



# **SVF TCL Library**

## **User's Guide**



## Table of Content

<b>Table of Content</b> .....	2
<b>Table of Tables</b> .....	2
<b>Table of Pictures</b> .....	2
<b>References</b> .....	3
<b>History</b> .....	3
1 Introduction .....	4
2 TCL Interpreter .....	5
2.1 Starting a TCL Session .....	5
2.2 Getting Information on TCL Procedures .....	7
3 SVF Library - SVFTclLib .....	8
3.1 Quick Reference Table .....	8
3.2 Procedure Detailed Description .....	9
3.2.1 Initialisation Procedure .....	9
SVFInit {} .....	9
3.2.2 SVF Procedures .....	9
ENDDR {State {Display 0}} .....	9
ENDIR {State {Display 0}} .....	9
FREQUENCY {{Freq 0}} .....	10
HDR { length {pTDI -1} {pTDO -1} {pMASK -1} {pSMASK -1} {Display 0}} .....	10
HIR { length {pTDI -1} {pTDO -1} {pMASK -1} {pSMASK -1} {Display 0}} .....	10
RUNTEST {run_state run_count run_clk min_time max_time end_state {Display 1}} .....	10
SDR { length {pTDI -1} {pTDO -1} {pMASK -1} {pSMASK -1} {Display 0}} .....	11
SIR { length {pTDI -1} {pTDO -1} {pMASK -1} {pSMASK -1} {Display 0}} .....	12
State {State} .....	12
TDR { length {pTDI -1} {pTDO -1} {pMASK -1} {pSMASK -1} {Display 0}} .....	12
TIR { length {pTDI -1} {pTDO -1} {pMASK -1} {pSMASK -1} {Display 0}} .....	13
TRST { trst_mode {Display 0}} .....	13
3.2.3 File Related Procedures .....	13
ExecSVFFile {FileName} .....	13
SVFFileLength {FileName} .....	13
3.3 Reference Sequences and Scripts .....	14

## Table of Tables

Table 1: Quick reference table of SVF procedures (by functionality) .....	8
---	---

## Table of Pictures

Figure 1: Tcl session start-up example .....	5
Figure 2: TCL console shortcut in 8PI Control Panel program group .....	6
Figure 3: Tcl console at start-up .....	6
Figure 4: Program group with TCL script examples .....	14



## References

[]

## History

Version	Date	Description
1.00	25-Oct-2005	Initial revision
1.01	02-Nov-2005	Reviewed
1.02	15-Feb-2006	Updated to comply with software version 1.03
1.03	13-Nov-2007	Update for GP Series introduction



## 1 Introduction

The objective of this document is to list and describe all the TCL procedures available in the SVF library provided to control the SVF operating mode of the GP Series device. Note that the SVF operating mode can be found on the JTAG page of the 8PI Control Panel. Each procedure functionality and parameters are described in detail. Scripts examples, which use some of these procedures, are also provided to help the users build their own test environments.

A section is also dedicated to the TCL interpreter provided with the *8PI Control Panel* application. The way to start it and to use it is briefly described.

## 2 TCL Interpreter

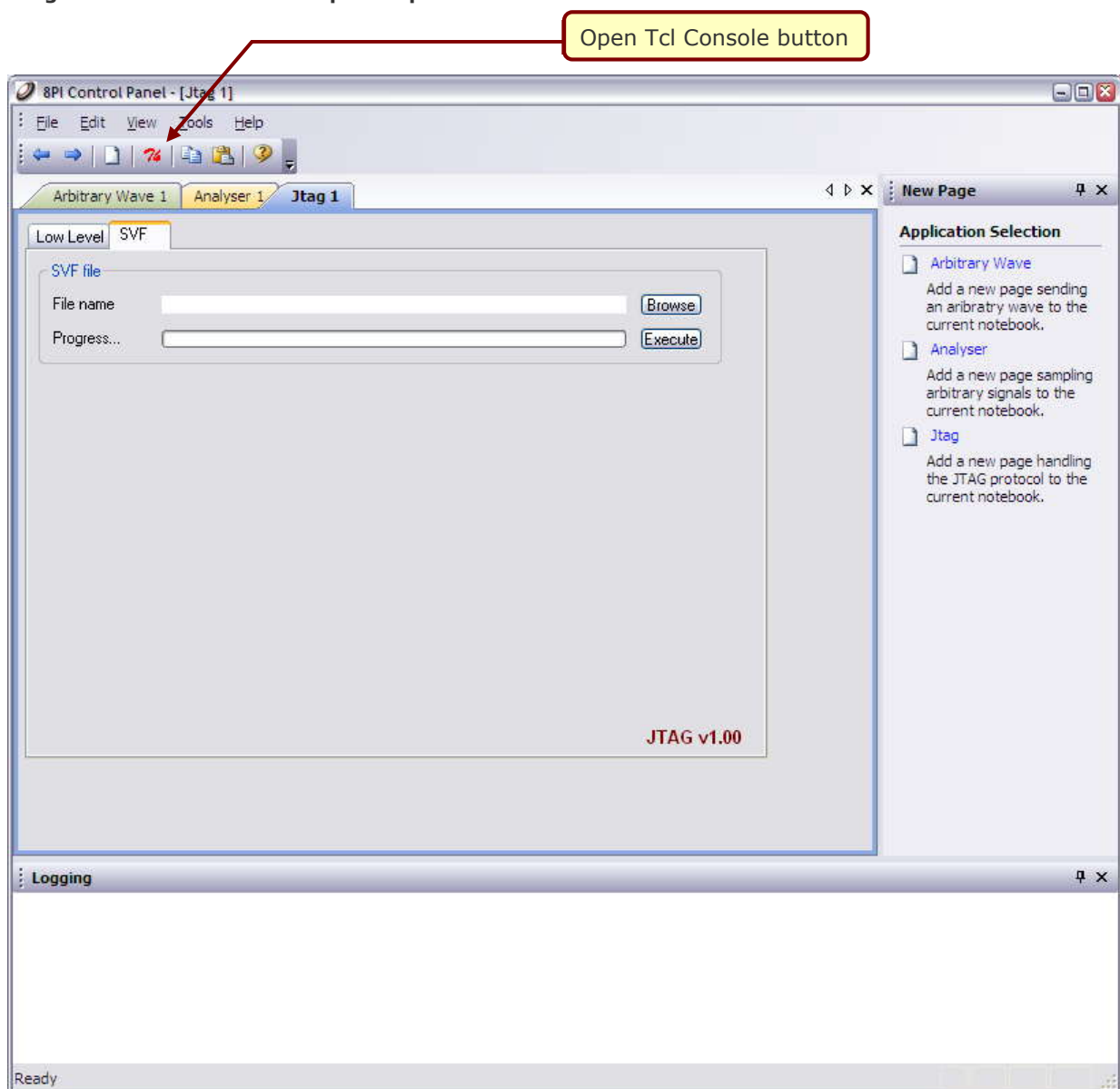
### 2.1 Starting a TCL Session

To start a TCL session from the 8PI Control Panel GUI:

1. From the 8PI Control Panel GUI, access the desired operating mode sheet.
2. Click on the 'Open Tcl Console' button (Figure 1).

This opens the Tcl console, loads the Tcl libraries relative to the chosen operating mode and initialises the Tcl session.

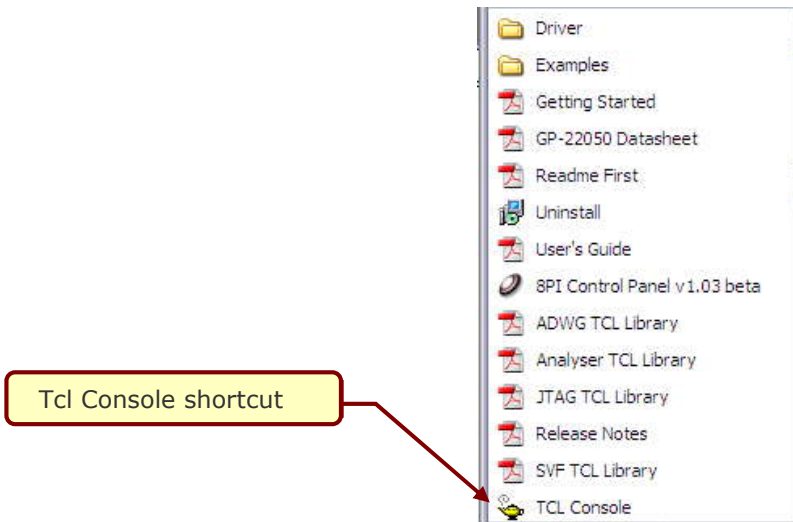
Figure 1: Tcl session start-up example



**To start a stand-alone TCL session (without running GUI):**

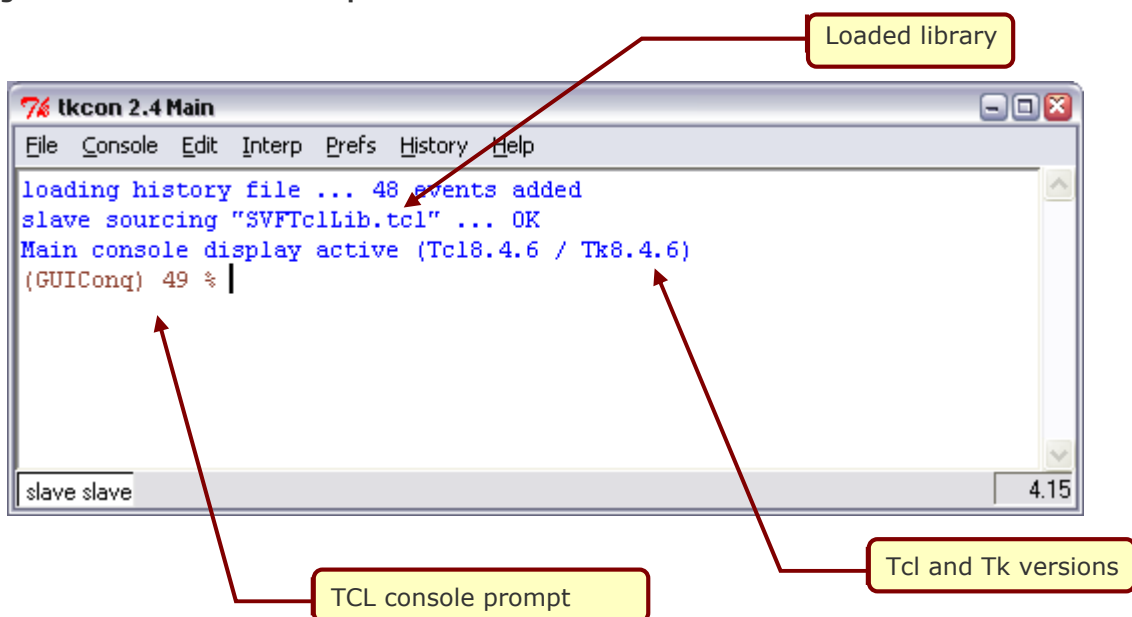
1. From the 'Byte Paradigm > 8PI Control Panel' program group, click on the 'TCL Console' shortcut. This starts the Wish84 interpreter with the tkcon console.
2. In the Tcl console, type: % source SVFTclLib.tcl  
This initialises your TCL session in Analyser operating mode.

**Figure 2: TCL console shortcut in 8PI Control Panel program group**



By default, the GP Series device software environment uses the WISH interpreter with the TKCON console (interactive mode) (Figure 3). For more information about the TKCON console, please check the following links: <http://tkcon.sourceforge.net/> - <http://wiki.tcl.tk/1878>. Please note that TCL is case sensitive.

**Figure 3: Tcl console at start-up**





## 2.2 Getting Information on TCL Procedures

Tcl provides built-in commands for getting information about the elements loaded in memory during a Tcl session. We simply describe a few of them for those unfamiliar to the Tcl language.

***To list the libraries loaded in the TCL environment:***

```
% info loaded
```

***To list all the procedures loaded in the TCL environment:***

```
% info procs
```

***To list the arguments of a given procedure:***

```
% info args <procedure name>
```

***To list the body of a given procedure:***

```
% info body <procedure name>
```

To learn more about the TCL/TK language, numerous man pages, tutorials and references can be found at the following location: <http://www.tcl.tk/doc/> .

## 3 SVF Library - SVFTclLib

This library contains all the procedures available to control the GP Series device when operating in SVF mode. Using these procedures in scripts or command lines allows the user to control any JTAG compliant Test Access Port (TAP) controller

### 3.1 Quick Reference Table

Table 1 gives a list of all procedures available for the SVF mode. They are grouped by type and level of abstraction.

**Table 1: Quick reference table of SVF procedures (by functionality)**

Procedures	Description
<b>Initialisation Procedures</b>	
SVFInit {}	SVF mode initialisation. This procedure is called automatically when the Tcl shell is started.
<b>SVF procedures</b>	
ENDDR {State {Display 0}}	Stable state after DR <sup>1</sup> shift.
ENDIR {State {Display 0}}	Stable state after IR <sup>2</sup> shift.
FREQUENCY {{Freq 0}}	Maximum TCK frequency.
HDR {length {pTDI -1} {pTDO -1} {pMASK -1} {pSMASK -1} {Display 0}}	Data pre-amble configuration.
HIR {length {pTDI -1} {pTDO -1} {pMASK -1} {pSMASK -1} {Display 0}}	Instruction pre-amble configuration.
RUNTEST {run state run count run clk min_time max_time end_state {Display 1}}	Forces to a state for a number of clock cycles.
SDR {length {pTDI -1} {pTDO -1} {pMASK -1} {pSMASK -1} {Display 0}}	Scans data register.
SIR {length {pTDI -1} {pTDO -1} {pMASK -1} {pSMASK -1} {Display 0}}	Scans instruction register.
STATE {State}	Forces to a state.
TDR {length {pTDI -1} {pTDO -1} {pMASK -1} {pSMASK -1} {Display 0}}	Data post-amble configuration.
TIR {length {pTDI -1} {pTDO -1} {pMASK -1} {pSMASK -1} {Display 0}}	Instruction post-amble configuration.
TRST {trst_mode {Display 0}}	Sets the TRST signal status.
<b>File Related Procedures</b>	
ExecSVFFile {FileName}	Executes an SVF file.
SVFFileLength {FileName}	Returns the number of SVF commands in the SVF file.

<sup>1</sup> DR : JTAG TAP controller data register

<sup>2</sup> IR : JTAG TAP controller instruction register

## 3.2 Procedure Detailed Description

This section gives a detailed description of each procedure available to control the GP Series device SVF operating mode. The procedures are listed in alphabetic order.

### 3.2.1 Initialisation Procedure

#### SVFInit { }

*parameters:* None  
*returns:*  
*description:* Downloads the configuration to the GP Series device and prepares it to operate in SVF mode.  
*conditions:* None  
*see also:*

### 3.2.2 SVF Procedures

#### ENDDR {State {Display 0}}

*parameters:* *State:* Stable state. The stable state can be entered as a full state name or an encoding, as described hereafter:

<i>State Name</i>	<i>or Encoding</i>
RESET	0
IDLE	1
DRSELECT	2
DRCAPTURE	3
DRSHIFT	4
DREXIT1	5
DRPAUSE	6
DREXIT2	7
DRUPDATE	8
IRSELECT	9
IRCAPTURE	10
IRSHIFT	11
IREXIT1	12
IRPAUSE	13
IREXIT2	14
IRUPDATE	15

*Display:* Optional. Displays feedback in the tcl console if 1.

*returns:*

*description:* The ENDDR command specifies the IEEE 1149.1 stable state that the bus will be forced to at the conclusion of a DR scan. Once specified, the ENDDR command remains in force until overridden by another ENDDR command. At start-up, ENDDR is set to IDLE.

*conditions:* None

*see also:* *ENDIR{ }*

#### ENDIR {State {Display 0}}

*parameters:* *State:* Stable state. The stable state is an integer value and must be entered as encoded in the list given for the ENDDR procedure.

*Display:* Optional. Displays feedback in the tcl console if 1.

*returns:*

*description:* The ENDIR command specifies the IEEE 1149.1 stable state that the bus



will be forced to at the conclusion of a IR scan. Once specified, the ENDIR command remains in force until overridden by another ENDIR command. At start-up, ENDIR is set to IDLE.

*conditions:* None  
*see also:* ENDIR{ }

### FREQUENCY {{Freq 0}}

*parameters:* *Freq:* Maximum TCK rate in Hz  
*returns:*  
*description:* Establishes a maximum IEEE 1149.1 test clock (TCK) frequency for subsequent scans (SDR and SIR), state changes (STATE), and test operations (RUNTEST). The new frequency remains in effect until the next FREQUENCY statement is executed or the end of file is reached.  
*conditions:* None  
*see also:*

### HDR { length {pTDI -1} {pTDO -1} {pMASK -1} {pSMASK -1} {Display 0}}

*parameters:* *length:* Number of bits to be shifted. Setting the length to 0 removes the header.  
*pTDI:* Optional. Reserved for future extensions.  
*pTDO:* Optional. Reserved for future extensions.  
*pMASK:* Optional. Reserved for future extensions.  
*pSMASK:* Optional. Reserved for future extensions.  
*Display:* Optional. Displays feedback in the tcl console if 1.  
*returns:* int -1 on error, 0 otherwise  
*description:* Specifies a default header pattern (pre-amble) length that is shifted in before every data scan operation.  
*conditions:* None  
*see also:* HIR { length {pTDI -1} {pTDO -1} {pMASK -1} {pSMASK -1} {Display 0}}

### HIR { length {pTDI -1} {pTDO -1} {pMASK -1} {pSMASK -1} {Display 0}}

*parameters:* *length:* Number of bits to be shifted. Setting the length to 0 removes the header.  
*pTDI:* Optional. Reserved for future extensions.  
*pTDO:* Optional. Reserved for future extensions.  
*pMASK:* Optional. Reserved for future extensions.  
*pSMASK:* Optional. Reserved for future extensions.  
*Display:* Optional. Displays feedback in the tcl console if 1.  
*returns:* int -1 on error, 0 otherwise  
*description:* Specifies a default header pattern (pre-amble) length that is shifted in before every instruction scan operation.  
*conditions:* None  
*see also:* HDR { length {pTDI -1} {pTDO -1} {pMASK -1} {pSMASK -1} {Display 0}}

### RUNTEST {run\_state run\_count run\_clk min\_time max\_time end\_state {Display 1}}

*parameters:* *run\_state:* The stable IEEE 1149.1 state that the bus will be forced to during the RUNTEST command. Valid run states are IRPAUSE, DRPAUSE, RESET, and IDLE. If the test bus is already in the run state, no state transition occurs. Once a run\_state is specified, subsequent RUNTEST commands will default to the same run state if one is not specified. The initial default is IDLE.



*count\_run*: The number of clocks that the IEEE 1149.1 bus will remain in the run state.

*run\_clk*: Specifies the clock used, either TCK (Test Clock) or SCK (System Clock). Must always be set to TCK.

*min\_state*: The minimum amount of time in seconds that the RUNTEST command must execute before finishing.

*max\_state*: The maximum amount of time in seconds that the RUNTEST command may execute before it must finish. The maximum time must be greater than the minimum time. If both *run\_count* and *max\_time* are specified, and not all clocking has completed when the *max\_time* is reached, the command finishes even though not all the requested clocking has been performed.

*end\_state*: The stable IEEE 1149.1 state that the 1149.1 bus will be forced to after executing the specified number of clocks, waiting the specified length of time, or both. Valid end states are IRPAUSE, DRPAUSE, RESET, and IDLE. If the test bus is already in the end state, no state transition occurs. If the *end\_state* is not specified, the default end state is used. When an *end\_state* is specified, it becomes the default. When a *run\_state* is specified, the new *run\_state* becomes the default *end\_state*. When a *run\_state* is not specified, the default *end\_state* remains valid. The initial default for *end\_state* is IDLE.

*Display*: Optional. Displays feedback in the tcl console if 1.

*returns*: int -1 on error, 0 otherwise

*description*: Forces the target IEEE 1149.1 bus to the specified run state for a specified number of clocks, a specified length of time, or both, then moves the bus to the specified end state.

*conditions*: None

*see also*:

**SDR { length {pTDI -1} {pTDO -1} {pMASK -1} {pSMASK -1} {Display 0}}**

*parameters*:

*length*: Number of data bits to be shifted.

*pTDI*: Optional. The value to be scanned into the target, expressed as a hex value. If this parameter is not present, the value of TDI to be scanned into the target will be all zero.

*pTDO*: Optional. The values to be compared against the actual values scanned out of the target, expressed as a hex string. This value will be bit-wise ORed with the actual values scanned out of the target. If this parameter is not present, no comparison will be performed. If no TDO parameter is present, the MASK will not be used.

*pMASK*: Optional. The mask to be used when comparing TDO values against the actual values scanned out of the target, expressed as a hex string. A '1' in a specific bit position indicates a care for that position. A '0' indicates a don't care. If no TDO parameter is present, the MASK will not be used.

*pSMASK*: Optional. Reserved for future extensions.

*Display*: Optional. Displays feedback in the tcl console if 1.

*returns*: int -1 on error, 0 otherwise

*description*: Specifies a scan pattern to be applied to the target data scan register.

*conditions*: None

*see also*: SIR { length {pTDI -1} {pTDO -1} {pMASK -1} {pSMASK -1} {Display 0}}



### SIR { length {pTDI -1} {pTDO -1} {pMASK -1} {pSMASK -1} {Display 0}}

*parameters:*    *length:*    Number of instruction bits to be shifted.  
                  *pTDI:*        Optional. The value to be scanned into the target, expressed as a hex value. If this parameter is not present, the value of TDI to be scanned into the target will be all zero.  
  
                  *pTDO:*        Optional. The values to be compared against the actual values scanned out of the target, expressed as a hex string. This value will be bit-wise ORed with the actual values scanned out of the target. If this parameter is not present, no comparison will be performed. If no TDO parameter is present, the MASK will not be used.  
  
                  *pMASK:*        Optional. The mask to be used when comparing TDO values against the actual values scanned out of the target, expressed as a hex string. A '1' in a specific bit position indicates a care for that position. A '0' indicates a don't care. If no TDO parameter is present, the MASK will not be used.  
  
                  *pSMASK:*      Optional. Reserved for future extensions.  
                  *Display:*      Optional. Displays feedback in the tcl console if 1.

*returns:*        int     -1 on error, 0 otherwise

*description:*    Specifies a scan pattern to be applied to the target instruction scan register.

*conditions:*    None

*see also:*        SDR { length {pTDI -1} {pTDO -1} {pMASK -1} {pSMASK -1} {Display 0}}

### State {State}

*parameters:*    *State:*        A list of one or more IEEE 1149.1 states describing the explicit path through the Test Access Port (TAP) state diagram that is taken in order to reach the final stable state. Valid states are RESET, IDLE, DRSELECT, DRCAPTURE, DRSHIFT, DRPAUSE, DREXIT1, DREXIT2, DRUPDATE, IRSELECT, IRCAPTURE, IRSHIFT, IRPAUSE, IREXIT1, IREXIT2, and IRUPDATE. The states must be listed in an order that complies with the TAP state diagram. If all intermediate states are not listed, a default path is assumed based on the current state and the final stable state. The last state must be one of the following states: IRPAUSE, DRPAUSE, RESET, and IDLE. When the list of states counts more than one single element, they must be given as a space-separated list in brackets (example: State {IDLE RESET DRCAPTURE IDLE}).

*returns:*        int     always 0

*description:*    Forces the bus to a stable IEEE 1149.1 state.

*conditions:*    None

*see also:*        None

### TDR { length {pTDI -1} {pTDO -1} {pMASK -1} {pSMASK -1} {Display 0}}

*parameters:*    *length:*        Number of bits to be shifted. Setting the length to 0 removes the trailer.  
  
                  *pTDI:*        Optional. Reserved for future extensions.  
                  *pTDO:*        Optional. Reserved for future extensions.  
                  *pMASK:*        Optional. Reserved for future extensions.  
                  *pSMASK:*      Optional. Reserved for future extensions.  
                  *Display:*      Optional. Displays feedback in the tcl console if 1.

*returns:*        int     -1 on error, 0 otherwise

*description:*    Specifies a default trailer pattern (post-amble) length that is shifted in after all subsequent data scan operation.



*conditions:* None  
*see also:* `TIR { length {pTDI -1} {pTDO -1} {pMASK -1} {pSMASK -1} {Display 0}}`

### **TIR { length {pTDI -1} {pTDO -1} {pMASK -1} {pSMASK -1} {Display 0}}**

*parameters:* `length:` Number of bits to be shifted. Setting the length to 0 removes the trailer.  
`pTDI:` Optional. Reserved for future extensions.  
`pTDO:` Optional. Reserved for future extensions.  
`pMASK:` Optional. Reserved for future extensions.  
`pSMASK:` Optional. Reserved for future extensions.  
`Display:` Optional. Displays feedback in the tcl console if 1.

*returns:* int -1 on error, 0 otherwise

*description:* Specifies a default trailer pattern (post-amble) length that is shifted in after all subsequent instruction scan operation.

*conditions:* None  
*see also:* `TDR { length {pTDI -1} {pTDO -1} {pMASK -1} {pSMASK -1} {Display 0}}`

### **TRST { trst\_mode {Display 0}}**

*parameters:* `trst_mode:` Specifies whether the TRST line is active, inactive, is high impedance or even exists. Valid trst\_mode states are:

State Name	Encoding
ON	Active (Logic 0)
OFF	Inactive (Logic 1)
Z	High Impedance
ABSENT	Not present

`Display:` Optional. Displays feedback if in the tcl console 1.

*returns:*

*description:* The TRST command describes the operation of the optional Test Reset signal.

*conditions:* None  
*see also:*

## **3.2.3 File Related Procedures**

### **ExecSVFFile {FileName}**

*parameters:* `FileName:` SVF file filename.

*returns:* int -2 if file not found  
-1 if no filename is empty  
0 no error

*description:* Executes the command in the specified SVF file.

*conditions:* None  
*see also:*

### **SVFFileLength {FileName}**

*parameters:* `FileName:` SVF file filename.

*returns:* int The number commands in the SVF file.

*description:* Returns the number of SVF commands in the SVF file.

*conditions:* None  
*see also:*

### 3.3 Reference Sequences and Scripts

Examples of the TCL scripts are provided with the TCL library. They can be accessed in the *Examples* group located under the *8PI Control Panel* program group created in your start menu during the installation of the application and driver on your computer.

**Figure 4: Program group with TCL script examples**

