintext pass key match is disabled. This occurs if HSM was used to program the device.	Provide a programming file with a pass key that matches pass key programmed into the device.  Match pass key using HSM.
0x804D	-36	<HSM related error message based on scenario>	HSM communication error. HSM call returns error.	Check if HSM the communication path to HSM is up. Make sure project is loaded properly and that HSM tickets have not been cleaned.

Error Code	Exit Code	Exit Message	Possible Cause	Possible Solution
0x804E	-37	Device already has Security programmed. Please erase the device using master file before reprogramming Security Settings.	HSM flow does not support reprogramming device directly if Security has already been programmed.	Erase security and try programming the device.

## Exit Codes for Software v8.6 and Above (SmartFusion, IGLOO, ProASIC3 and Fusion)

The table below lists exit codes for SmartFusion, IGLOO, ProASIC3 and Fusion devices in software v8.6 and ABOVE only. See the [Device Exit Codes for pre-v8.6 Software](#) help topic for exit codes for older versions.

**Note:** Exit codes with positive integers are reserved for current and future standard EXIT codes of the STAPL standard. Exit codes with negative integers are reserved for vendor-specific EXIT codes.

Table 3 · Exit Codes for SmartFusion, IGLOO, ProASIC3 and Fusion Family Devices in Software v8.6 and Above

Error Code	Exit Code	Exit Message	Possible Cause	Possible Solution
	0	Passed (no error)		
	1	A physical chain does not match the expected set up from the STAPL file. Also known as Checking Chain Error.	Physical chain configuration has been altered. Something has become disconnected in the chain. The specific IR length of non-Microsemi devices may be incorrect. The order of the specified chain may be incorrect.	
0x8052	5	Failed to enter programming mode.	Unstable VPUMP voltage level.  Unstable VCC  Signal integrity issues on JTAG pins.  Device is in FlashFreeze mode (ProASICL or IGLOO devices)  Older software or	Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your <a href="#">device datasheet</a> for more information on transient specifications.  Monitor VJTAG during programming; measure JTAG signals for noise or reflection.  Disable the FlashFreeze pin (ProASICL or IGLOO devices)  Generate STAPL file with the latest version of Designer/FlashPro. Use latest version of FlashPro software.

Error Code	Exit Code	Exit Message	Possible Cause	Possible Solution
			programming file used.	
0x801D 0x8053	6	Failed to verify IDCODE	Incorrect programming file  Incorrect device in chain  Signal integrity issues on JTAG pins	Choose the correct programming file and select the correct device in chain.  Measure JTAG pins and noise or reflection. If TRST is left floating then add pull-up to pin.  Reduce the length of ground connection.
0x8005 0x8009 0x800B	6	Failed to verify AES Sec.	Programming file generated with an older version of software	Generate STAPL file with the latest version of Designer/FlashPro. Use latest version of FlashPro software.  Try again at a slower TCK.  Contact Microsemi Technical Support.
0x8008	6	Failed to verify IDCODE.  Target is an M7 device	File is not for M7, but target device is M7  Signal integrity issues on JTAG pins.	Check that the target device is M7 enabled.  Make sure that the programming file you generated is for an M7-enabled device.  Measure JTAG pins, noise and reflection.
0x800A	6	Failed to verify IDCODE  Target is an M1 device	Files not for M1, but target device is M1.  Signal integrity issues on JTAG pins	Check that the target device is M1 enabled.  Make sure the programming file generated is for an M1-enabled device.  Monitor VJTAG during programming; measure JTAG signals for noise or reflection.
0x800C	6	Failed to verify IDCODE.  Core enabled device detected	File is not for target device.  Signal integrity issues on JTAG pins	Check the target device; make sure the programming file generated is matches the target device.  Monitor VJTAG during programming; measure JTAG signals for noise or reflection.
0x800D	6	Failed to verify IDCODE.  The target is not M7 device	File is for M7 but target device is not M7.  Signal integrity issues on JTAG pins.	Check that the target device is not M7 enabled.  Make sure that the programming file generated is for non-M7 enabled device.  Monitor VJTAG during programming; measure JTAG signals for noise or reflection.
0x800E	6	Failed to verify IDCODE.  Target is not an M1 device	File is for M1, but target device is not M1.	Check that the target device is not M1 enabled.  Make sure that the generated programming file is for non-M1 enabled device.

Error Code	Exit Code	Exit Message	Possible Cause	Possible Solution
			Signal integrity issues on JTAG pins	Monitor VJTAG during programming; measure JTAG signals for noise or reflection.
0x8006	6	Failed to verify IDCODE.  Target is not a P1 device	File is not for P1, but target device is a P1 device.  Signal integrity issues on JTAG pins	Check that the target device is P1 enabled.  Make sure programming file generated is for M1 enabled device.  Monitor VJTAG during programming; measure JTAG signals for noise or reflection.
0x801E	6	A3PE600 Engineering Sample Device Detected. This device is supported with pre-v8.3 SP1 STAPL files only		Contact Microsemi Technical Support
0x8057	8	Failed Erase Operation.	Unstable VPUMP voltage level.  Unstable VCC  Unstable VCC_OSC (Fusion only)  Unstable VCC_ROSC voltage level (SmartFusion only)  Signal integrity issues on JTAG pins.	Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your <a href="#">device datasheet</a> for more information on transient specifications.  Monitor VJTAG during programming; measure JTAG signals for noise or reflection.
0x8058	10	Failed to program FPGA array at row <row number>.	Unstable VPUMP voltage level.  Unstable VCC  Unstable VCC_OSC (Fusion only)  Unstable VCC_ROSC voltage level (SmartFusion only)  Signal integrity issues on JTAG pins.	Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your <a href="#">device datasheet</a> for more information on transient specifications.  Monitor VJTAG during programming; measure JTAG signals for noise or reflection.



Error Code	Exit Code	Exit Message	Possible Cause	Possible Solution
0x805D 0x805E 0x807B	10	Failed to enable FPGA Array.	Unstable VPUMP voltage level.  Unstable VCC  Unstable VCC_OSC (Fusion only)  Unstable VCC_ROSC voltage level (SmartFusion only)  Signal integrity issues on JTAG pins.	Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your <a href="#">device datasheet</a> for more information on transient specifications.  Monitor VJTAG during programming; measure JTAG signals for noise or reflection.
0x8095 0x8096 0x8097	10	Failed to disable FPGA Array.	Unstable VPUMP voltage level.  Unstable VCC  Unstable VCC_OSC (Fusion only)  Unstable VCC_ROSC voltage level (SmartFusion only)  Signal integrity issues on JTAG pins.	Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your <a href="#">device datasheet</a> for more information on transient specifications.  Monitor VJTAG during programming; measure JTAG signals for noise or reflection.
0x8061 0x8062	10	Failed to program FlashROM.	Unstable VPUMP voltage level.  Unstable VCC  Unstable VCC_OSC (Fusion only)  Unstable VCC_ROSC voltage level (SmartFusion only)  Signal integrity issues on JTAG pins.	Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your <a href="#">device datasheet</a> for more information on transient specifications.  Monitor VJTAG during programming; measure JTAG signals for noise or reflection.
0x801B 0x801C 0x806C	10	Error programming Embedded Flash	Unstable VCC_NVM/VCC_OS C voltage level	Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your

Error Code	Exit Code	Exit Message	Possible Cause	Possible Solution
0x806D 0x806E		Memory Block (EFMB)	(Fusion only)  Unstable VCC_ENVM/VCC_R COSC voltage level (SmartFusion only)  Signal integrity issues on JTAG pins  NVM corruption is possible when writing from your design; check the NVM status for confirmation.	<a href="#">device datasheet</a> for more information on transient specifications.  Monitor VJTAG during programming; measure JTAG signals for noise or reflection.  Reset signal is not properly tied off in your design.  <b>Inspect device using Device Debug.</b>
0x807D 0x807E	10	Error programming system init and boot clients	Unstable VCC  Unstable VCC_OSC (Fusion only)  Unstable VCC_ROSC voltage level (SmartFusion only)  Signal integrity issues on JTAG pins	Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your <a href="#">device datasheet</a> for more information on transient specifications.  Monitor VJTAG during programming; measure JTAG signals for noise or reflection.  <b>Inspect device using Device Debug.</b>
0x8069 0x806A 0x806B	10	Error programming Embedded Flash Memory Block (EMFB)	Programming file generated with an older version of software	Generate STAPL file with the latest version of Designer/FlashPro; use the latest version of FlashPro software  Try again at a slower TCK  <b>Inspect device using Device Debug.</b>  Contact Microsemi Technical Support
0x808E 0x808F 0x8090 0x8091	10	Error programming Embedded Flash Memory Block (EFMB)		Try reprogramming  Contact Microsemi Technical Support
0x807F 0x8080	10	Error programming system init and boot clients	Programming file generated with an older version of software	Generate STAPL file with the latest version of Designer/FlashPro; use the latest version of FlashPro software  Try again at a slower TCK  <b>Inspect device using Device Debug.</b>

Error Code	Exit Code	Exit Message	Possible Cause	Possible Solution
				Contact Microsemi Technical Support
0x8059 0x805B	11	Verify 0 failed at row <row number>  Verify 1 failed at row <row number>.	Unstable VPUMP voltage level.  Unstable VCC  Unstable VCC_OSC (Fusion only)  Unstable VCC_ROSC voltage level (SmartFusion only)  Signal integrity issues on JTAG pins.	Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your <a href="#">device datasheet</a> for more information on transient specifications.  Monitor VJTAG during programming; measure JTAG signals for noise or reflection.
0x8060	11	Failed to verify FlashROM at row <FlashROM row number>.	Device is programmed with a different design.  Unstable VPUMP voltage level.  Unstable VCC  Unstable VCC_OSC (Fusion only)  Unstable VCC_ROSC voltage level (SmartFusion only)  Signal integrity issues on JTAG pins.	Run VERIFY_DEVICE_INFO to verify the device is programmed with the correct data/design.  Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your <a href="#">device datasheet</a> for more information on transient specifications.  Monitor VJTAG during programming; measure JTAG signals for noise or reflection.
0x8075 0x8076 0x8077	11	Failed to verify Embedded Flash Memory Block (EFMB)	Device is programmed with a different design.  Unstable VCC  Unstable VCC_NVM/VCC_OSC (Fusion only)  Unstable VCC_ENVM/VCC_ROSC voltage level (SmartFusion only)	Verify the device is programmed with the correct data/design.  Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your <a href="#">device datasheet</a> for more information on transient specifications.  Measure JTAG pins, and noise or reflection.  Run DEVICE_INFO to confirm if the target EFMB block is locked with FlashLock (pass key). If the target EFMB block is locked, then

Error Code	Exit Code	Exit Message	Possible Cause	Possible Solution
			<p>Signal integrity issues on JTAG pins.</p> <p>The EFMB data was modified in your FPGA design after programming. This could have occurred during standalone verify.</p> <p>The target EFMB is locked with FlashLock when running ACTION PROGRAM_NVM_ACTIVE_ARRAY or VERIFY_NVM_ACTIVE_ARRAY.</p>	<p>you must unlock it by erasing the security and then reprogramming with the desired security settings. After unlocking the target EFMB block attempt to rerun the target ACTION.</p> <p><b>Inspect device using Device Debug.</b></p>
0x8085 0x8086	11	Failed to verify system init and boot clients	<p>Device is programmed with a different design.</p> <p>Unstable VCC</p> <p>Unstable VCC_NVM/VCC_OSC (Fusion only)</p> <p>Unstable VCC_ENVM/VCC_ROSC voltage level (SmartFusion only)</p> <p>Signal integrity issues on JTAG pins.</p> <p>The EFMB data was modified in your FPGA design after programming. This could have occurred during standalone verify.</p> <p>The target EFMB is locked with FlashLock when running ACTION PROGRAM_NVM_ACTIVE_ARRAY or</p>	<p>Verify the device is programmed with the correct data/design.</p> <p>Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your <a href="#">device datasheet</a> for more information on transient specifications.</p> <p>Measure JTAG pins, and noise or reflection.</p> <p>Run DEVICE_INFO to confirm if the target EFMB block is locked with FlashLock (pass key). If the target EFMB block is locked, then you must unlock it by erasing the security and then reprogramming with the desired security settings. After unlocking the target EFMB block attempt to rerun the target ACTION.</p> <p><b>Inspect device using Device Debug.</b></p>

Error Code	Exit Code	Exit Message	Possible Cause	Possible Solution
			VERIFY_NVM_ACTIVE_ARRAY.	
0x8072 0x8073 0x8074	11	Failed to verify Embedded Flash Memory Block (EFMB)	Programming file generated with an older version of software	<p>Generate STAPL file with the latest version of Designer/FlashPro; use the latest version of FlashPro software</p> <p>Try again at a slower TCK</p> <p><b>Inspect device using Device Debug.</b></p> <p>Contact Microsemi Technical Support</p>
0x8083 0x8084	11	Failed to verify system init and boot clients	Programming file generated with an older version of software	<p>Generate STAPL file with the latest version of Designer/FlashPro; use the latest version of FlashPro software</p> <p>Try again at a slower TCK</p> <p><b>Inspect device using Device Debug.</b></p> <p>Contact Microsemi Technical Support</p>
0x8014 0x8015	11	Failed to verify calibration data	<p>Unstable VCC</p> <p>Unstable VCC_NVM/VCC_OSC (Fusion only)</p> <p>Unstable VCC_ENVM/VCC_ROSC voltage level (SmartFusion only)</p> <p>Signal integrity issues on JTAG pins</p>	<p>Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your <a href="#">device datasheet</a> for more information on transient specifications.</p> <p>Monitor VJTAG during programming; measure JTAG signals for noise or reflection.</p> <p>Try reprogramming.</p> <p>Workaround: Disable optional procedure CHECK_AND_BACKUP_CALIB</p>
0x805A 0x805C	11	<p>Verify 0 failed at row &lt;row number&gt; .</p> <p>Verify 1 failed at row &lt;row number&gt;</p>	<p>Device is programmed with a different design</p> <p>Unstable VPUMP voltage level.</p> <p>Unstable VCC</p> <p>Unstable VCC_OSC (Fusion only)</p> <p>Unstable VCC_ROSC voltage level (SmartFusion only)</p>	<p>Run VERIFY_DEVICE_INFO to verify the device is programmed with the correct data/design.</p> <p>Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your <a href="#">device datasheet</a> for more information on transient specifications.</p> <p>Monitor VJTAG during programming; measure JTAG signals for noise or reflection.</p>

Error Code	Exit Code	Exit Message	Possible Cause	Possible Solution
			Signal integrity issues on JTAG pins	
0x8063	14	Failed to program Silicon Signature. Failed to program security lock settings.	Signal integrity issues on JTAG pins.	Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your <a href="#">device datasheet</a> for more information on transient specifications.  Monitor VJTAG during programming; measure JTAG signals for noise or reflection.
0x8068	-18	Failed to authenticate the encrypted data.	Incorrect AES key.  Signal integrity issues on JTAG pins.	Generate a programming file with the correct AES key.  Monitor VJTAG during programming; measure JTAG signals for noise or reflection.
0x805F	-20	Failed to verify FlashROM at row <FlashROM row number>.	Programming file generated with an older version of software  Device is programmed with a different design.  Unstable VPUMP voltage level.  Unstable VCC  Unstable VCC_OSC (Fusion only)  Unstable VCC_ROSC voltage level (SmartFusion only)  Signal integrity issues on JTAG pins.	Generate STAPL file with the latest version of Designer/FlashPro; use the latest version of FlashPro software.  Program with the correct data/design.  Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your <a href="#">device datasheet</a> for more information on transient specifications.  Measure JTAG pins and noise or reflection.
0x8065	-22	Failed to program pass key.	Unstable VPUMP voltage level.  Unstable VCC  Unstable VCC_OSC (Fusion only)  Unstable VCC_ROSC voltage	Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your <a href="#">device datasheet</a> for more information on transient specifications.  Monitor VJTAG during programming; measure JTAG signals for noise or reflection.

Error Code	Exit Code	Exit Message	Possible Cause	Possible Solution
			level (SmartFusion only)  Signal integrity issues on JTAG pins.	
0x8066	-23	Failed to program AES key.	Unstable VPUMP voltage level.  Unstable VCC  Unstable VCC_OSC (Fusion only)  Unstable VCC_ROSC voltage level (SmartFusion only)  Signal integrity issues on JTAG pins.	Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your <a href="#">device datasheet</a> for more information on transient specifications.  Measure JTAG pins and noise or reflection.
0x8055 0x8056	-24	Failed to program UROW.	Unstable VPUMP voltage level.  Unstable VCC  Unstable VCC_OSC (Fusion only)  Unstable VCC_ROSC voltage level (SmartFusion only)  Signal integrity issues on JTAG pins.	Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your <a href="#">device datasheet</a> for more information on transient specifications.  Monitor VJTAG during programming; measure JTAG signals for noise or reflection.  Make sure you mounted <b>0.01μF</b> and <b>0.33μF</b> caps on Vpump (close to the pin).
0x802A	-27	FlashROM Write/Erase is protected by the passkey. A valid passkey needs to be provided.	File contains no passkey and device is secured with a passkey.  Passkey in the file does not match device.	Provide a programming file with a passkey that matches the passkey programmed into the device.
0x8025	-28	FPGA Array Write/Erase is protected by the passkey. A valid pass key	File contains no passkey and device is secured with a passkey. Passkey in the file	Provide a programming file with a passkey that matches the passkey programmed into the device.

Error Code	Exit Code	Exit Message	Possible Cause	Possible Solution
		needs to be provided.	does not match device.	
0x802B 0x802D	-29	FlashROM Read is protected by passkey. A valid passkey needs to be provided.	File contains no passkey and device is secured with a passkey. Passkey in the file does not match device.	Provide a programming file with a pass key that matches the passkey programmed into the device
0x8024 0x8026	-30	FPGA Array verification is protected by a passkey. A valid passkey needs to be provided.	File contains no passkey and device is secured with a passkey. Passkey in the file does not match device.	Provide a programming file with a passkey that matches the passkey programmed into the device.
0x804B 0x8001 0x8007	-31	Failed to verify AES key.	<p>AES key in the file does not match the device.</p> <p>Unstable VCC</p> <p>Unstable VCC_OSC (Fusion only)</p> <p>Unstable VCC_ROSC voltage level (SmartFusion only)</p> <p>Unstable JTAG/VPUMP voltage level.</p>	<p>Provide a programming file with an AES key that matches the AES key programmed into the device.</p> <p>Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your <a href="#">device datasheet</a> for more information on transient specifications.</p> <p>Monitor VJTAG during programming; measure JTAG signals for noise or reflection.</p>
0x8000	-31	Failed to verify AES key.	Programming file generated with an older version of software	<p>Generate STAPL file with the latest version of Designer/FlashPro; use the latest version of FlashPro software.</p> <p>Try again at a slower TCK</p> <p>Contact Microsemi Technical Support</p>
0x8020 0x8022 0x8028	-33	FPGA Array encryption is enforced. A programming file with encrypted FPGA array data needs to be provided.	File contains unencrypted array data, but device contains AES key.	Provide a programming file with an encrypted FPGA Array data.
0x802C 0x802F	-34	FlashROM encryption is	File contains unencrypted	Provide a programming file with an encrypted FlashROM data.



Error Code	Exit Code	Exit Message	Possible Cause	Possible Solution
		enforced. A programming file with encrypted FlashROM data needs to be provided.	FlashROM data, but the device contains an AES key.	
0x801F 0x804A	-35	Failed to match pass key.	Pass key in file does not match pass key in device.	Provide a programming file with a pass key that matches the pass key programmed into the device.
0x802E 0x8030	-36	FlashROM Encryption is not enforced.  Cannot guarantee valid AES key present in target device.  Unable to proceed with Encrypted FlashROM programming.	File contains encrypted FlashROM, but device encryption is not enforced for FlashROM	Regenerate security programming file with proper AES key.  Program device security.  Retry programming FlashROM with encrypted programming file.
0x8021 0x8023 0x8027 0x8029	-37	FPGA Array Encryption is not enforced.  Cannot guarantee valid AES key present in target device.  Unable to proceed with Encrypted FPGA Array verification.	File contains encrypted FPGA Array, but the device encryption is not enforced for FPGA Array.	Regenerate security programming file with proper AES key.  Program device security.  Retry programming FPGA Array with encrypted programming file.
0x8067	-38	Failed to program pass key.	Unstable VPUMP voltage level.  Unstable VCC  Unstable VCC_OSC (Fusion only)  Unstable VCC_ROSC voltage level (SmartFusion only)  Signal integrity issues on JTAG pins. Bad device.	Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your <a href="#">device datasheet</a> for more information on transient specifications.  Measure JTAG pins and noise or reflection.

Error Code	Exit Code	Exit Message	Possible Cause	Possible Solution
0x806F 0x8070 0x8071 0x8081 0x8082 0x8089	-39	ERROR: 2 or more errors found on this page	Unstable VCC_NVM/VCC_OS C voltage (Fusion only)  Unstable VCC_ENVM/VCC_R OSC (SmartFusion only)  NVM reset signal is floating in user design  2 or more ECC errors found when reading the eNVM	Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your <a href="#">device datasheet</a> for more information on transient specifications.  Bias NVM reset to a logic state in user design.  Try reprogramming.
0x8010	-39	ERROR: 2 or more errors found on this page.	2 or more ECC errors found when reading the master calibration data	The master calibration data has been corrupted. Try restoring master calibration from backup, if it exists, by running RECOVER_CALIB.  Workaround: Disable optional procedure CHECK_AND_BACKUP_CALIB
0x8013	-39	ERROR: 2 or more errors found on this page.	2 or more ECC errors found when verifying the backup calibration	Rerun action to attempt to write backup calibration again.  Workaround: Disable optional procedure CHECK_AND_BACKUP_CALIB
0x8078 0x8079 0x807A 0x8087 0x8088	-40	Embedded Flash Memory Block MAC Failure.	Data in the file is encrypted with a different AES key than the device.	Verify the programming file is generated from the latest version of Designer/FlashPro.
0x8002 0x8003	-42	Failed to verify security settings.	File security settings do not match device.	Provide a programming file with security setting that match the security settings programmed into the device.
0x8093	-42	Failed to verify eNVM/EFMB client JTAG protection settings	Device eNVM/EFMB client JTAG protection settings are not programmed or are programmed with different settings	Verify the device is programmed with the correct eNVM/EFMB client JTAG protection settings
0x8004	-43	Failed to verify design information.	File checksum and design name do not match the device.	Verify the device is programmed with the correct data and design.

Error Code	Exit Code	Exit Message	Possible Cause	Possible Solution
0x8049	-44	Failed to verify AES key.	The AES key in the file does not match the AES key in the device. File does not contain an AES key and the device is secured with an AES key.	Provide a programming file with an AES key that matches the AES key programmed into the device.
0x8054	-45	Device package does not match the programming file.	Programming file was generated with an older version of software	Generate STAPL file with the latest version of Designer/FlashPro; use the latest version of FlashPro software.
0x8033 0x8038 0x803D 0x8042 0x8045 0x8046 0x8047 0x8048	-46	Embedded Flash Memory Block X Read is protected by pass key. A valid pass key needs to be provided.	File contains no pass key or incorrect pass key but EFMB read is secured with a pass key.	Provide a programming file with the correct pass key.
0x8034 0x8039 0x803E 0x8043	-46	Embedded Flash Memory Block (EFMB) block X Read is not protected by pass key.  EFMB content is not secure after encrypted programming.  Unable to proceed with encrypted NVM programming.	File contains encrypted EFMB for block X but the device encryption is not enforced for EFMB block X.	Regenerate security programming file with the proper AES key.  Program device security. Retry programming with EFMB block X with encrypted programming file.
0x8032 0x8037 0x803C 0x8041	-47	Embedded Flash Memory Block (EFMB) block X encryption is enforced. A programming file with encrypted EFMB data needs to be provided.	The programming EFMB data is not encrypted, but the device contains an AES key with encryption enforced.	Provide a programming file with encrypted EFMB data.
0x8031 0x8036 0x803B 0x8040	-48	Embedded Flash Memory Block (EFMB) block X Write is protected by pass key.	File contains no pass key or incorrect pass key, but device is secured with a pass key.	Provide a programming file with a passkey that matches the passkey programmed into the device.

Error Code	Exit Code	Exit Message	Possible Cause	Possible Solution
		A valid pass key needs to be provided.		
0x8035 0x803A 0x803F 0x8044	-49	<p>Embedded Flash Memory Block (EFMB) block X Encryption is not enforced.</p> <p>Cannot guarantee valid AES key present in target device.</p> <p>Unable to proceed with Encrypted EFMB programming.</p>	File contains encrypted EFMB for block X, but the device encryption is not enforced for EFMB block X.	<p>Regenerate security programming file with proper AES key.</p> <p>Program device security. Retry programming EFMB block X with encrypted programming file.</p>
0x801A	-50	No backup calibration data found or backup calibration data has been corrupted	No backup calibration copy has been made or the backup copy has been corrupted	<p>If master copy is still intact, rerun Action to create backup calibration copy.</p> <p>Workaround: Disable optional procedure CHECK_AND_BACKUP_CALIB</p>
8x804E	-51	Failed to access Embedded Flash Memory. (AFS600 only)	This version of the silicon does not support programming of the Embedded Flash Memory Block while the FPGA Array is active.	If programming the EFMB while the FPGA is active is not required, then use actions PROGRAM_NVM or VERIFY_NVM. Otherwise, use latest revision of silicon.
0x804F	-52	Failed to access Embedded Flash Memory. (AFS1500 only)	This version of the silicon does not support programming of the Embedded Flash Memory Block while the FPGA Array is active.	If programming the EFMB while the FPGA is active is not required, then use actions PROGRAM_NVM or VERIFY_NVM. Otherwise, use latest revision of silicon.
0x8050	-53	Failed to access Embedded Flash Memory. (AFS1500 only)	This version of the silicon does not support programming block 3 of the EFMBs while the FPGA Array is active.	If programming the EFMB while the FPGA is active is not required, then use actions PROGRAM_NVM or VERIFY_NVM. Otherwise, use EFMB blocks 0, 1, or 2, but do not use block 3.
0x8051	-54	Failed to access Embedded Flash Memory.	FPGA Array is accessing the target EFMB block while	If programming the EFMB while the FPGA is active is not required, then use actions PROGRAM_NVM or VERIFY_NVM. Otherwise,

Error Code	Exit Code	Exit Message	Possible Cause	Possible Solution
			attempting programming.  NVM reset signal is stuck in design.  Unstable VCC  MSS Clock is disabled during programming.  MSS Clock is not properly routed to the correct pin.	check the FPGA design or use a different EFMB block that is not being accessed. Check if target EFMB block logic is tied to reset.  Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your device datasheet for more information on transient specifications.  Verify that the NVM reset signal in the design is not stuck.  Verify the MSS clock is enabled during programming.  If the MSS clock is defined as an external I/O, then verify that it is properly routed to the correct pin.
0x808A 0x8094	-55	Failed to read Embedded Flash Memory Block (EFMB)	Programming file generated with an older version of software	Generate STAPL file with the latest version of Designer/FlashPro; use the latest version of FlashPro software  Try again at a slower TCK  Inspect device using Device Debug  Contact Microsemi Technical Support
0x808B	-55	Failed to read Embedded Flash Memory Block (EFMB)	Unstable VPUMP voltage level.  Unstable VCC  Unstable VCC_OSC (Fusion only)  Unstable VCC_ROSC voltage level (SmartFusion only)  Signal integrity issues on JTAG pins	Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your <a href="#">device datasheet</a> for more information on transient specifications.  Monitor VJTAG during programming; measure JTAG signals for noise or reflection.
0x808C	-55	Failed to read Embedded Flash Memory Block (EFMB)	Internal error	Contact Microsemi Technical Support
0x8011	-56	Failed to read calibration data		Try reprogramming.  Workaround: Disable optional procedure CHECK_AND_BACKUP_CALIB

Error Code	Exit Code	Exit Message	Possible Cause	Possible Solution
0x8012	-56	Failed to read calibration data	Unstable VCC  Unstable VCC_NVM/VCC_OSC (Fusion only)  Unstable VCC_ENVM/VCC_ROSC voltage level (SmartFusion only)  Signal integrity issues on JTAG pins	Monitor related power supplies that cause the issue during programming; check for transients outside of Microsemi specifications. See your <a href="#">device datasheet</a> for more information on transient specifications.  Measure JTAG voltages, noise, and reflection.  Try reprogramming.  Workaround: Disable optional backup procedure CHECK_BACKUP_CALIB
0x808D 0x8092	-57	eNVM/EFMB is protected by a Pass Key; you must provide a valid Pass Key	File contains no Pass Key and device is secured with a Pass Key  Pass Key in the file does not match device	Provide a programming file with a Pass Key that matches the Pass Key programmed into the device

## Exit Codes for pre-v8.6 Software (SmartFusion, IGLOO, ProASIC3 and Fusion)

The table below lists exit codes for SmartFusion, IGLOO, ProASIC3 and Fusion devices in pre-v8.6 software only. This includes v8.5 SP2, v8.5 SP1, v8.5, etc. See the [Device Exit Codes for Software v8.6 and Above](#) help topic for exit codes for older versions.

**Note:** Exit codes with positive integers are reserved for current and future standard EXIT codes of the STAPL standard. Exit codes with negative integers are reserved for vendor-specific EXIT codes.

Table 4 · Exit Codes for SmartFusion, IGLOO, ProASIC3 and Fusion Family Devices in pre-v8.6 Software

Exit Code	Exit Message	Possible Cause	Possible Solution
0	Passed (no error).		
1	A physical chain does not match the expected set up from the STAPL file. Also known as Checking Chain Error.	Physical chain configuration has been altered. Something has become disconnected in the chain. The specific IR length of non-Microsemi devices may be incorrect. The order of the specified chain may be incorrect.	
5	Failed to enter programming mode.	Unstable VPUMP voltage	Monitor VPUMP voltage during programming

Exit Code	Exit Message	Possible Cause	Possible Solution
		level.  Signal integrity issues on JTAG pins.  Older software or programming file used.	Measure JTAG voltages, noise, and reflection.  Generate STAPL file with the latest version of Designer/FlashPro. Use latest version of FlashPro software.
6	Failed to verify IDCODE.	Signal integrity issues on JTAG pins.	Measure JTAG pins, noise and reflection.
8	Failed Erase Operation.	Signal integrity issues on JTAG pins.	Monitor VPUMP voltage during programming. Measure JTAG voltages, noise, and reflection.
10	Failed to program FPGA array at row ", rowNumber, "."	Signal integrity issues on JTAG pins.	Monitor VPUMP voltage during programming. Measure JTAG voltages, noise, or reflection.
10	Failed to enable FPGA Array.	Signal integrity issues on JTAG pins.	Monitor VPUMP voltage during programming. Measure JTAG voltages, noise, or reflection.
10	Failed to program FlashROM.	Signal integrity issues on JTAG pins.	Monitor VPUMP voltage during programming. Measure JTAG voltages, noise, and reflection.
11	Verify 0 failed at row",rowNumber, "." Verify 1 failed at row",rowNumber, "." Failed to verify FlashROM at row",from rowNumber-1.	Device is programmed with a different design.  Signal integrity issues on JTAG pins.	Run VERIFY_DEVICE_INFO to verify the device is programmed with the correct data/design. Monitor VPUMP voltage during programming. Measure JTAG voltages, noise and reflection .
14	Failed to program Silicon Signature. Failed to program security lock settings.	Signal integrity issues on JTAG pins.	Monitor VPUMP voltage during programming. Measure JTAG voltages, noise, and reflection.
-18	Failed to authenticate the encrypted data.	Incorrect AES key.  Signal integrity issues on JTAG pins.	Generate a programming file with the correct AES key.  Measure JTAG voltages, noise and reflection
-20	Failed to verify FlashROM at row ", FROmRowNumber-1.	Device is programmed with a different design.  Signal integrity issues on JTAG pins.	Program with the correct data/design.  Monitor VPUMP level during programming. Measure JTAG pins and noise or reflection.
-22	Failed to program pass key.	Unstable VPUMP voltage level.  Signal integrity issues on JTAG pins.	Monitor <b>VPUMP</b> voltage during programming.  Measure JTAG voltages, noise, and reflection.

Exit Code	Exit Message	Possible Cause	Possible Solution
-23	Failed to program AES key.	Unstable VPUMP voltage level.  Signal integrity issues on JTAG pins.	Monitor VPUMP voltage during programming.  Measure JTAG pins and noise or reflection.
-24	Failed to program UROW.	Unstable VPUMP voltage level.  Signal integrity issues on JTAG pins.	Monitor VPUMP voltage during programming.  Measure JTAG voltages, noise, and reflection. Make sure you mounted <b>0.01iF</b> and <b>0.33iF</b> caps on Vpump (close to the pin).
-25	Failed to enter programming mode	Signal integrity issues on JTAG pins.	Measure JTAG voltages, noise, and reflection.
-26	Failed to enter programming mode	Signal integrity issues on JTAG pins.	Measure JTAG voltages, noise, and reflection.
-27	FlashROM Write/Erase is protected by the passkey. A valid passkey needs to be provided.	File contains no passkey and device is secured with a passkey. Passkey in the file does not match device.	Provide a programming file with a passkey that matches the passkey programmed into the device.
-28	FPGA Array Write/Erase is protected by the passkey. A valid pass key needs to be provided.	File contains no passkey and device is secured with a passkey. Passkey in the file does not match device.	Provide a programming file with a passkey that matches the passkey programmed into the device.
-29	FlashROM Read is protected by passkey. A valid passkey needs to be provided.	File contains no passkey and device is secured with a passkey. Passkey in the file does not match device.	Provide a programming file with a pass key that matches the passkey programmed into the device
-30	FPGA Array verification is protected by a passkey. A valid passkey needs to be provided.	File contains no passkey and device is secured with a passkey. Passkey in the file does not match device.	Provide a programming file with a passkey that matches the passkey programmed into the device.
-31	Failed to verify AES key.	AES key in the file does not match the device.  Unstable JTAG/VPUMP voltage level.	Provide a programming file with an AES key that matches the AES key programmed into the device.  Monitor VPUMP/VJTAG voltage during programming.  Measure JTAG voltages, noise, and reflection.
-32	Failed to verify IDCODE. Target is an M7 device	File is not for M7, but target device is an M7.  Signal integrity issues on JTAG pins.	Check that the target device is M7 enabled. Make sure programming file generated is for M7 enabled device.  Measure JTAG pins , noise, and reflection.



Exit Code	Exit Message	Possible Cause	Possible Solution
-32	Failed to verify IDCODE. Target is an M1 device	File is not for M1, but target device is an M1 device.  Signal integrity issues on JTAG pins.	Check that the target device is M1 enabled. Make sure programming file generated is for M1 enabled device.  Measure JTAG pins, noise, and reflection.
-32	Failed to verify IDCODE. Core enabled device detected	File is not for target device.  Signal integrity issues on JTAG pins	Check the target device. Make sure programming file generated for target device.  Measure JTAG voltages, noise, and reflection.
-32	Failed to verify IDCODE. The target is not an M7 device	File is for M7, but target device is not M7.  Signal integrity issues on JTAG pins.	Check that the target device is not M7 enabled. Make sure programming file generated is for non M7 enabled device.  Measure JTAG voltages, noise, and reflection.
-32	Failed to verify IDCODE. The target is not an M1 device	File is for M1, but target device is not an M1 device.  Signal integrity issues on JTAG pins.	Check that the target device is not M1 enabled. Make sure programming file generated is for non M1 enabled device.  Measure JTAG voltages, noise and reflection.
-33	FPGA Array encryption is enforced. A programming file with encrypted FPGA array data needs to be provided.	File contains unencrypted array data, but device contains AES key.	Provide a programming file with an encrypted FPGA Array data.
-34	FlashROM encryption is enforced. A programming file with encrypted FlashROM data needs to be provided.	File contains unencrypted FlashROM data, but the device contains an AES key.	Provide a programming file with an encrypted FlashROM data.
-35	Failed to match pass key.	Pass key in file does not match pass key in device.	Provide a programming file with a pass key that matches the pass key programmed into the device.
-36	FlashROM Encryption is not enforced.  Cannot guarantee valid AES key present in target device.  Unable to proceed with Encrypted FlashROM programming.	File contains encrypted FlashROM, but device encryption is not enforced for FlashROM	Regenerate security programming file with proper AES key.  Program device security.  Retry programming FlashROM with encrypted programming file.

Exit Code	Exit Message	Possible Cause	Possible Solution
-37	<p>FPGA Array Encryption is not enforced.</p> <p>Cannot guarantee valid AES key present in target device.</p> <p>Unable to proceed with Encrypted FPGA Array verification.</p>	<p>File contains encrypted FPGA Array, but the device encryption is not enforced for FPGA Array.</p>	<p>Regenerate security programming file with proper AES key.</p> <p>Program device security.</p> <p>Retry programming FPGA Array with encrypted programming file.</p>
-38	<p>Failed to program pass key.</p>	<p>Unstable VPUMP voltage level.</p> <p>Signal integrity issues on JTAG pins.</p> <p>Bad device.</p>	<p>Monitor VPUMP voltage during programming.</p> <p>Measure JTAG pins and noise or reflection.</p>
-39	<p>Failed to verify Embedded Flash Memory Block (EFMB).</p>	<p>Device is programmed with a different design. Signal integrity issues on JTAG pins.</p> <p>The EFMB data was modified through user FPGA design after programming; this could occur during standalone verify.</p> <p>The target EFMB block is locked with FlashLock when running ACTION PROGRAM_NVM_ACTIVE_ARRAY or VERIFY_NVM_ACTIVE_ARRAY.</p>	<p>Verify the device is programmed with the correct data/design.</p> <p>Monitor VPUMP voltage during programming.</p> <p>Measure JTAG pins and noise or reflection.</p> <p>Run DEVICE_INFO to confirm if the target EFMB block is locked with FlashLock (pass key). If the target EFMB block is locked, then you must unlock it by erasing the security and then reprogramming with the desired security settings. After unlocking the target EFMB block attempt to rerun the target ACTION.</p>
-40	<p>Embedded Flash Memory Block MAC Failure.</p>	<p>Data in the file is encrypted with a different AES key than the device.</p>	<p>Verify the programming file is generated from the latest version of Designer/FlashPro.</p>
-41	<p>Error programming Embedded Flash Memory Block. (EFMB)</p>	<p>Signal integrity issues on JTAG pins.</p>	<p>Measure JTAG pins and noise or reflection.</p>
-42	<p>Failed to verify security settings.</p>	<p>File security settings do not match device.</p>	<p>Provide a programming file with security setting that match the security settings programmed into the device.</p>
-43	<p>Failed to verify design information.</p>	<p>File checksum and design name do not match the device.</p>	<p>Verify the device is programmed with the correct data and design.</p>

Exit Code	Exit Message	Possible Cause	Possible Solution
-44	Failed to verify AES key.	The AES key in the file does not match the AES key in the device. File does not contain an AES key and the device is secured with an AES key.	Provide a programming file with an AES key that matches the AES key programmed into the device.
-45	Device package does not match the programming file.		
-46	Embedded Flash Memory Block X Read is protected by pass key. A valid pass key needs to be provided.	File contains no pass key or incorrect pass key but EFMB read is secured with a pass key.	Provide a programming file with the correct pass key.
-47	Embedded Flash Memory Block, block X encryption is enforced. A programming file with encrypted EFMB data needs to be provided.	The programming EFMB data is not encrypted, but the device contains an AES key with encryption enforced.	Provide a programming file with encrypted EFMB data.
-48	Embedded Flash Memory Block (EFMB) block X Write is protected by pass key.  A valid pass key needs to be provided.	File contains no pass key or incorrect pass key, but device is secured with a pass key.	Provide a programming file with a passkey that matches the passkey programmed into the device.
-49	Embedded Flash Memory Block (EFMB) block X Encryption is not enforced.  Cannot guarantee valid AES key present in target device.  Unable to proceed with Encrypted EFMB programming.	File contains encrypted EFMB for block X, but the device encryption is not enforced for EFMB block X.	Regenerate security programming file with proper AES key.  Program device security. Retry programming EFMB block X with encrypted programming file.
-51	Failed to access Embedded Flash Memory. (AFS600 only)	This version of the silicon does not support programming of the Embedded Flash Memory Block while the FPGA Array is active.	If programming the EFMB while the FPGA is active is not required, then use actions PROGRAM_NVM or VERIFY_NVM. Otherwise, use latest revision of silicon.

Exit Code	Exit Message	Possible Cause	Possible Solution
-52	Failed to access Embedded Flash Memory. (AFS1500 only)	This version of the silicon does not support programming of the Embedded Flash Memory Block while the FPGA Array is active.	If programming the EFMB while the FPGA is active is not required, then use actions PROGRAM_NVM or VERIFY_NVM. Otherwise, use latest revision of silicon.
-53	Failed to access Embedded Flash Memory. (AFS1500 only)	This version of the silicon does not support programming block 3 of the EFMBs while the FPGA Array is active.	If programming the EFMB while the FPGA is active is not required, then use actions PROGRAM_NVM or VERIFY_NVM. Otherwise, use EFMB blocks 0, 1, or 2, but do not use block 3.
-54	Failed to access Embedded Flash Memory.	FPGA Array is accessing the target EFMB block while attempting programming.  NVM reset signal is stuck in design.	If programming the EFMB while the FPGA is active is not required, then use actions PROGRAM_NVM or VERIFY_NVM. Otherwise, check the FPGA design or use a different EFMB block that is not being accessed. Check if target EFMB block logic is tied to reset.  Verify that the NVM reset signal in the design is not stuck.

## Exit Codes (ProASIC<sup>PLUS</sup>)

The table below lists the exit codes for ProASIC<sup>PLUS</sup> devices.

Table 5 · ProASIC<sup>PLUS</sup> Family Devices Exit Codes

Exit Code	Exit Message	Possible Cause	Possible Solution
0	This message means passed. This does not indicate an error.		
1	A physical chain does not match the expected set up from the STAPL file. Also known as Checking Chain Error.	Physical chain configuration has been altered. Something has become disconnected in the chain. The specific IR length of non-Microsemi devices may be incorrect. The order of the specified chain may be incorrect.	
2	There is a reading device ID failure.	The device either does not have a valid device ID or the data cannot be read correctly.	Check the device ID.
3	This occurs when using ProASIC <sup>PLUS</sup> devices.	Connect was set up for a ProASIC device and the device is actually ProASIC <sup>PLUS</sup> .	Set up for a ProASIC <sup>PLUS</sup> device.

Exit Code	Exit Message	Possible Cause	Possible Solution
5	Programming set up problem. Also known as Entering ISP Failure.	The A500K device senses the VDDL power supply as being on.	Power the VDDL down during programming. Check the device has the correct voltages on VDDP, VDDL, VPP, and VPN.
6	The IDCODE of the target device does not match the expected value in the STAPL file. This is a JEDEC standard message.	The device targeted in the STAPL file does not match the device being programmed. User selected wrong device. Device TRST pin is grounded. Noise or reflections on one or more of the JTAG pins caused by the IR Bits reading it back incorrectly.	Choose the correct STAPL file and select the correct device. Measure JTAG pins and noise or reflection. TRST should be floating or tied high. Cut down the extra length of ground connection.
7	Unknown algorithm: alg=x, prev=x Invalid data read from device	This occurs with current STAPL files when the revision written into the factory row is not rev 1 for ProASICPLUS devices. The STAPL files from last year may "exit 7" with newer devices or the older revision may cause this failure if the STAPL file used is from latest version. This error can also occur if the programmer has trouble reading the factory row due to signal noise, crosstalk, or reflections on the JTAG signal and clock lines. It can occur if you program a - F ProASICPLUS device with an old STAPL file. This error occurs if you connected VPP and VPN the wrong way. It occurs if there are no bypass Caps on VPP VPN, which damaged the device.  This error may occur if your power supply cannot source the correct current for programming.	Re-generate STAPL file from Designer 6.1 SP1. Double check VPP and VPN connections. Make sure VPP and VPN have correct bypass caps. Make sure that your power supply can deliver the correct current during programming.
8	FPGA failed during the erase operation.	The device is secured, and the corresponding STAPL file is not loaded. The device has been permanently secured and cannot be unlocked.	Load the correct STAPL file.

Exit Code	Exit Message	Possible Cause	Possible Solution
11	FPGA failed verify	The device is secured and the corresponding STAPL file is not loaded. You used the Libero IDE software v2.3 or earlier or the Designer R1-2003 software or earlier to generate the STAPL file. VPN caps were soldered in the wrong polarity.	Load the correct STAPL file. Use later software versions—at least Libero v2.3 SP1 and Designer R1-2003 SP1. Double-check the VPN bypass caps polarity.
12	Security is enabled.	The device is secured and the wrong key/STAPL file was entered. The device is damaged. The verification was interrupted and therefore fails, causing the software to think the device is secure.	
14	Program security failure.		
15	This is a factory Calibration Data CRC error.	During program, erase, or verify, you must read back Calibration Data from the FPGA. The data contains a CRC. You use the CRC to ensure the data is not corrupted/wrong. Device is damaged. Noise on the FTAL signals causes the programmer to read back wrong data.	
17	The device has been secured. Write-security is enabled.	The device is secured and the wrong key or STAPL file was entered. The device is damaged.	Load the correct STAPL file.
-54	Failed to access Embedded Flash Memory	Analog power supplies (Vcc15A, Vcc33A, GNDAQ and GNDA) are not connected.	Connect the analog power supplies (Vcc15A, Vcc33A, GNDAQ and GNDA)
-90	Unexpected RCK detected.	Noise on the RCK signal. You connected a CLK source to the RCK signal. The polarized bypass capacitors on VPP or VPN are reversed-biased and are affecting the programmer's VPP or VPN output voltage. This causes programming to fail. Several FlashPros are	Disconnect the RCK and make sure TCK has a clean signal. Separate FlashPros away from each other while they are programming Internal ISP. Connect programmer as an Admin in FlashPro.

Exit Code	Exit Message	Possible Cause	Possible Solution
		programming at the same time and are too close to each other. Programmer not properly installed by Admin.	
-91	Calibration data parity error.	Device is damaged.	Replace the device.
	Null	Several FlashPros are programming at the same time and are too close to each other. FlashPro connects to PC parallel port through a dongle key. Data length mismatch when performing DRSCAN on STAPL file.	
	Cable to target is not connected properly.	When the Analyze command is executed, the FlashPro looks for target devices. If the cable connection is wrong, FlashPro assumes that nothing is connected at all.	Confirm the connection between the header to the device. If the board supplies the power to the device, make sure the voltage level is correct.
	Chain integrity test failed: xx	The connection between the FlashPro programmer and the device is broken. The programmer cable might not be securely inserted into the header. The header is not connected to the JTAG pins of the FPGA correctly. The configuration setting (ProASIC/ProASICPLUS) does not match the target device. Noise or reflections on the JTAG pins has caused communication between the programmer and the device to fail. A dongle is plugged in between the PC parallel port and the FlashPro parallel port cable.	Secure the connections. Check the JTAG pins for signal activity. Check for broken TDO, TMS, and TCK pins. After checking all type of connections if the failure exists, you may need to replace the first device (the devices closest to the TDO of the programming header) in the chain. Remove the dongle.
	Could not connect to programmer on port lp1 or parallel port device does not support IEEE-1284 negotiation protocol	The remote device does not respond to the negotiation protocol, for a variety of reasons.	Make sure the port is connected. Make sure the connected device is a FlashPro/Lite programmer. Turn the programmer on. Check parallel port setting in BIOS. Make sure that there are no dongles in

Exit Code	Exit Message	Possible Cause	Possible Solution
			<p>between the parallel port and the FlashPro connection.</p> <p>Try another parallel cable, the parallel cable might be defective.</p> <p>Check to see if the programmer is damaged.</p> <p>Make sure the FlashPro Lite has power. The FlashPro Lite is powered from the target board through the Vdd pin of the programming header.</p> <p>Make sure the Vdd pin is connected and the target board is powered up.</p> <p>Secure the connection between the cable connector and the programming header.</p> <p>Before you program any devices, you should run the self-diagnostic test. The diagnostic software can be found on the Microsemi web site. If the test fails, please contact Microsemi Customer Technical Support at <a href="mailto:tech@Microsemi.com">tech@Microsemi.com</a> for credit and replacement.</p> <p>Note: The Self-test is only available for FlashPro, not FlashPro Lite.</p>
	External voltage detected on <Supply>	The voltage supply for the FPGA is driven by another source (board, external power-supply), but the user forgot to turn off the supply in the Connect menu.	Set appropriate options in the Connect menu.
	VDPP Disconnected.	<p>There is no Vddp voltage supply to the FPGA.</p> <p>You accidentally turned off the Vddp supply in the Connect menu.</p> <p>The Vddp supply on the board is not functioning.</p>	Check the Vddp supply on the board for appropriate voltages and correct the Connect menu.
	More than one unidentified device.		
	<p>If you want to perform an operation on the ProASIC device, the rest of the devices in the chain must be in bypass mode.</p> <p>To put devices in bypass mode, select Configuration &gt; Chain Parameter (or click the Chain Parameter button in the Single STAPL Configuration window), then set the</p>	<p>STAPL settings of Pre IR, Pre DR, Post IR, and Post DR do not match the chain configuration.</p> <p>One or more of the devices in the chain is damaged and the ID CODE cannot be read back.</p>	<p>Make sure you have set Pre IR, Pre DR, Post IR, and Post DR to match the chain configuration. If you are still experiencing the failure, it is likely that the device's ID CODE cannot be read and you need to replace the device.</p>



Exit Code	Exit Message	Possible Cause	Possible Solution
	Pre IR, Pre DR, Post IR or Post DR.		
	Cannot find the programmer with ID xxx	The programmer is removed from the PC.	Delete programmer (or reconnect programmer) and select the <b>Refresh Programmer</b> button. See Connecting Programmers for more information.
	Fatal Error: Please check programmer set up.	Software cannot resolve the error encountered in the programmer.	Save the project file, restart the software, and power cycle the programmer.
	External voltage xxx mV is detected on xxx.	You have specified the programmer to drive the xxx but external xxx is detected.	Deselect the xxx in the programmer setting.
	Executing action xxx failed.	The STAPL runtime failed.	
	Executing action xxx with serial index/action xx failed.	The STAPL runtime failed.	
	No Vpump voltage source is detected.		Select the Vpump in the Programmer setting. Make sure the external Vpump is properly turned on.
	Vpump short detected.		Use a different programmer. If the problem persists, check the board layout.
	xxx Mhz TCK frequency in this STAPL file is not supported by the FlashPro Lite detected. It supports only 4 MHz TCK frequency.		Check FlashPro Lite version being used. Use FlashPro Lite Rev C or modify the STAPL file to 4 MHz.
	xxx Mhz TCK frequency in this STAPL file is not supported by the FlashPro Lite RevC detected. It supports only 1, 2 or 4 Mhz TCK frequency.		Modify STAPL file to 1, 2, or 4 MHz.
	Cannot find the serial Index/Action xxx in STAPL file.	Mismatch between STAPL file and the Index/Action selection.	Make sure the STAPL file was not overwritten. Save the project with updated serial/action selection.
	Duplicated serial Index/Action xxx was removed.	Mismatch between STAPL file and the Index/Action selection.	Make sure the STAPL file was not overwritten. Save the project with updated serial/action selection.

Exit Code	Exit Message	Possible Cause	Possible Solution
	Using local backup copy xxx	Cannot find original copy.	Check for available space on the disk. Check that write permissions are enabled.
	FlashPro cannot rename the programmer/device with an existing name.	Name is already in use.	Create a new name.
	FlashPro cannot rename the programmer/device with an invalid character.	Invalid character used in programmer/device name.	Do not use invalid characters.
	Automatic check for updates.		FlashPro can check the Microsemi website to find if an updated version of the software is available. If you would like to have FlashPro automatically check for software updates, choose Preferences from the File menu. From the Updates tab, you can choose your automatic software update settings. You can also select Software Updates from the Help menu for updates to the FlashPro software.
	FlashPro parse error.	FlashPro software failed to parse the file.	
	FlashPro does not support STAPL files for xxx.	STAPL file not allowed.	Use a STAPL file for your device that is supported by FlashPro.

---

# SmartDebug

---

Microsemi's SmartDebug tool complements design simulation by allowing verification and troubleshooting at the hardware level.

SmartDebug can be run in the following modes:

- Integrated mode from the Libero Design Flow
- Standalone mode
- Demo mode

## Integrated Mode

When run in integrated mode from Libero, SmartDebug can access all design and programming hardware information. No extra setup step is required. In addition, the Probe Insertion feature is available in Debug FPGA Array.

To open SmartDebug in the Libero Design Flow window, expand **Debug Design** and double-click **SmartDebug Design**.

## Standalone Mode

SmartDebug can be installed separately in the setup containing FlashPro, FlashPro Express, and Job Manager. This provides a lean installation that includes all the programming and debug tools to be installed in a lab environment for debug. In this mode, SmartDebug is launched outside of the Libero Design Flow. When launched in standalone mode, you must go through SmartDebug project creation and import a Design Debug Data Container (DDC) file, exported from Libero, to access all debug features in the supported devices.

**Note:** In standalone mode, the Probe Insertion feature is not available in FPGA Array Debug, as it requires incremental routing to connect the user net to the specified I/O.

## Demo Mode

Demo mode allows you to experience SmartDebug features (Active Probe, Live Probe, Memory Blocks, **SERDES**) without connecting a board to the system running SmartDebug.

**Note:** SmartDebug demo mode is for demonstration purposes only, and does not provide the functionality of integrated mode or standalone mode.

**Note:** You cannot switch between demo mode and normal mode while SmartDebug is running.

**For detailed information about SmartDebug, refer to the SmartDebug User Guide.**

# Electrical Parameters

## DC Characteristics for FlashPro5/4/3/3X

**Note:** The target board must provide the VCC, VCCI, VPUMP, and VJTAG during programming. However, if there is only one ProASIC3 device on the target board, the FlashPro5/4/3/3X can provide the VPUMP power supply via the USB port.

**Note:** The VJTAG signal is driven from the target/DUT board. The VJTAG pin is sensed by the FP4 to configure the internal input and output buffers to the same IO Voltage levels. The VJTAG pin is only an input pin to the programmer.

Table 6 · DC Characteristic for FlashPro5/4/3/3X

Description	Symbol	Min	Max	Unit
Input low voltage, TDO	VIL	-0.5	0.35*VJTAG	V
Input high voltage, TDO	VIH	0.65*VJTAG	3.6	V
Input current, TDO	IIL, IIH	-20	+20	mA
Input capacitance, TDO			40	pF
Output voltage, VPUMP, operating	VPP	+3.0	+3.6	V
Output current, VPUMP	IPP		250	mA
VJTAG = 1.5V				
Output low voltage, TCK, TMS, TDI, 100µA load	VOL	0.0	0.2	V
Output low voltage, TCK, TMS, TDI, 4mA load	VOL	0.0	0.30*VJTAG	V
Output high voltage, TCK, TMS, TDI, 100µA load	V	VJTAG-0.2	VJTAG	V
Output high voltage, TCK, TMS, TDI, 4mA load	VOH	0.70*VJTAG	VJTAG	V
Output current, TCK, TMS, TDI	IOL, IOH	-4	+4	mA
VJTAG = 1.8V				
Output low voltage, TCK, TMS, TDI, 100µA load	VOL	0.0	0.2	V
Output low voltage, TCK, TMS, TDI, 6mA load	VOL	0.0	0.3	V
Output high voltage, TCK, TMS, TDI, 100µA load	VOH	VJTAG-0.2	VJTAG	V
Output high voltage, TCK, TMS, TDI, 6mA load	VOH	1.25	VJTAG	V
Output current, TCK, TMS, TDI	IOL, IOH	-6	+6	mA
VJTAG = 2.5V				
Output low voltage, TCK, TMS, TDI, 100µA load	VOL	0.0	0.2	V

Description	Symbol	Min	Max	Unit
Output low voltage, TCK, TMS, TDI, 8mA load	VOL	0.0	0.6	V
Output high voltage, TCK, TMS, TDI, 100µA load	VOH	VJTAG-0.2	VJTAG	V
Output high voltage, TCK, TMS, TDI, 8mA load	VOH	1.8	VJTAG	V
Output current, TCK, TMS, TDI	IOL, IOH	-8	+8	mA
VJTAG = 3.3V				
Output low voltage, TCK, TMS, TDI, 100µA load	VOL	0.0	0.2	V
Output low voltage, TCK, TMS, TDI, 8mA load	VOL	0.0	0.4	V
Output high voltage, TCK, TMS, TDI, 100µA load	VOH	VJTAG-0.2	VJTAG	V
Output high voltage, TCK, TMS, TDI, 8mA load	VOH	2.4	VJTAG	V
Output current, TCK, TMS, TDI	IOL, IOH	-8	+8	mA

## DC Characteristics for FlashPro Lite

Table 7 · DC Characteristic for FlashPro Lite

Description	Symbol	Min	Max	Unit
Input low voltage, TDO	VIL	-0.5	0.7	V
Input high voltage, TDO	VIH	1.7	5.0	V
Input current, TDO	IIL, IIH	-10	+10	µA
Input capacitance, TDO			40	pF
Input voltage, VDD, operating ( <a href="#">see note</a> )		+2.3	+3.5	V
Input voltage, VDD, power off		-1.0	+1.0	V
Input current, VDD	IVDD		500	mA
Output voltage, VPP, operating	VPP	+15.9	+16.5	V
Output voltage, VPN, operating	VPN	-13.8	-13.4	V
Output current, IPP	IPP	0	35	mA
Output current, IPN	IPN	0	-15	mA
Output low voltage, TCK, TMS, TDI, 100µA load	VOL	0.0	0.2	V
Output low voltage, TCK, TMS, TDI, 1mA load	VOL	0.0	0.5	V

Description	Symbol	Min	Max	Unit
Output low voltage, TCK, TMS, TDI, 2mA load	VOL	0.0	0.8	V
Output high voltage, TCK, TMS, TDI, 100uA load	VOH	2.1	2.5	V
Output high voltage, TCK, TMS, TDI, 1mA load	VOH	1.9	2.5	V
Output high voltage, TCK, TMS, TDI, 2mA load	VOH	1.6	2.5	V
Output current, TCK, TMS, TDI, nTRST	IOL, IOH	-2	+2	mA

**Note:** Up to 3.5 V can be supplied to the FlashPro Lite on the VDD pin. However, if the VDD supply for the FlashPro is also connected to the APA VDD supply, the voltage for the VDD pin cannot exceed 2.7 V.

## DC Characteristics for FlashPro

Table 8 • DC Characteristic for FlashPro

Description	Symbol	Min	Max	Unit
Input low voltage, TDO	VIL	-0.5	0.30 * VDDP	V
Input high voltage, TDO	VIH	0.70 * VDDP	5.5	V
Input current, TDO	IIL, IIH	-10	+10	uA
Input voltage, VDDP, VDDL		0	5.25	V
Input voltage, VPP		0	21.0	V
Input voltage, VPN		-21.0		V
Input current, VDDP, VDDL, VPN, VP	IVCC		5.0	mA
Output voltage range, VDDP	VDDP	1.5	3.3	V
Output voltage range, VPP	VPP	15.0	18.0	V
Output voltage range, VPN	VPN	-16.0	-12.0	V
Output voltage resolution / Accuracy			100 / $\pm 50$	mV
Output current, IDDP	IDDP	-135 <sup>1</sup>	+135	mA
Output current, IDDL	IDDL	-135 <sup>1</sup>	+135	mA
Output current, IPP	IPP	-270 <sup>1</sup>	+270	mA
Output current, IPN	IPN	-270	+270 <sup>1</sup>	mA
Output low voltage, TCK, TMS, TDI, OUT0, nTRST	VOL	0.0	0.4	V

Description	Symbol	Min	Max	Unit
Output high voltage, TCK, TMS, TDI, OUT0, nTRST	VOH	0.85 * VDDP	+ 0.3 VDDP	V
Output current, TCK, TMS, TDI, OUT0, nTRST	IOL, IOH	-12	+12	mA

**Note:** Note (1): When power supply mode is set to ABI\_GROUND.

**Note:** \* - If you want to power-up the device from the board power supply, clear the checkboxes for VDDL and VDDP. VPP and VPN are required during programming only and are supplied by the FlashPro programmer.

**Note:** (2) Microsemi does not have operating temperature information for the FlashPro programmer. FlashPro is intended to be used as lab or production equipment and not tested at extreme temperatures. All devices in the unit are commercial temp. FlashPro4 went through a burn-in cycle operating at 100C for 250 hours during quality testing. This involved repeatedly powering the programmers and then programming after the burn in; they are not actively programming during the burn in.

# Electrical Specifications

## FlashPro5

The FlashPro5 is a JTAG and a SPI based programmer for flash based Microsemi devices.

The FlashPro5 output is supplied via a connector to which a detachable 10-pin cable is fitted. The connector on the FlashPro5 unit is a 2x5, RA male Header connector, which is manufactured by AMP and has a manufacturer's part number of 103310-1. This is a standard 2x5, 0.1 pitch connector which is keyed. Use the 10 pin right-angle header, AMP P/N 103310-1 (DigiKey P/N A26285-ND) for FlashPro4 and use the 10 pin straight header, AMP P/N 103308-1 (DigiKey P/N A26267-ND) for the straight version.

The signals on the pins of the FlashPro5 10-pin connector are shown in the figure below.

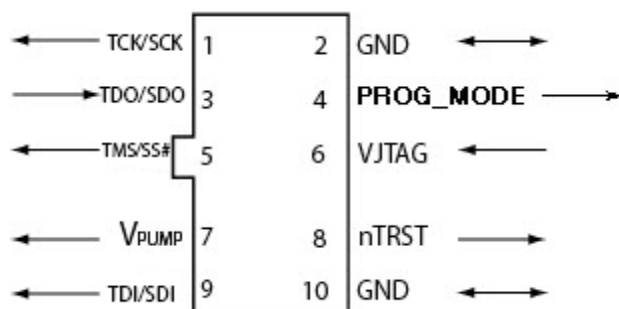


Figure 16 · FlashPro5 10-Pin Connector

**Note:** All ground pins must be connected. The rectangular shape shows connections on the programmer itself. Arrows show current flow towards or from the rectangular programmer.

The table below shows a description of the signals.

Table 9 · FlashPro5 Signal Description

Signal	Description
VPUMP	3.3V Programming voltage
GND	Signal reference
TCK/SCK	JTAG clock; SPI clock
TDI/SDI	JTAG data input to device; SPI MOSI
TDO/SDO	JTAG data output from device; SPI MISO
TMS/SS#	JTAG mode select; SPI Chip Select
nTRST	Programmable output pin may be set to off, toggle, low, or high level
VJTAG	Reference voltage from the target board
PROG_MODE	IGLOO v2 family - Used for switching from VCC 1.2V to 1.5V during programming

Some designers of high-integrity boards (military and avionic) may arrange their boards so that TRST is tied to ground via a weak pull-down resistor. The purpose of this is to hold the JTAG state-machine in a reset state by



default, so that even with TCK oscillating, some sudden ion bombardment or other electrical event will not suddenly throw the JTAG state-machine into an unknown state. If your design also uses a weak pull-down resistor on TRST on your board, then enabling the “Drive TRST” flag will be required to force the JTAG state-machine out of reset to permit programming to take place. With most boards, there is no need to select this flag.

# Electrical Specifications

## FlashPro4

The FlashPro4 output is supplied via a connector to which a detachable 10-pin cable is fitted. The connector on the FlashPro4 unit is a 2x5, RA male Header connector, which is manufactured by AMP and has a manufacturer's part number of 103310-1. This is a standard 2x5, 0.1 pitch connector which is keyed. Use the 10 pin right-angle header, AMP P/N 103310-1 (DigiKey P/N A26285-ND) for FlashPro4 and use the 10 pin straight header, AMP P/N 103308-1 (DigiKey P/N A26267-ND) for the straight version..

The signals on the pins of the FlashPro4 10-pin connector are shown in the figure below (extracted from FlashPro4 product specification):

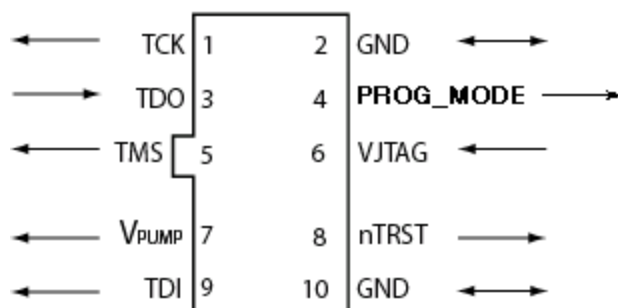


Figure 17 · FlashPro4 10-Pin Connector

**Note:** All ground pins must be connected. The rectangular shape shows connections on the programmer itself. Arrows show current flow towards or from the rectangular programmer.

The table below shows a description of the signals.

Table 10 · FlashPro4 Signal Description

Signal	Description
VPUMP	3.3V Programming voltage
GND	Signal reference
TCK	JTAG clock
TDI	JTAG data input to device
TDO	JTAG data output from device
TMS	JTAG mode select
nTRST	Programmable output pin may be set to off, toggle, low, or high level
VJTAG	Reference voltage from the target board
PROG_MODE	IGLOO v2 family - Used for switching from VCC 1.2V to 1.5V during programming

Some designers of high-integrity boards (military and avionic) may arrange their boards so that TRST is tied to ground via a weak pull-down resistor. The purpose of this is to hold the JTAG state-machine in a reset state by default, so that even with TCK oscillating, some sudden ion bombardment or other electrical even will not

suddenly throw the JTAG state-machine into an unknown state. If your design also uses a weak pull-down resistor on TRST on your board, then enabling the “Drive TRST” flag will be required to force the JTAG state-machine out of reset to permit programming to take place. With most boards, there is no need to select this flag.

## FlashPro3

The FlashPro3 output is supplied via a connector to which a detachable 10-pin cable is fitted. The connector on the FlashPro3 unit is a 2x5, RA male Header connector, which is manufactured by AMP and has a manufacturer's part number of 103310-1. This is a standard 2x5, 0.1 pitch connector which is keyed. Use the 10 pin right-angle header, AMP P/N 103310-1 (DigiKey P/N A26285-ND) for FlashPro5/4/3/3X and use the 10 pin straight header, AMP P/N 103308-1 (DigiKey P/N A26267-ND) for the straight version.

The signals on the pins of the FlashPro3 10-pin connector are shown in the figure below (extracted from FlashPro3 product specification):

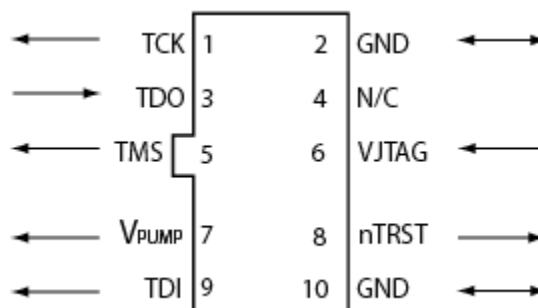


Figure 18 · FlashPro3 10-Pin Connector

**Note:** All ground pins must be connected. The rectangular shape shows connections on the programmer itself. Arrows show current flow towards or from the rectangular programmer.

The table below shows a description of the signals.

Table 11 · FlashPro3 Signal Description

Signal	Description
VPUMP	3.3V Programming voltage
GND	Signal reference
TCK	JTAG clock
TDI	JTAG data input to device
TDO	JTAG data output from device
TMS	JTAG mode select
nTRST	Programmable output pin may be set to off, toggle, low, or high level
VJTAG	Reference voltage from the target board
N/C	Programmer does not connect to this pin

Some designers of high-integrity boards (military and avionic) may arrange their boards so that TRST is tied to ground via a weak pull-down resistor. The purpose of this is to hold the JTAG state-machine in a reset state by default, so that even with TCK oscillating, some sudden ion bombardment or other electrical even will not

suddenly throw the JTAG state-machine into an unknown state. If your design also uses a weak pull-down resistor on TRST on your board, then enabling the “Drive TRST” flag will be required to force the JTAG state-machine out of reset to permit programming to take place. With most boards, there is no need to select this flag.

## FlashPro Lite

For FlashPro Lite, the existing 26-pin connector is shown in the figure below.

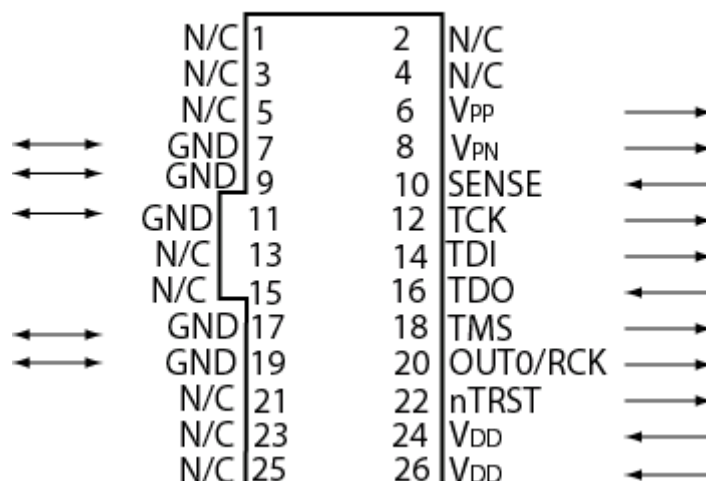


Figure 19 · 26-pin Connector for FlashPro Lite

**Note:** All ground pins must be connected. The rectangular shape shows connections on the programmer itself. Arrows show current flow towards or from the rectangular programmer.

The appropriate SAMTEC micro connector target cable for this is:

Samtec FFSD-13-D-12.00-01-N.

The 12 inch cable is specified. This is likely to be more than enough to connect to the board and reducing the inductance will help compared with 18 inches, which is supplied by the default with FlashPro Lite.

See the table below for a description of the signals.

Table 12 · FlashPro Lite Signal Description

Signal	Description
VDDP	VDD supply for logic I/O pads
VDDL	VDD supply for core
VPP	Positive programming supply (+16.5V)
VPN	Negative programming supply(-13.8V)
GND	Signal reference
SENSE	Input from target board to programmer to indicate connection to ground
TCK	JTAG clock
TDI	JTAG data input to device
TDO	JTAG data output from device
TMS	JTAG mode select

Signal	Description
nTRST	Programmable output pin may be set to off, toggle, low, or high level
RCK/OUT0	Programmable output pin may be set to off, toggle, low, or high level
N/C	Programmer does not connect to this pin

Some designers of high-integrity boards (military and avionic) may arrange their boards so that TRST is tied to ground via a weak pull-down resistor. The purpose of this is to hold the JTAG state-machine in a reset state by default, so that even with TCK oscillating, some sudden ion bombardment or other electrical even will not suddenly throw the JTAG state-machine into an unknown state. If your design also uses a weak pull-down resistor on TRST on your board, then enabling the "Drive TRST" flag will be required to force the JTAG state-machine out of reset to permit programming to take place. With most boards, there is no need to select this flag.

## FlashPro

For FlashPro, you can use the same 26-pin target cable you used for FlashPro Lite, but the connections are shown in the figure below.

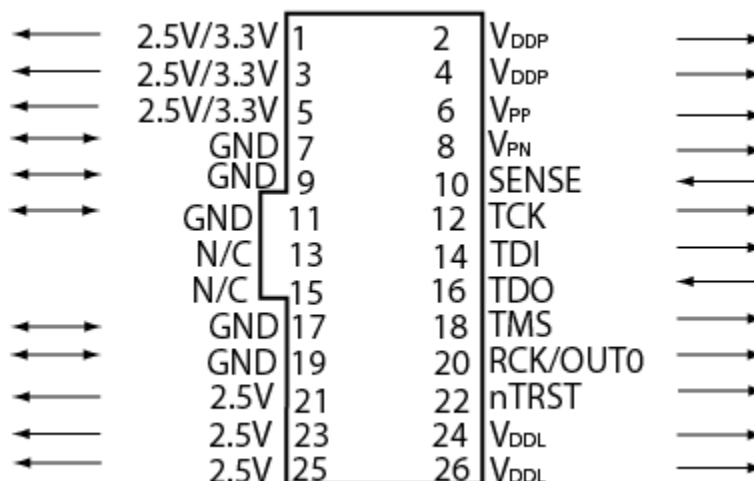


Figure 20 · 26-pin connections for FlashPro

**Note:** All ground pins must be connected. The rectangular shape shows connections on the programmer itself. Arrows show current flow towards or from the rectangular programmer.

The table below shows the signal pin descriptions for FlashPro.

Table 13 · FlashPro Signal Description

Signal	Description
VDDP	VDD supply for logic I/O pads
VDDL	VDD supply for core
VPP	Positive programming supply (+16.5 V)
VPN	Negative programming supply (-13.8 V)
GND	Signal reference
SENSE	Input from target board to programmer to indicate connection to ground

Signal	Description
TCK	JTAG clock
TDI	JTAG data input to device
TDO	JTAG data output from device
TMS	JTAG mode select
nTRST	Programmable output pin may be set to off, toggle, low, or high level
RCK/OUT0	Programmable output pin may be set to off, toggle, low, or high level
2.5V, 2.5V/3.3V, N/C	Programmer does not connect to these pins

Some designers of high-integrity boards (military and avionic) may arrange their boards so that TRST is tied to ground via a weak pull-down resistor. The purpose of this is to hold the JTAG state-machine in a reset state by default, so that even with TCK oscillating, some sudden ion bombardment or other electrical even will not suddenly throw the JTAG state-machine into an unknown state. If your design also uses a weak pull-down resistor on TRST on your board, then enabling the "Drive TRST" flag will be required to force the JTAG state-machine out of reset to permit programming to take place. With most boards, there is no need to select this flag.

## JTAG Switching Characteristics

### FlashPro 5/4/3/3X Characteristics

Table 14 · JTAG Switching Characteristics for FlashPro5/4/3/3X

Description	Symbol	Min	Max	Unit
Output delay from TCK to TDI, TMS	TTCKTDI	-2	2	ns
TDO setup time before TCK rising, VJTAG=3.3	TTDOTCK	12		ns
TDO setup time before TCK rising, VJTAG=1.5	TTDOTCK	14.5		ns
TDO hold time after TCK rising	TTCKTDO	0		ns
TCK period	TTCK	41.7	10667	ns

### FlashPro and FlashPro Lite Characteristics

The table below shows the JTAG switching characteristics for FlashPro and FlashPro Lite measured at the programmer end of the JTAG cable.

Table 15 · JTAG Switching Characteristics for FlashPro and FlashPro Lite

Description	Symbol	Min	Max	Unit
Output delay from TCK falling to TDI, TMS	TTCKTDI	-2	2	ns
TDO setup time before TCK rising	TTDOTCK	5.0		ns

Description	Symbol	Min	Max	Unit
TDO hold time after TCK rising	TTCKTDO	0		ns
TCK period	TTCK	40	10240	ns

## Illustration of the JTAG Switching Characteristics

The figure below is an illustration of the JTAG switching characteristics.

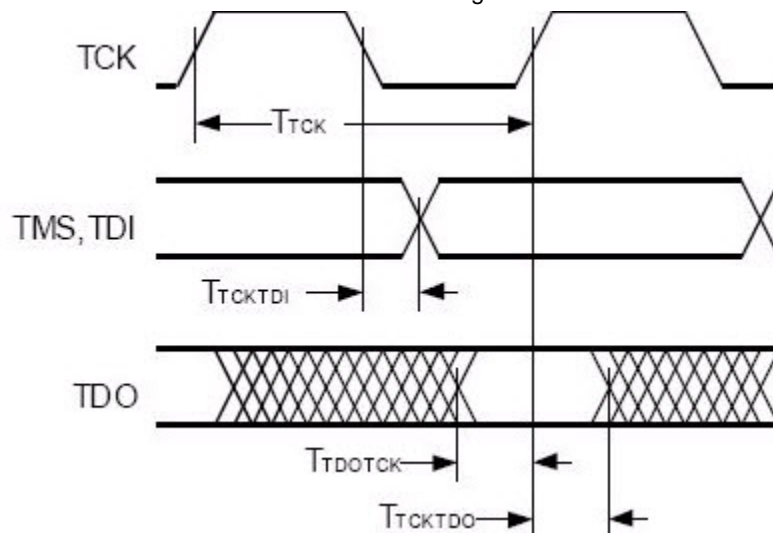


Figure 21 · JTAG Switching Characteristics

# FlashPro Express Reference

## FlashPro Express Start Page

The FlashPro Express Start Page is the first page to show when the tool starts up. This page provides the interface for loading a project into the tool by either navigating to the project location, or clicking on one of the recently opened projects.

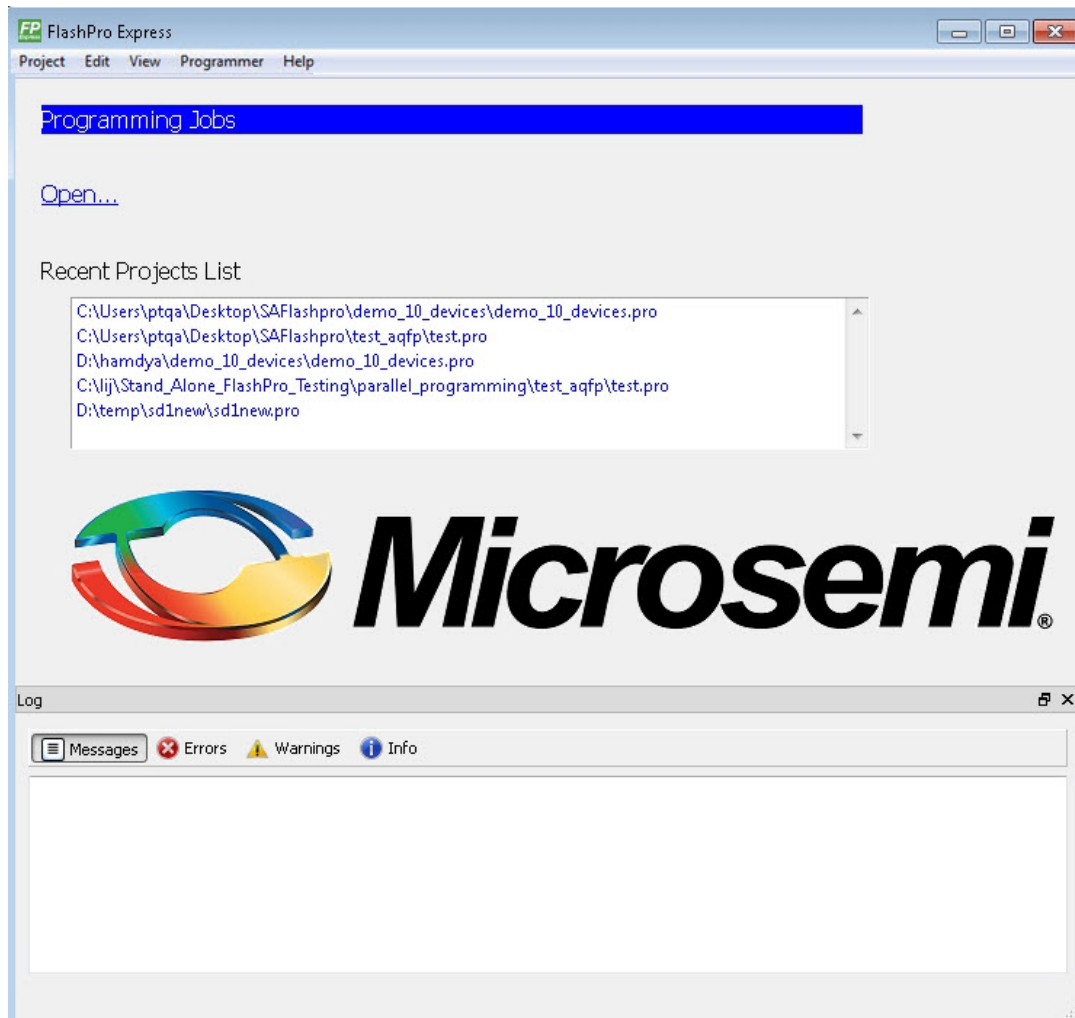


Figure 22 · FlashPro Express Start Page

## FlashPro Express Project Menu

Command	Function
Create Job Project from Programming Job	New job project folder with programming job name will be created at the specified location.



Command	Function
Open Job Project	Loads a job project into the tool by reading the information in user specified .pro file.
Close Job Project	Closes the current job project
Save Job Project	Saves the current job project
Set Log File	Sets the location of the Log file to your specified location.
Export Log File	Exports the Log file to your specified location.
Execute Script	Runs your specified Tcl script.
Export Script File	Exports all commands run in this session to your specified path as a Tcl script
Exit	Exits FlashPro Express

## FlashPro Express Edit Menu

Command	Function
Clear Log Window	Clears the Log window

## FlashPro Express View Menu

The View menu shows or hides the FlashPro Express GUI elements.

Command	Function
Log Window	Shows/hides the Log window

## FlashPro Express Tools Menu

Command	Function
Programmer Settings	Opens the Programmer Settings dialog box; enables you to set options for all types of supported Microsemi programmers
Run	Runs the current <a href="#">Programming Action</a> .

## FlashPro Express Help Menu

Command	Function
Help Topics	Opens the help
Microsemi Technical Support	Opens the Microsemi technical support site.
Microsemi Web Site	Opens the <a href="#">Microsemi website</a> in your default browser
User Guide	Opens the FlashPro Express User Guide.
Check for Software Updates	Checks for software updates (works only if you are connected to the internet)
About FlashPro Express	Lists the FlashPro Express release information

## FlashPro Express Log Window and Status Bar

### FlashProExpress Log Window

The FlashPro Express log window shows status messages for user activity. Click on the appropriate tab (Messages, Errors, Warning, Info) to filter messages by type.

- Use the right-click menu to copy text, clear the log, and scroll the log.
- Use the def variable LOG\_WINDOW\_BUFFER\_SIZE to set the buffer size.
- Use the View menu to show or hide the Log window.

### FlashProExpress Status Bar

The Status Bar at the bottom displays the status of the load project action.

---

# Contacting Microsemi

---

## Microsemi SoC Products Group Headquarters

Microsemi Corporation is a supplier of innovative programmable logic solutions, including field-programmable gate arrays (FPGAs) based on antifuse and flash technologies, high-performance intellectual property (IP) cores, software development tools, and design services targeted for the high-speed communications, application-specific integrated circuit (ASIC) replacement, and radiation-tolerant markets.

Address:	Microsemi SoC Products Group 3870 North First Street San Jose, CA 95134
Phone:	408-643-6000

## Contact Information

For the most up-to-date contact information, check the [Microsemi Home Page](#).

Contact information for FPGAs & SoCs can be found at the [FPGAs and SoCs Support Page](#)

If you do not have internet access, the following information was accurate at the time of publication:

- Technical Support
  - Web: <https://soc.microsemi.com/mycases>
  - Phone (NA): 800.262.1060
  - Phone (Int'l): +1 650.313.4460
  - Email: [soc\\_tech@microsemi.com](mailto:soc_tech@microsemi.com)
- Customer (non-technical) Support
  - Phone: +1 650.318.2470
  - Email: [customer.service@microsemi.com](mailto:customer.service@microsemi.com)
- Sales Support
  - For pricing, order status and lead time information for all Microsemi SoC products, contact your [Microsemi Sales Representative](#)
- Technical Support for RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR)
  - Phone (NA): 888.988.ITAR
  - Phone (Int'l): +1 650.318.4900
  - Email: [soc\\_tech\\_itar@microsemi.com](mailto:soc_tech_itar@microsemi.com)

## Regulatory and Compliance Information

EU Declaration of Conformity:

This product complies with Directive 2011/65/EU of the European Parliament and of the Council of 8 June 2011 on the restriction of the use of certain hazardous substances in electrical and electronic equipment.

To view the Declaration of Conformity in English:

[http://www.microsemi.com/index.php?option=com\\_docman&task=doc\\_download&gid=131772](http://www.microsemi.com/index.php?option=com_docman&task=doc_download&gid=131772)

Non-English:

[http://www.microsemi.com/index.php?option=com\\_docman&task=doc\\_download&gid=131748](http://www.microsemi.com/index.php?option=com_docman&task=doc_download&gid=131748)

Markings:



This product complies with 2004/108/EC, Electromagnetic Compatibility (EMC) Directive