

PSpice TCL Command Reference

Product Version 23.1

September 2023

© 2023 Cadence Design Systems, Inc.
Printed in the United States of America.

Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA.

Open SystemC, Open SystemC Initiative, OSCI, SystemC, and SystemC Initiative are trademarks or registered trademarks of Open SystemC Initiative, Inc. in the United States and other countries and are used with permission.

Trademarks: Trademarks and service marks of Cadence Design Systems, Inc. (Cadence) contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence's trademarks, contact the corporate legal department at the address shown above or call 800.862.4522.

All other trademarks are the property of their respective holders.

Restricted Permission: This publication is protected by copyright law and international treaties and contains trade secrets and proprietary information owned by Cadence. Unauthorized reproduction or distribution of this publication, or any portion of it, may result in civil and criminal penalties. Except as specified in this permission statement, this publication may not be copied, reproduced, modified, published, uploaded, posted, transmitted, or distributed in any way, without prior written permission from Cadence. Unless otherwise agreed to by Cadence in writing, this statement grants Cadence customers permission to print one (1) hard copy of this publication subject to the following conditions:

1. The publication may be used only in accordance with a written agreement between Cadence and its customer.
2. The publication may not be modified in any way.
3. Any authorized copy of the publication or portion thereof must include all original copyright, trademark, and other proprietary notices and this permission statement.
4. The information contained in this document cannot be used in the development of like products or software, whether for internal or external use, and shall not be used for the benefit of any other party, whether or not for consideration.

Disclaimer: Information in this publication is subject to change without notice and does not represent a commitment on the part of Cadence. Except as may be explicitly set forth in such agreement, Cadence does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy or usefulness of the information contained in this document. Cadence does not warrant that use of such information will not infringe any third party rights, nor does Cadence assume any liability for damages or costs of any kind that may result from use of such information. Cadence is committed to using respectful language in our code and communications. We are also active in the removal and replacement of inappropriate language from existing content. This product documentation may however contain material that is no longer considered appropriate but still reflects long-standing industry terminology. Such content will be addressed at a time when the related software can be updated without end-user impact.

Restricted Rights: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor.

Contents

1	7
Introduction	7
2	8
Accessing PSpice using TCL Commands	8
PSpiceACSetup	9
PSpiceBanner	10
PSpiceBuildLibIndex	11
PSpiceCommandDo	11
PSpiceCurrentTime	12
PSpiceDoAnalysis	13
PSpiceGetModel	14
PSpiceGetVoltage	15
PSpiceParamGetValue	16
PSpiceParamSetValue	17
PSpiceSaveSimulationState	18
PSpiceSetCallback	19
PSpiceSetLicenseBatchMode	20
PSpiceSetProbeTitle	21
PSpiceSetSimulationTemperature	22
PSpiceSetupAnalysis	23
PSpiceTranEnd	23
PSpiceTranNextTime	24
PSpiceTranRun	25
SetAppendDatMode	26
PSpiceSimulationPercent	27
PSpiceSimulationTime	28
ConvergenceFailure	28
3	30
Accessing PSpice Data Files using TCL Commands	30
addBlock	31
addDataName	31
addDataPoint	32

addTimePoint	33
calculateBH	34
close	35
closeBlock	36
closeFile	37
create	37
Evaluate	38
get	39
getDataNames	40
getJSON	41
getNumberOfBlocks	42
loadFile	43
setBlock	43
setHeaderData	44
4	46
Accessing PSpice Advanced Analysis Design Database using TCL commands	46
PSAARReader_enableAAInst_TclEval	46
PSAARReader_enableAAInstDevSmoke	47
PSAARReader_getAAInstParam	48
PSAARReader_getAAInstParam_Val	49
PSAARReader_getAAVar	50
PSAARReader_getPSpiceInstParam	51
PSAARReader_getPSpiceParam	52
PSAARReader_openDesign	53
PSAARReader_save	54
PSAARReader_setAADevParam	55
PSAARReader_setAAInstDevParam	55
PSAARReader_setAAInstParam	56
PSAARReader_setAAInstParam_Tol	57
PSAARReader_setAAVar	58
PSAARReader_setPSpiceInstParam	59
PSAARReader_setPSpiceParam	60
5	62
Customizing Various Analysis in PSpice Advanced Analysis using TCL	

Commands	62
_cdnPspAAAnalysisComplete	62
_cdnPspAAProcessDesignStart	63
_cdnPspAAProcessInst	64
_cdnPspAAProcessPostInst	64
_cdnPspAASmokeDerateReliability	65
_cdnPspAASmokeDerateTemp	66
_cdnPspAASmokePostPDM&TB	67
_cdnPspAASmokePostPDM&TJ	68
_cdnPspAASmokePostPDML&TJL	69
6	70
Accessing PSpice Advanced Analysis User Interface using TCL Commands	70
pspAAGetActiveAap	70
pspAAGetAssociated	71
pspAAOpenAssociated	72
pspAASelectAap	73
7	74
Executing Advanced Analysis using TCL Commands	74
TCL Commands to Execute PSpice Advanced Analysis	74
addSpecifications	75
getMonteCarloSettings	76
getMonteController	77
getOptimizerController	77
getPplotController	78
getSensitivityController	79
getSmokeController	79
openAAProfile	80
pspAAcreateInterface	81
pspAAdeleteInterface	82
saveAAProfile	82
saveAssociatedProfile	83
setCurrentEngine	84
setCurrentGear	85
setMonteCarloSettings	86
setSettings	86
setSpecifications	87

TCL Commands for the Monte Carlo Advanced Analysis	88
delete_CMonteSettings	88
delete_CMonteSpecification	89
new_CMonteSettings	90
new_CMonteSpecification	90
TCL Commands for the Sensitivity Advanced Analysis	91
delete_CSensParameter	91
delete_CSensSpecification	92
new_CSensParameter	93
new_CSensSpecification	94
TCL Commands for the Smoke Advanced Analysis	94
TCL Commands for the Optimizer Advanced Analysis	95
TCL Commands for the Parametric Plotter Advanced Analysis	95

Introduction

The PSpice TCL Command Reference document provides set of various PSpice TCL commands that can be used to automate different tasks, such as opening a design, running transient analysis, simulating the design with Monte Carlo analysis etc. This document covers TCL commands for the following tasks:


- [Accessing PSpice](#)
- [Accessing PSpice Data Files](#)
- [Accessing PSpice Advanced Analysis User Interface](#)
- [Accessing PSpice Advanced Analysis Design Database](#)
- [Executing Various Advanced Analysis](#)
- [Customizing Various Analysis in PSpice Advanced Analysis](#)

Accessing PSpice using TCL Commands

Following are the TCL commands that can be used to access PSpice from PSpice command window.

- [PSpiceACSetup](#)
- [PSpiceBanner](#)
- [PSpiceBuildLibIndex](#)
- [PSpiceCommandDo](#)
- [PSpiceCurrentTime](#)
- [PSpiceDoAnalysis](#)
- [PSpiceGetModel](#)
- [PSpiceGetVoltage](#)
- [PSpiceParamGetValue](#)
- [PSpiceParamSetValue](#)
- [PSpiceSaveSimulationState](#)
- [PSpiceSetCallback](#)
- [PSpiceSetLicenseBatchMode](#)
- [PSpiceSetProbeTitle](#)
- [PSpiceSetSimulationTemperature](#)
- [PSpiceSetupAnalysis](#)
- [PSpiceTranEnd](#)
- [PSpiceTranNextTime](#)
- [PSpiceTranRun](#)
- [SetAppendDatMode](#)
- [PSpiceSimulationPercent](#)
- [PSpiceSimulationTime](#)

- [ConvergenceFailure](#)

 **Important**
Before using these commands, ensure that you load `orPSP_ENG64.dll` or `orPIODData64.dll` in the command window. For examples, see [PSpice TCL Sample Scripts](#).

PSpiceACSetup

Sets up an AC Analysis

Return

int

Syntax

PSpiceACSetup <SweepType> <Steps> <StartFreq> <EndFreq>

Parameters

Parameter	Description	Type	Optional	Value Required
SweepType	The available SweepType include: 1 for Linear 2 for Octave 3 for Decade	int	false	false
Steps	The total number of steps	int	false	false
StartFreq	The start frequency for AC Analysis	int	false	false
EndFreq	The end frequency for an AC Analysis	int	false	false

Examples

```
PSpiceACSetup 1 100 100 10e6
```

Related Commands

[PSpiceSetupAnalysis](#)

PSpiceBanner

Updates the data section banner in the output(.out) file.

Return

None

Syntax

```
PSpiceBanner <bannerText> <run>
```

Parameters

Parameter	Description	Type	Optional	Value Required
bannerText	Any valid string	string	false	false
run	The number of data section	int	false	false

Examples

```
BoolTest true
```

Related Commands

[PSpiceSetProbeTitle](#)

PSpiceBuildLibIndex

Rebuilds the library index

Return

int

Syntax

```
PSpiceBuildLibIndex <filename>
```

Parameters

Parameter	Description	Type	Optional	Value Required
filename	File Name	string	false	false

Examples

```
BoolTest true
```

Related Commands

No Related Command

PSpiceCommandDo

Executes command to Run Simulation

Return

bool

Syntax

PSpiceCommandDo <command> <args>

Parameters

Parameter	Description	Type	Optional	Value Required
command	One of the following: doACAN for AC Analysis doDCAN for DC Analysis doTRAN for Transient Analysis doALL for ALL the Analyses FINISH for finishing the analysis	enum	false	false
args	One of the following: true for Initilization required false for Initialization not required	bool	false	false

Examples

- PSpiceCommandDo DoTRAN true
- PSpiceCommandDo FINISH true

Related Commands

No Related Command

PSpiceCurrentTime

Returns Current Simulation Time (For use during callback processing)

Return

int

Syntax

```
PSpiceCurrentTime
```

Parameters

No Parameters

Examples

```
BoolTest true
```

Related Commands

[PSpiceTranNextTime](#)

PSpiceDoAnalysis

Run all analyses written in the circuit file

Return

bool

Syntax

```
PSpiceDoAnalysis <cirFileName> <outFileName> <datFileName> <libpath>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<code>cirFileName</code>	Circuit File Name	To Be Added	false	false
<code>outFileName</code>	Output File Name	To Be Added	false	false
<code>datFileName</code>	Probe Data File Name	To Be Added	false	false
<code>libpath</code>	Location to search for PSpice library files	To Be Added	false	false

Examples

```
BoolTest true
```

Related Commands

- [PSpiceSetupAnalysis](#)
- [PSpiceCommandDo](#)

PSpiceGetModel

Model Name or sub-circuit name for the specified instance.

Return

string

Syntax

```
PSpiceGetModel <Instance>
```

Parameters

Parameter	Description	Type	Optional	Value Required
Instance	Instance Name	string	false	false

Examples

```
BoolTest true
```

Related Commands

[PSpiceGetVoltage](#)

PSpiceGetVoltage

Returns the current voltage value for the specified canonical node name

Return

int

Syntax

```
PSpiceGetVoltage pNodeName
```

Parameters

Parameter	Description	Type	Optional	Value Required
pNodeName	The canonical node name for which voltage or current value is required	string	false	false

Examples

```
PSpiceGetVoltage N1
```

Related Commands

No Related Commands

PSpiceParamGetValue

Gets the Parameter value from the simulator

Return

int

Syntax

```
PSpiceParamGetValue <name> <pMode=0>
```

Parameters

Parameter	Description	Type	Optional	Value Required
Name	Parameter Name	string	false	false
Mode	Mode can be one of the following: 0 for Effective Value 1 for instance value 2 for Model value	int	false	false

Examples

```
BoolTest true
```


Related Commands

[PSpiceParamSetValue](#)

PSpiceParamSetValue

Assigns the value of the named parameter, only if valid this function will succeed if the parameters are valid, otherwise it will fail.

Return

int

Syntax

```
PSpiceParamSetValue <name> <value> <ModelParam=false>
```

Parameters

Parameter	Description	Type	Optional	Value Required
name	Parameter Name, which can be one of the following: Instance, Global, or Model. Predefined Global Parameters are as follows: RELTOL, ABSTOL, VNTOL, GMIN, TSTOP, TMAX, ITL1, ITL2, ITL4, TNOM, TEMP(for Temperature), AUTOCONVERGE, AutoConverge.ITL1, AutoConverge.ITL, AutoConverge.ITL4, AutoConverge.RELTOL, AutoConverge.ABSTOL, AutoConverge.VNTOL, AutoConverge.PIVTOL, AutoConverge.RESTART, METHOD, STEPGMIN, GMINSTEPS, NOSTEPSRC, ITL6, ADVCONV, PSEUDOTRAN, TRTOL, GMINSRC, NOGMINI, PTRANSTEP, NOSTEPDEP, BRKDEPSRC	string	false	false
value	Parameter Value	int	false	false
ModelParam	False update on Instance or Global only	bool	false	false

Examples

- `PSpiceParamSetValue R1.value 1000`
- `PSpiceParamSetValue X1.R1.value 2 true`

Related Commands

[PSpiceParamGetValue](#)

PSpiceSaveSimulationState

Save the simulation check-point for a transient analysis.

Return

bool

Syntax

```
PSpiceSaveSimulationState <FileName>
```

Parameters

Parameter	Description	Type	Optional	Value Required
FileName	Filename of the saved data	string	false	false

Examples

```
BoolTest true
```

Related Commands

No Related Command

PSpiceSetCallback

Sets up the callback in simulator for different conditions

Return

int

Syntax

```
PSpiceSetCallback <callback function>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<code>callback_function</code>	Callback function can be on of the following: PSpiceTCLCallbacks::ConvergenceFailure PSpiceTCLCallbacks::PreInterrupt PSpiceTCLCallbacks::PostInterrupt Note: Package PSpiceTCLCallbacks needs to be defined	string	false	false

Examples

```
PSpiceSetCallback cbk
```

Related Commands

PSpiceSetLicenseBatchMode

Sets and uses the specified license string

Return

int

Syntax

```
PSpiceSetLicenseBatchMode <PreferredFeature>
```

Parameters

Parameter	Description	Type	Optional	Value Required
PreferredFeature	License Feature Name	string	false	false

Examples

```
PSpiceSetLicenseBatchMode PSpiceAD
```

Related Commands

No Related Command

PSpiceSetProbeTitle

Sets title to the probe section.

Return

None

Syntax

```
PSpiceSetProbeTitle <Title>
```

Parameters

Parameter	Description	Type	Optional	Value Required
Title	Any valid string	string	false	false

Examples

```
PSpiceSetProbeTitle "R1=1000"
```

Related Commands

[PSpiceBanner](#)

PSpiceSetSimulationTemperature

Sets the simulation temperature

Return

int

Syntax

```
PSpiceSetSimulationTemperature <Temperature>
```

Parameters

Parameter	Description	Type	Optional	Value Required
Temperature	Temparature in Celsius degree	int	false	false

Examples

```
PSpiceSetSimulationTemperature 37
```

Related Commands

[PSpiceCommandDo](#)

PSpiceSetupAnalysis

Read the circuit file into the PSpice simulator

Return

bool

Syntax

```
PSpiceSetupAnalysis <cirFileName> <outFileName> <datFileName> <libpath>
```

Parameters

Parameter	Description	Type	Optional	Value Required
cirFileName	Circuit File Name	string	false	false
outFileName	Output File Name	string	false	false
datFileName	Probe Data File Name	string	false	false
libpath	Location to search for PSpice library files	string	false	false

Examples

```
PSpiceSetupAnalysis rc.cir rc.out rc.dat "<installation Path>\tools\pspice\library"
```

Related Commands

[PSpiceACSetup](#)

PSpiceTranEnd

Command marks the transient simulation run as complete. Probe and output files data sections are flushed and closed.

Return

bool

Syntax

```
PSpiceTranEnd
```

Parameters

No Parameters

Examples

```
PSpiceTranEnd
```

Related Commands

- [PSpiceTranRun](#)
- [PSpiceTranNextTime](#)

PSpiceTranNextTime

Returns the next time point needed by PSpice simulation (For use during callback processing).

Return

int

Syntax

```
PSpiceTranNextTime
```


Parameters

No Parameters

Examples

```
BoolTest true
```

Related Commands

- [PSpiceTranEnd](#)
- [PSpiceTranRun](#)

PSpiceTranRun

Runs transient analysis upto a specified run time.

Return

int

Syntax

```
PSpiceTranRun <runtime> <relativeTime = false>
```

Parameters

Parameter	Description	Type	Optional	Value Required
runtime	Stop time for transient analysis	int	false	false
relativeTime	True Stop time is relative to last run. False Stop time is absolute.	bool	false	false

Examples

- `PSpiceTranRun 1 false`
- `PSpiceTranRun 1 true`

Related Commands

- [PSpiceTranEnd](#)
- [PSpiceTranNextTime](#)

SetAppendDatMode

Set mode for writing the probe data(.dat) file

Return

string

Syntax

`SetAppendDatMode <appendToDatFile>`

Parameters

Parameter	Description	Type	Optional	Value Required
<code>appendToDatFile</code>	0 - overwrite, 1 - append	int	false	false

Examples

`BoolTest true`

Related Commands

No Related Command

PSpiceSimulationPercent

Displays the simulation time or percentage for each 1% increase in simulation time.

Return

None

Syntax

PSpiceSimulationPercent

Parameters

No Parameters

Example

```
package provide PSpiceTCLCallbacks 1.0
namespace eval PSpiceTCLCallbacks {
}
proc PSpiceTCLCallbacks::PspiceSimulationPercent {} {
    # Callback Function :: To display the current percentage of simulation.
    puts "TIME = [PSpiceCurrentTime] Percentage = [PSpiceCurrentPercentage]%"
}
load {D:/Cadence/17.40/tools/bin/orPSP_ENG64.dll} orpspeng
PSpiceTCLCallback cbk
PSpiceSetCallback cbk

PSpiceSetLicenseBatchMode PSpiceAD
PSpiceSetupAnalysis trans.cir trans.out trans.dat "D:/Cadence/17.40/tools/pspice/library"
PSpiceCommandDo DoTRAN true
PSpiceCommandDo FINISH true
PSpiceTranEnd
```

PSpiceSimulationTime

Displays all the accepted time points as simulation runs.

Return

None

Syntax

```
PSpiceSimulationTime
```

Parameters

No Parameters

Example

```
package provide PSpiceTCLCallbacks 1.0
namespace eval PSpiceTCLCallbacks {
}
proc PSpiceTCLCallbacks::PspiceSimulationTime {} {
    # Call back function :: To display each time points.
    puts "TIME = [PSpiceCurrentTime] "
}
load {D:/Cadence/17.40/tools/bin/orPSP_ENG64.dll} orpspeng
PSpiceTCLCallback cbk
PSpiceSetCallback cbk
PSpiceSetLicenseBatchMode PSpiceAD
PSpiceSetupAnalysis trans.cir trans.out trans.dat "D:/Cadence/17.40/tools/pspice/library"
PSpiceCommandDo DoTRAN true
PSpiceCommandDo FINISH true
PSpiceTranEnd
```

ConvergenceFailure

Modifies simulator options and resumes the simulation in case of convergence failure.

Return

None

Syntax

ConvergenceFailure

Parameters

No Parameters

Example

```
package provide PSpiceTCLCallbacks 1.0
namespace eval PSpiceTCLCallbacks {
}
set isConvergedCalledOnce 0

proc PSpiceTCLCallbacks::ConvergenceFailure {} {
    PSpiceParamSetValue ITL4 10
    PSpiceParamSetValue ABSTOL 10E-12
    ## After setting the values, resume the simulation
    if { $::isConvergedCalledOnce == 0 } {
        PSpiceResume
    }
    set ::isConvergedCalledOnce 1
}

load {D:\Cadence\17.40\tools\bin\orPSP_ENG64.dll} orpspeng
PSpiceTCLCallback cbk
PSpiceSetCallback cbk
PSpiceSetLicenseBatchMode PSpiceAD
PSpiceSetupAnalysis tran.cir tran.out tran.dat "D:/Cadence/17.40/tools/pspice/library"
PSpiceCommandDo DoTRAN true
PSpiceCommandDo FINISH true
PSpiceTranEnd
```

Accessing PSpice Data Files using TCL Commands

Following commands can be used to access PSpice data files:

- [addBlock](#)
- [addDataName](#)
- [addDataPoint](#)
- [addTimePoint](#)
- [calculateBH](#)
- [close](#)
- [closeBlock](#)
- [closeFile](#)
- [create](#)
- [Evaluate](#)
- [get](#)
- [getDataNames](#)
- [getJSON](#)
- [getNumberOfBlocks](#)
- [loadFile](#)
- [setBlock](#)
- [setHeaderData](#)

addBlock

Adds a new block in the file. Each block can hold digital and analog data.

Return

None

Syntax

```
addBlock
```

Parameters

No Parameters

Examples

```
addBlock
```

Related Commands

[closeBlock](#)

addDataName

Adds a new data trace to a DAT file block.

Return

None

Syntax

`addDataName <Name> <Type>`

Parameters

Parameter	Description	Type	Optional	Value Required
Name	Name of new data trace	string	false	false
Type	Type of trace, 3 for analog and 4 for digital.	int	false	false

Examples

- `addDataName "Frequency" 0`
- `addDataName "Gain" 0`

Related Commands

- [addDataPoint](#)
- [addTimePoint](#)

addDataPoint

Adds a data point to .dat file block

Return

None

Syntax

```
addDataPoint <numberReal> [numberImg]
```

Parameters

Parameter	Description	Type	Optional	Value Required
numberReal	String with value of size double	double	false	false
numberImg	String with value of size double.	double	false	false

Examples

```
addDataPoint 0 0
```

Related Commands

- [addTimePoint](#)
- [addDataName](#)

addTimePoint

Adds a time point to a .dat file block.

Return

None

Syntax

```
addTimePoint <number> <Type>
```

Parameters

Parameter	Description	Type	Optional	Value Required
number	String with value of size double	string	false	false
Type	The following Time Point types are supported: 1 - Analog and 2 - Digital.	int	false	false

Examples

```
addTimePoint 0 0
```

Related Commands

- [addDataPoint](#)
- [addDataName](#)

calculateBH

Returns the area under BH curve.

Return

int

Syntax

```
calculateBH <H> <B>
```

Parameters

Parameter	Description	Type	Optional	Value Required
H	The magnetic field trace	string	false	false
B	The flux density trace	string	false	false

Examples

No Examples

Related Commands

[Evaluate](#)

close

Closes the DAT file

Return

bool

Syntax

```
close
```

Parameters

No Parameters

Examples

`close`

Related Commands

- [loadFile](#)
- [getJSON](#)

closeBlock

Closes the current active block.

Return

None

Syntax

`closeBlock`

Parameters

No Parameters

Examples

`closeBlock`

Related Commands

[addBlock](#)

closeFile

Closes the .dat file during data writing.

Return

None

Syntax

```
closeFile
```

Parameters

No Parameters

Examples

```
closeFile
```

Related Commands

[create](#)

create

Creates a new DAT file.

Return

bool

Syntax

```
create <Name> <Version> <MinorVersion> <Type>
```

Parameters

Parameter	Description	Type	Optional	Value Required
Name	Name of the new data file (.dat)	string	false	false
Version	Version number of the new data file (.dat), which is either 9 or 10.	int	false	false
MinorVersion	The default value is 0.	int	false	false
Type	The default value is 0.	int	false	false

Examples

```
create {AC.dat} 10
```

Related Commands

[closeFile](#)

Evaluate

Returns the evaluated expression using PSpice Advanced Analysis expression evaluator

Return

string

Syntax

Evaluate <expression> <StartBlock> <EndBlock>

Parameters

Parameter	Description	Type	Optional	Value Required
expression	The expression to evaluate, such as Max(V(N1)).	string	false	false
StartBlock	The starting block number. This is optional.	int	true	false
EndBlock	The ending block number. This is optional.	int	true	false

Examples

- `evaluate Max(I(C2))`
- `evaluate Avg(I(C2))`
- `evaluate Min(I(C2))`

Related Commands

[calculateBH](#)

get

Returns comma separated data

Return

string

Syntax

get <trace>

Parameters

No Parameters

Examples

- get "Time"
- get "I (C2) "

Related Commands

[getDataNames](#)

getDataNames

Returns all the data names in string

Return

bool

Syntax

getDataNames <Type>

Parameters

Parameter	Description	Type	Optional	Value Required
Type	0 - Analog circuit, 1 - Digital circuit, and 2 - mixed signal circuit	int	false	false

Examples

```
getDataNames 0
```

Related Commands

[get](#)

getJSON

Returns JSON formatted String containing DAT file info

Return

string

Syntax

```
getJSON
```

Parameters

No Parameters

Examples

getJSON

Related Commands

- [loadFile](#)
- [close](#)

getNumberOfBlocks

Returns the total number of blocks in a file. For example, if a monte carlo analysis runs from 1 to 10, the command returns 10.

Return

bool

Syntax

getNumberOfBlocks

Parameters

No Parameters

Examples

getNumberOfBlocks

Related Commands

[setBlock](#)

loadFile

Opens and loads a DAT file

Return

bool

Syntax

```
loadFile <Filename>
```

Parameters

Parameter	Description	Type	Optional	Value Required
Filename	DAT File name	string	false	false

Examples

```
loadFile "D:/tran/tran.dat"
```

Related Commands

- [close](#)
- [getJSON](#)

setBlock

Sets the active block for reading or writing data

Return

bool

Syntax

```
setBlock <blockNumber>
```

Parameters

Parameter	Description	Type	Optional	Value Required
blockNumber	Should be less than the total number of blocks	int	false	false

Examples

```
setBlock 0
```

Related Commands

[getNumberOfBlocks](#)

setHeaderData

Sets the header of the block.

Return

none

Syntax

```
setHeaderData <Name> {string, double, long}
```

Parameters

Parameter	Description	Type	Optional	Value Required
Name	Parameter Name	string	false	false
Value	One of the followings: string, long, or double value	string or long or double	false	false

Examples

- `setHeaderData "CircuitName" "TCL Test circuit"`
- `setHeaderData "SimulationTime" "14:00::53"`
- `setHeaderData "ProgramID" 45645`

Related Commands

- [setBlock](#)
- [addBlock](#)
- [closeBlock](#)
- [getNumberofBlocks](#)

Accessing PSpice Advanced Analysis Design Database using TCL commands

The following TCL commands are used to access PSpice Advanced Analysis design database, such as instance level tolerance, simulation value, smoke values:

- [PSAARReader_enableAAInst_TclEval](#)
- [PSAARReader_enableAAInstDevSmoke](#)
- [PSAARReader_getAAInstParam](#)
- [PSAARReader_getAAInstParam_Val](#)
- [PSAARReader_getAAVar](#)
- [PSAARReader_getPSpiceInstParam](#)
- [PSAARReader_getPSpiceParam](#)
- [PSAARReader_openDesign](#)
- [PSAARReader_save](#)
- [PSAARReader_setAADevParam](#)
- [PSAARReader_setAAInstDevParam](#)
- [PSAARReader_setAAInstParam](#)
- [PSAARReader_setAAInstParam_Tol](#)
- [PSAARReader_setAAVar](#)
- [PSAARReader_setPSpiceInstParam](#)
- [PSAARReader_setPSpiceParam](#)

PSAARReader_enableAAInst_TclEval

Enables TCL to evaluate the instance

Return

int

Syntax

```
PSAARReader_enableAAInst_TclEval <Instance>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<Instance>	Instance Name	String	FALSE	-

Examples

```
PSAARReader_enableAAInst_TclEval "R1"
```

Related Commands

[PSAARReader_enableAAInstDevSmoke](#)

PSAARReader_enableAAInstDevSmoke

Enables Smoke profile on a instance.

Return

int

Syntax

```
PSAARReader_enableAAInstDevSmoke <Instance>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<Instance>	Instance Name	String	FALSE	-

Examples

```
PSAARReader_enableAAInstDevSmoke "R1"
```

Related Commands

[PSAARReader_enableAAInst_TclEval](#)

PSAARReader_getAAInstParam

Gets an instance parameter.

Return

int

Syntax

```
PSAARReader_getAAInstParam <Inst> <param> <value> <postol> <negtol>
```


Parameters

Parameter	Description	Type	Optional	Value Required
<Inst>	Instance Name	string	FALSE	-
<Param>	Parameter Name	string	FALSE	-
<Value>	String to get value	string	FALSE	-
<postol>	String to get postol.	string	FALSE	-
<Negtol>	String to get Negtol.	string	FALSE	-

Examples

No Examples

Related Commands

- [PSAARReader_getAAInstParam_Val](#)
- [PSAARReader_getAAVar](#)
- [PSAARReader_getPSpiceInstParam](#)
- [PSAARReader_getPSpiceParam](#)

PSAARReader_getAAInstParam_Val

Gets an instance parameter value only.

Return

int

Syntax

PSAARReader_getAAInstParam_Val <inst> <param> <value>

Parameters

Parameter	Description	Type	Optional	Value Required
<Inst>	Instance Name	string	FALSE	-
<Param>	Parameter Name	string	FALSE	-
<Value>	String to get value	string	FALSE	-

Examples

No Examples

Related Commands

- [PSAARReader_getAAInstParam](#)
- [PSAARReader_getAAVar](#)
- [PSAARReader_getPSpiceInstParam](#)

PSAARReader_getAAVar

Gets Advanced Analysis Variable Values.

Return

string

Syntax

PSAARReader_getAAVar <name> <value>

Parameters

Parameter	Description	Type	Optional	Value Required
<Name>	Variable name	string	FALSE	-
<Value>	string to get value	string	FALSE	-

Examples

No Examples

Related Commands

- [PSAARReader_getPSpiceInstParam](#)
- [PSAARReader_getAAInstParam_Val](#)
- [PSAARReader_getAAInstParam](#)

PSAARReader_getPSpiceInstParam

Gets PSpice instance param from simulator.

Return

int

Syntax

PSAARReader_getPSpiceInstParam <inst> <param> <value>

Parameters

Parameter	Description	Type	Optional	Value Required
<inst>	Instance Name	string	FALSE	-
<param>	Parameter name	string	FALSE	-
<value>	String to get value	string	FALSE	-

Examples

No Examples

Related Commands

[PSAARReader_getPSpiceParam](#)

PSAARReader_getPSpiceParam

Gets PSpice Parameters from Simulator.

Return

string

Syntax

```
PSAARReader_getPSpiceParam <name> <value>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<name>	Variable name	string	FALSE	-
<value>	String to get value	string	FALSE	-

Examples

No Examples

Related Commands

[PSAARReader_getPSpiceInstParam](#)

PSAARReader_openDesign

Opens a design specified in profile.

Return

int

Syntax

```
PSAARReader_openDesign
```

Parameters

No Parameters

Examples

No Examples

Related Commands

No Related Command

PSAARReader_save

Saves profile.

Return

int

Syntax

```
PSAARReader_save
```

Parameters

No Parameters

Examples

No Examples

Related Commands

No Related Command

PSAARReader_setAADevParam

Sets Device parameter.

Return

int

Syntax

```
PSAARReader_setAADevParam <dev> <param> <value>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<dev>	Device name	string	FALSE	-
<param>	Parameter name	string	FALSE	-
<value>	Value to set	int	FALSE	-

Examples

No Examples

Related Commands

[PSAARReader_setAAInstDevParam](#)

PSAARReader_setAAInstDevParam

Sets instance parameter.

Return

int

Syntax

```
PSAARReader_setAAInstDevParam <inst> <param> <value>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<inst>	Instance name	string	FALSE	-
<param>	Parameter name	string	FALSE	-
<value>	Value to set	int	FALSE	-

Examples

No Examples

Related Commands

[PSAARReader_setAAInstParam](#)

PSAARReader_setAAInstParam

Sets instance parameter.

Return

int

Syntax

PSAARReader_setAAInstParam <inst> <param> <value> <postol> <negtol>

Parameters

Parameter	Description	Type	Optional	Value Required
<inst>	Instance name	string	FALSE	-
<param>	Parameter name	string	FALSE	-
<value>	Value to set	int	FALSE	-
<postol>	value to set	int	FALSE	-
<negtol>	Value to set.	int	FALSE	-

Examples

No Examples

Related Commands

- [PSAARReader_setAAInstDevParam](#)
- [PSAARReader_setAAInstParam_Tol](#)

PSAARReader_setAAInstParam_Tol

Sets tolerance parameters.

Return

int

Syntax

```
PSAARReader_setAAInstParam_Tol <inst> <param> <postol> <negtol> [<Distribution>]
```

Parameters

Parameter	Description	Type	Optional	Value Required
<inst>	Instance name	string	FALSE	-
<param>	Parameter name	string	FALSE	-
<postol>	Value to set	string	FALSE	-
<negtol>	Value to set	string	FALSE	-
<Distribution>	Value to set	string	TRUE	-

Examples

```
PSAARReader_setAAInstParam_Tol $Instance "C3" "VALUE" "10%" "10%" "gauss 0.4"
```

Related Commands

No Related Command

PSAARReader_setAAVar

Set AA Variable.

Return

int

Syntax

PSAARReader_setAAVar <name> <Value>

Parameters

Parameter	Description	Type	Optional	Value Required
<name>	Variable name	string	FALSE	-
<value>	string to get value	string	FALSE	-

Examples

No Examples

Related Commands

No Related Command

PSAARReader_setPSpiceInstParam

Sets PSpice instance parameters in the PSpice engine.

Return

int

Syntax

PSAARReader_setPSpiceInstParam <inst> <param> <value>

Parameters

Parameter	Description	Type	Optional	Value Required
<inst>	Instance name	string	FALSE	-
<param>	Parameter name	string	FALSE	-
<value>	string to get value	string	FALSE	-

Examples

No Examples

Related Commands

No Related Command

PSAARReader_setPSpiceParam

Sets PSpice parameters in the Pspice engine.

Return

int

Syntax

```
PSAARReader_setPSpiceParam <param> <value>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<param>	Parameter name	string	FALSE	-
<value>	string to set value	string	FALSE	-

Examples

No Examples

Related Commands

No Related Command

Customizing Various Analysis in PSpice Advanced Analysis using TCL Commands

The following TCL commands are used for callbacks to customize tolerance on instance parameters for Sensitivity or Monte Carlo analysis. These commands can also be used to customize derating and smoke tests for Smoke analysis:

- `_cdnPspAAAnalysisComplete`
- `_cdnPspAAProcessDesignStart`
- `_cdnPspAAProcessInst`
- `_cdnPspAAProcessPostInst`
- `_cdnPspAASmokeDerateReliability`
- `_cdnPspAASmokeDerateTemp`
- `_cdnPspAASmokePostPDM&TB`
- `_cdnPspAASmokePostPDM&TJ`
- `_cdnPspAASmokePostPDML&TJL`

`_cdnPspAAAnalysisComplete`

Called after Analysis is complete.

Return

int

Syntax

```
_cdnPspAAAnalysisComplete
```

Parameters

No Parameters

Examples

No Examples

Related Commands

No Related Command

_cdnPspAAProcessDesignStart

Called on start of Sensitivity and MonteCarlo analysis.

Return

int

Syntax

```
_cdnPspAAProcessDesignStart
```

Parameters

No Parameters

Examples

No Examples

Related Commands

No Related Command

_cdnPspAAProcessInst

Called if instance has been enabled with PSAARReader_enableAAInst_TclEval

Return

int

Syntax

```
_cdnPspAAProcessInst <Instance_Name>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<Instance_Name>	Instance name	int	FALSE	-

Examples

No Examples

Related Commands

No Related Command

_cdnPspAAProcessPostInst

Called after Instance has been processed.

Return

int

Syntax

_cdnPspAAProcessPostInst

Parameters

No Parameters

Examples

No Examples

Related Commands

No Related Command

_cdnPspAASmokeDerateReliability

Called for user defined reliability deration for instances enabled with PSAARader_enableAAInst_TclEval.

Return

int

Syntax

_cdnPspAASmokeDerateReliability

Parameters

No Parameters

Examples

No Examples

Related Commands

No Related Command

_cdnPspAASmokeDerateTemp

Called for user defined temperature deration for instances enabled with PSAARreader_enableAAInst_TclEval.

Return

int

Syntax

```
_cdnPspAASmokeDerateTemp <inst> <param>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<inst>	Instance name	string	FALSE	-
<param>	Parameter name	string	FALSE	-

Examples

No Examples

Related Commands

No Related Command

_cdnPspAASmokePostPDM&TB

Called for user defined temperature calculations for instances enabled with PSAARReader_enableAAInst_TclEval.

Return

int

Syntax

```
_cdnPspAASmokePostPDM&TB <inst>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<inst>	Instance name	string	FALSE	-

Examples

No Examples

Related Commands

No Related Command

_cdnPspAASmokePostPDM&TJ

Called for user defined Junction temperature calculations for instances enabled with PSAARReader_enableAAInst_TclEval.

Return

int

Syntax

```
_cdnPspAASmokePostPDM&TJ <inst>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<inst>	Instance name	string	FALSE	-

Examples

No Examples

Related Commands

No Related Command

_cdnPspAASmokePostPDML&TJL

Called for user defined Power loss calculations for instances enabled with PSAARreader_enableAAInst_TclEval.

Return

int

Syntax

```
_cdnPspAASmokePostPDML&TJL <inst>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<inst>	Instance name	string	FALSE	-

Examples

No Examples

Related Commands

No Related Command

Accessing PSpice Advanced Analysis User Interface using TCL Commands

Following are the TCL commands to access user interface of PSpice Advanced Analysis:

- [pspAAGetActiveAap](#)
- [pspAAGetAssociated](#)
- [pspAAOpenAssociated](#)
- [pspAASelectAap](#)

pspAAGetActiveAap

Returns an active advanced analysis profile path that is opened in PSpice Advanced Analysis.

Return

string

Syntax

```
pspAAGetActiveAap
```

Parameters

No Parameters

Examples

No Examples

Related Commands

No Related Command

pspAAGetAssociated

Returns the JSON string format for all associated profiles of the specified advanced analysis profile.

Return

string

Syntax

```
pspAAGetAssociated <aap_file>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<aap_file>	PSpice Advanced Analysis profile path	string	FALSE	-

Examples

```
pspAAGetAssociated ac.aap
```

Related Commands

- [pspAAGetActiveAap](#)
- [pspAASelectAap](#)
- [pspAAOpenAssociated](#)

pspAAOpenAssociated

Opens the associated profile for the specified advanced analysis profile

Return

int

Syntax

```
pspAAOpenAssociated <.aap file> <profile>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<aap_file>	PSpice Advanced Analysis Profile Path	string	FALSE	-
<profile>	PSpice Simulation Profile path	string	FALSE	-

Examples

```
pspAAOpenAssociated schematic1.aap ac.sim
```


Related Commands

- [pspAAGetActiveAap](#)
- [pspAASelectAap](#)
- [pspAAGetAssociated](#)

pspAASelectAap

Opens a File - Open window to open a .aap profile in PSpice Advanced Analysis.

Return

string

Syntax

```
pspAASelectAap
```

Parameters

No Parameters

Examples

No Examples

Related Commands

No Related Command

Executing Advanced Analysis using TCL Commands

This chapter covers the following TCL commands that can be used to execute various advanced analysis and specific advanced analysis, such as Monte Carlo, Sensitivity, Smoke, Optimizer, and Parametric Plot:

- [TCL Commands to Execute PSpice Advanced Analysis](#)
- [TCL Commands for the Monte Carlo Advanced Analysis](#)
- [TCL Commands for the Sensitivity Advanced Analysis](#)
- [TCL Commands for the Smoke Advanced Analysis](#)
- [TCL Commands for the Optimizer Advanced Analysis](#)
- [TCL Commands for the Parametric Plotter Advanced Analysis](#)

TCL Commands to Execute PSpice Advanced Analysis

The following TCL commands are used for PSpice Advanced Analysis execution:

- [addSpecifications](#)
- [getMonteCarloSettings](#)
- [getMonteController](#)
- [getOptimizerController](#)
- [getPplotController](#)
- [getSensitivityController](#)
- [getSmokeController](#)
- [openAAProfile](#)

- [pspAAcreateInterface](#)
- [pspAAdeleteInterface](#)
- [saveAAPProfile](#)
- [saveAssociatedProfile](#)
- [setCurrentEngine](#)
- [setCurrentGear](#)
- [setMonteCarloSettings](#)
- [setSettings](#)
- [setSpecifications](#)

addSpecifications

Adds PSpice Advanced Analysis specifications object.

Return

int

Syntax

```
addSpecifications <Object>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<Object>	PSpice Advanced Analysis Specifications Object	object	FALSE	-

Examples

No Examples

Related Commands

[setSpecifications](#)

getMonteCarloSettings

Returns the MonteCarlo Settings.

Return

object

Syntax

```
getMonteCarloSettings <MonteSettings>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<MonteSettings>	MonteSettings created from new_MonteSettings.	int	FALSE	-

Examples

No Examples

Related Commands

[setMonteCarloSettings](#)

getMonteController

Returns the MonteController object in AA Server.

Return

Object

Syntax

```
getMonteController
```

Parameters

No Parameters

Examples

No Examples

Related Commands

No Related Command

getOptimizerController

Returns the OptimizerController object in AA Server.

Return

Object

Syntax

```
getOptimizerController
```

Parameters

No Parameters

Examples

No Examples

Related Commands

- [getPplotController](#)
- [getSensitivityController](#)
- [getSmokeController](#)

getPplotController

Returns the Parametric Plot Controller object in AA Server.

Return

Object

Syntax

```
getPplotController
```

Parameters

No Parameters

Examples

No Examples

Related Commands

- [getSensitivityController](#)
- [getSmokeController](#)
- [getOptimizerController](#)

getSensitivityController

Returns the Sentitivity Controller object in AA Server.

Return

Object

Syntax

```
getSensitivityController
```

Parameters

No Parameters

Examples

No Examples

Related Commands

- [getOptimizerController](#)
- [getPplotController](#)
- [getSmokeController](#)

getSmokeController

Returns the Smoke Controller object in AA Server.

Return

Object

Syntax

```
getSmokeController
```

Parameters

No Parameters

Examples

No Examples

Related Commands

- [getSensitivityController](#)
- [getPplotController](#)
- [getOptimizerController](#)

openAAProfile

Opens the AA profile.

Return

int

Syntax

```
openAAProfile <Path>
```


Parameters

Parameter	Description	Type	Optional	Value Required
<Path>	Advanced Analysis Profile path	string	FALSE	-

Examples

```
$pAAInterface openAAProfile "D:/Cadence/tools/pspice/capture_samples/advanls/bpf/bpf-  
PSpiceFiles/bpf/bpf.aap"
```

Related Commands

- [saveAAProfile](#)
- [saveAssociatedProfile](#)

pspAAcreateInterface

Returns a new interface object to Advanced Analysis (AA) Server.

Return

object

Syntax

```
pspAAcreateInterface
```

Parameters

No Parameters

Examples

```
pspAAcreateInterface
```

Related Commands

[pspAAdeleteInterface](#)

pspAAdeleteInterface

Deletes the interface object to Advanced Analysis (AA) Server.

Return

int

Syntax

```
deleteAAInterface <interface>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<interface>	AA server interface object returned by pspAAcreateInterface command.	int	FALSE	-

Examples

```
pspAAdeleteInterface $pAAInterface
```

Related Commands

[pspAAcreateInterface](#)

saveAAProfile

Saves the AA Profile.

Return

int

Syntax

```
saveAAProfile <Path>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<Path>	The AA profile to save. Empt string, such as "", saves the current profile.	int	FALSE	-

Examples

```
$pAAInterface saveAAProfile ""
```

Related Commands

- [openAAProfile](#)
- [saveAssociatedProfile](#)

saveAssociatedProfile

Saves the current Profile settings to Associated Profile directory.

Return

int

Syntax

```
saveAssociatedProfile <Path>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<Path>	Associated AA Profile Path	string	FALSE	-

Examples

No Examples

Related Commands

- [openAAProfile](#)
- [saveAAProfile](#)

setCurrentEngine

Sets the engine type for optimizer, such as, HYBRID, RANDOM, LSQ, DISCRETE.

Return

int

Syntax

```
setCurrentEngine <Engine>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<Engine>	One of the optimizer engine types	string	FALSE	-

Examples

No Examples

Related Commands

[setCurrentGear](#)

setCurrentGear

Sets the Gear type for optimizer, such as LEGACY_GEAR, DEFAULT_GEAR, SIM_REF_DATA_ALIKE, WEIGHTED_REF_DATA.

Return

int

Syntax

```
setCurrentGear <gear>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<Gear>	Type of gear for optimizer	string	FALSE	-

Examples

No Examples

Related Commands

[setCurrentEngine](#)

setMonteCarloSettings

Sets the MonteCarlo Settings.

Return

int

Syntax

```
setMonteCarloSettings <MonteSettings>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<MonteSettings>	MonteCarlo Settings Object.	int	FALSE	-

Examples

No Examples

Related Commands

[getMonteCarloSettings](#)

setSettings

Sets Advanced Analysis Settings Object.

Return

int

Syntax

```
setSettings <Object>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<Object>	Advanced Analysis Settings Object	object	FALSE	-

Examples

No Examples

Related Commands

No Related Command

setSpecifications

Sets Advanced Analysis Specifications Object.

Return

int

Syntax

```
setSpecifications <Object>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<Object>	Advanced Analysis Specifications Object.	object	FALSE	-

Examples

No Examples

Related Commands

[addSpecifications](#)

TCL Commands for the Monte Carlo Advanced Analysis

The following TCL commands are used for the Monte Carlo Advanced Analysis:

delete_CMonteSettings

Deletes MonteCarlo Settings Object.

Return

int

Syntax

```
delete_CMonteSettings <Object>
```


Parameters

Parameter	Description	Type	Optional	Value Required
<Object>	MonteCarlo Settings object	Object	FALSE	-

Examples

No Examples

Related Commands

[new_CMonteSettings](#)

delete_CMonteSpecification

Deletes MonteCarlo Specifications Object.

Return

int

Syntax

```
delete_CMonteSpecification <Object>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<Object>	Deletes MonteCarlo Specifications Object.	object	FALSE	-

Examples

No Examples

Related Commands

[new_CMonteSpecification](#)

new_CMonteSettings

Returns a new MonteCarlo Settings Object.

Return

object

Syntax

```
new_CMonteSettings
```

Parameters

No Parameters

Examples

No Examples

Related Commands

No Related Command

new_CMonteSpecification

Returns a new MonteCarlo Specifications Object.

Return

object

Syntax

```
new_CMonteSpecification
```

Parameters

No Parameters

Examples

No Examples

Related Commands

No Related Command

TCL Commands for the Sensitivity Advanced Analysis

The following TCL commands are related to Sensitivity analysis:

- [delete_CSensParameter](#)
- [delete_CSensSpecification](#)
- [new_CSensParameter](#)
- [new_CSensSpecification](#)

delete_CSensParameter

Deletes Sensitivity Parameter Object

Return

int

Syntax

```
delete_CSensParameter <Object>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<Object>	Sensitivity Parameter Object.	object	FALSE	-

Examples

No Examples

Related Commands

[new_CSensParameter](#)

delete_CSensSpecification

Deletes Sensitivity Specifications Object.

Return

int

Syntax

```
delete_CSensSpecification <Object>
```

Parameters

Parameter	Description	Type	Optional	Value Required
<Object>	Sensitivity Specifications Object	Object	FALSE	-

Examples

No Examples

Related Commands

[new_CSensSpecification](#)

new_CSensParameter

Returns a new Sensitivity Parameter Object.

Return

int

Syntax

```
new_CSensParameter
```

Parameters

No Parameters

Examples

No Examples

Related Commands

[new_CSensSpecification](#)

new_CSensSpecification

Returns a new Sensitivity Specifications Object.

Return

int

Syntax

```
new_CSensSpecification
```

Parameters

No Parameters

Examples

No Examples

Related Commands

[new_CSensParameter](#)

TCL Commands for the Smoke Advanced Analysis

You can get information about the TCL commands used for Smoke analysis using the following TCL commands in the PSpice Advanced Analysis command window:

```
package require PspAACtrlTcl

foreach x [lsort [info commands *Smoke*]] {puts $x}

foreach x [lsort [info commands *Smk*]] {puts $x}
```

TCL Commands for the Optimizer Advanced Analysis

You can get more information about the TCL commands used for Optimizer analysis using the following TCL commands in the command window:

```
package require PspAACtrlTcl  
  
foreach x [lsort [info commands *Opt*]] {puts $x}
```

TCL Commands for the Parametric Plotter Advanced Analysis

You can get more information about the TCL commands used for the Parametric Plotter analysis using the following TCL commands in the command window:

```
package require PspAACtrlTcl  
  
foreach x [lsort [info commands *Pplot*]] {puts $x}
```